

MANUAL OF TECHNICAL PROVISIONS FOR THE AERONAUTICAL TELECOMMUNICATION NETWORK (ATN)

SECOND EDITION — 1999



*Approved by the Secretary General
and published under his authority*

INTERNATIONAL CIVIL AVIATION ORGANIZATION

FOREWORD

The material contained in this document was originally developed as the detailed part of the first set of Standards and Recommended Practices (SARPs) for the aeronautical telecommunication network (ATN) which has commonly been referred to as the CNS/ATM-1 Package. It was intended to make the material an appendix to the new Chapter 3 of Annex 10, Volume III, Part I, containing broad, general, stable and mostly regulatory-type provisions (the core part of new ATN SARPs).

In December 1997, the Air Navigation Commission (ANC), while conducting the final review of draft ATN SARPs, noted that actual implementation and operational experience was yet to be gained by the international civil aviation community. In this regard, the ANC agreed that the detailed part of ATN SARPs should be published as an ICAO manual (to be updated annually, if necessary), while retaining its SARPs-style language. The ANC will review the status of the document, in its entirety or in parts, after sufficient implementation and operational experience has been gained and the requirements for further standardization, in the interests of safety, regularity and efficiency of international civil aviation have been better ascertained.

This document consists of five Sub-Volumes:

- Sub-Volume I — Introduction and System Level Requirements
- Sub-Volume II — Air-Ground Applications
- Sub-Volume III — Ground-Ground Applications
- Sub-Volume IV — Upper Layer Communications Service (ULCS)
- Sub-Volume V — Internet Communications Service (ICS)

Provisions contained in Sub-Volumes II, III, IV and V have been developed in accordance with system requirements specified in Sub-Volume I.

In line with the agreement by the ANC that the document should be updated on a yearly basis (if deemed necessary), the Second Edition has been published to incorporate changes necessitated by continuing validation and actual implementation activities.

TABLE OF CONTENTS

SUB-VOLUME I. INTRODUCTION AND SYSTEM LEVEL REQUIREMENTS

1.1	Definitions and References	I-1
1.1.1	Definitions	I-1
1.1.2	References	I-22
1.2	General	I-31
1.3	System Level Requirements	I-33

SUB-VOLUME II. AIR-GROUND APPLICATIONS

2.1	Context Management Application	II-1
2.1.1	Introduction	II-1
2.1.2	General Requirements	II-4
2.1.3	The Abstract Service	II-5
2.1.4	Formal Definitions of Messages	II-17
2.1.5	Protocol Definition	II-21
2.1.6	Communication Requirements	II-64
2.1.7	CM User Requirements	II-65
2.1.8	Subsetting Rules	II-71
2.2	Automatic Dependent Surveillance Applications	II-75
2.2.1	Automatic Dependent Surveillance Application	II-75
2.2.2	Automatic Dependent Surveillance Report Forwarding Application	II-211
2.3	Controller Pilot Data Link Communication Application	II-241
2.3.1	Introduction	II-241
2.3.2	General Requirements	II-244
2.3.3	The Abstract Service	II-245
2.3.4	Formal Definitions of Messages	II-257
2.3.5	Protocol Definition	II-307
2.3.6	Communication Requirements	II-358
2.3.7	CPDLC User Requirements	II-359
2.3.8	Subsetting Rules	II-416
2.4	Flight Information Services Application	II-420
2.4.1	Introduction	II-420
2.4.2	General Requirements	II-427
2.4.3	The Abstract Service	II-428
2.4.4	Formal Definitions of Messages	II-439
2.4.5	Protocol Definition	II-456
2.4.6	Communication Requirements	II-508

2.4.7	FIS User Requirements	II-509
2.4.8	Subsetting Rules	II-516

SUB-VOLUME III. GROUND-GROUND APPLICATIONS

3.1	ATS Message Handling Services (ATSMHS)	III-1
3.1.1	Introduction	III-1
3.1.2	ATS Message Service	III-7
3.1.3	ATN Pass-through Service	III-111
3.2	ATS Interfacility Data Communications	III-123
3.2.1	Introduction	III-123
3.2.2	General Requirements	III-127
3.2.3	The AIDC-AE Abstract Service	III-128
3.2.4	The AIDC-ASE Abstract Service	III-144
3.2.5	The AIDC Control Function	III-155
3.2.6	The AIDC-ASE Protocol Definition	III-188
3.2.7	AIDC Formal Definitions	III-229
3.2.8	Communication Requirements	III-251
3.2.9	AIDC-user Requirements	III-252
3.2.10	Sequence Diagrams	III-255

SUB-VOLUME IV. UPPER LAYER COMMUNICATIONS SERVICE

4.1	Introduction	IV-1
4.1.1	Scope and Objectives	IV-1
4.1.2	Background	IV-2
4.1.3	Structure of UL Communications Service Specification	IV-3
4.1.4	Upper Layer Functionality	IV-4
4.2	Dialogue Service Description	IV-6
4.2.1	Scope of Dialogue Service	IV-6
4.2.2	Service Primitives	IV-7
4.2.3	Service Definition	IV-8
4.3	Application Entity (AE) Description	IV-17
4.3.1	Introduction	IV-17
4.3.2	Application Level Naming and Context Definition	IV-19
4.3.3	Control Function Specification	IV-27
4.4	Session Layer Requirements	IV-58
4.4.1	Protocol Versions Implemented	IV-58
4.4.2	Session Functional Units	IV-59
4.4.3	Protocol Mechanisms	IV-60
4.4.4	Supported Roles	IV-62
4.4.5	Supported SPDUs	IV-64
4.4.6	Use of Null-encoding and Short-connect Protocol Options	IV-67

4.4.7	Mapping to the ATN Internet Transport Service	IV-68
4.5	Presentation Layer Requirements	IV-70
4.5.1	Protocol Mechanisms	IV-70
4.5.2	Use of Null-encoding and Short-connect Protocol Options	IV-71
4.5.3	Mapping of Presentation Primitives to the Null Encoding Option	IV-72
4.5.4	Functional Units	IV-73
4.5.5	Elements of Procedure	IV-75
4.5.6	Supported Presentation Protocol Data Units (PPDUs)	IV-77
4.6	ACSE Specification	IV-79
4.6.1	Protocol Details	IV-79
4.6.2	Protocol Versions	IV-80
4.6.3	Supported Roles	IV-81
4.6.4	Protocol Mechanisms	IV-83
4.6.5	ACSE Functional Units	IV-84
4.6.6	Supported APDUs	IV-85
4.6.7	Mapping to the Presentation Service	IV-91

SUB-VOLUME V. INTERNET COMMUNICATIONS SERVICE

5.1	Introduction	V-1
5.2	Definitions and Concepts	V-2
5.2.1	Objectives and Goals	V-2
5.2.2	Definitions	V-3
5.2.3	ATN End Systems	V-8
5.2.4	ATN Routers	V-9
5.2.5	ATN Subnetworks	V-12
5.2.6	Quality of Service Concept	V-14
5.2.7	ATN Security Concept	V-15
5.2.8	ATN Use of Priority	V-19
5.3	ATN Routing	V-23
5.3.1	Introduction	V-23
5.3.2	Service Provided by an ATN Router	V-24
5.3.3	The Deployment of ATN Components	V-31
5.3.4	Ground/Ground Interconnection	V-33
5.3.5	Air/Ground Interconnection	V-35
5.3.6	Handling Routing Information	V-55
5.3.7	Policy Based Selection of Routes for Advertisement to Adjacent RDs	V-56
5.4	Network and Transport Addressing Specification	V-64
5.4.1	Introduction	V-64
5.4.2	Transport Layer Addressing	V-65
5.4.3	Network Layer Addressing	V-67

5.5	Transport Service and Protocol Specification	V-79
5.5.1	General	V-79
5.5.2	Connection Mode Transport Layer Operation	V-81
5.5.3	Connectionless Mode Transport Protocol Operation	V-103
5.6	Internetwork Service and Protocol Specification	V-107
5.6.1	Introduction	V-107
5.6.2	ATN Specific Features	V-108
5.6.3	ATN Specific Requirements for ISO/IEC 8473	V-115
5.6.4	APRLs	V-117
5.7	Specification of Subnetwork Dependent Convergence Functions	V-138
5.7.1	Introduction	V-138
5.7.2	Service Provided by the SND CF	V-139
5.7.3	SND CF for ISO/IEC 8802-2 Broadcast Subnetworks	V-141
5.7.4	SND CF for the Common ICAO Data Interchange Network (CIDIN)	V-142
5.7.5	SND CF for ISO/IEC 8208 General Topology Subnetworks	V-143
5.7.6	SND CF for ISO/IEC 8208 Mobile Subnetworks	V-144
5.7.7	ATN SND CF Protocol Requirements List	V-206
5.8	Routing Information Exchange Specification	V-219
5.8.1	Introduction	V-219
5.8.2	End System to Intermediate System Routing Information Exchange Protocol (ES-IS) over Mobile Subnetworks	V-220
5.8.3	Intermediate System to Intermediate System Inter-Domain Routing Information Exchange Protocol	V-225
5.9	Systems Management Provisions	V-252
5.9.1	Introduction	V-252

Sub-Volume I

Introduction and System Level Requirements

NOTE ON THE SECOND EDITION

The list below shows the parts of this sub-volume that are different from similar parts of the first edition.

Reference	Nature of change
1.1.1 (definition of ATIS)	Modification
1.1.2 (additional notes and ISO/IEC 8802-2 and 3 references)	Addition

1.1 DEFINITIONS AND REFERENCES

1.1.1 DEFINITIONS

Note 1.— The aeronautical telecommunication network (ATN) comprises application entities and communication services which allow ground, air-to-ground and avionics data subnetworks to interoperate by adopting common interface services and protocols based on the International Organization for Standardization (ISO) open systems interconnection (OSI) reference model.

Note 2.— This document addresses the following ATN technical requirements:

- a) General and system level requirements;*
- b) ATN application entity requirements;*
 - 1) System application entity requirements;*
 - i) Context management (CM) application;*
 - 2) Air-ground application entity requirements;*
 - i) Controller pilot data link communication (CPDLC) application;*
 - ii) Automatic dependent surveillance (ADS) application;*
 - iii) Flight information service (FIS) applications;*
 - 3) Ground-ground application entity requirements;*
 - i) Air traffic services (ATS) inter-centre communication (ICC) applications;*
 - ii) ATS message handling service (ATSMHS) application;*
- c) ATN communication service requirements;*
 - 1) Upper layer communications service;*
 - 2) Internet communications service.*

Note 3.— An overview of this document is depicted in Figure 1-1.

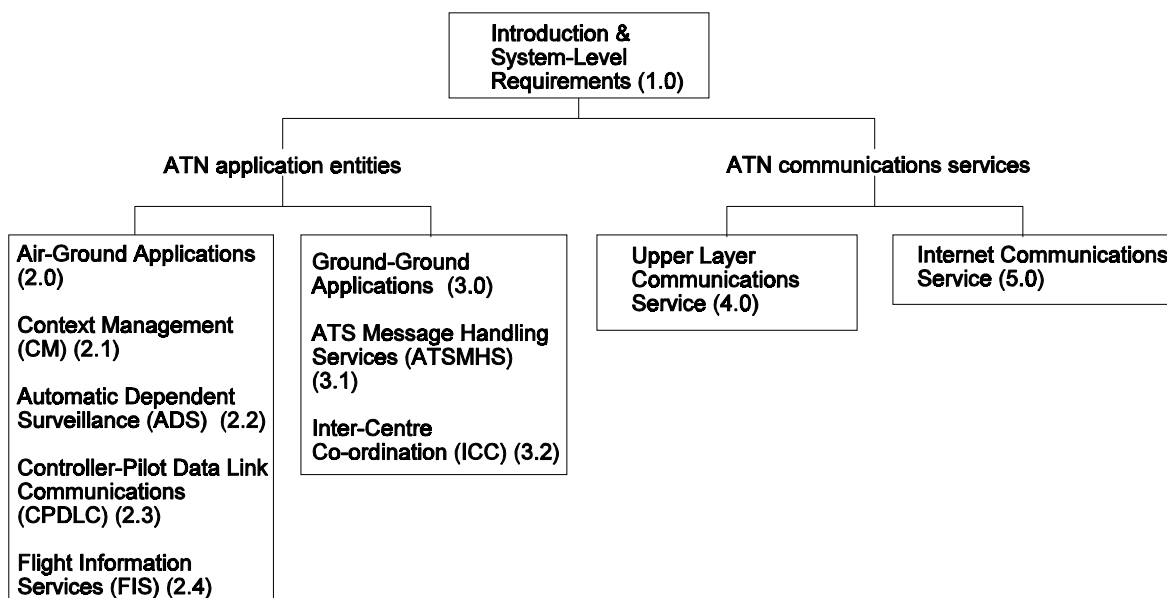


Figure 1.1. Overview of document

When the following terms are used in this document, they have the following meanings:

Abstract service interface. The abstract interface between the application entity (AE) and the application user.

Abstract syntax notation One (ASN.1). Abstract syntax notation One is defined in ISO/IEC 8824-1. The purpose of this notation is to enable data types to be defined, and values of those types specified, without determining their actual representation (encoding) for transfer by protocols.

Addressing plan. A plan that provides common address syntax and management of global addresses for the unambiguous identification of all end and intermediate systems in accordance with the rules prescribed in ISO/IEC 7498-3 and ISO/IEC TR 10730.

Administrative domain. A collection of end systems, intermediate systems and subnetworks operated by a single organization or administrative authority. An administrative domain may be internally divided into one or more routing domains..

ADS. The symbol used to designate automatic dependent surveillance.

ADS application. An ATN application that provides ADS data from the aircraft to the ATS unit(s) for surveillance purposes.

ADS Contract. An agreement between the ADS ground-user and the ADS air-user that the latter will provide reports to the former under the conditions specified in the contract.

Aeronautical administrative communication (AAC). Communication used by aeronautical operating agencies related to the business aspects of operating their flights and transport services. This

communication is used for a variety of purposes, such as flight and ground transportation, bookings, deployment of crew and aircraft or any other logistical purposes that maintains or enhances the efficiency of overall flight operation.

Aeronautical administrative messages. Messages regarding the operation or maintenance of facilities provided for the safety or regularity of aircraft operation. Messages concerning the functioning of the ATN and messages exchanged between government civil aviation authorities relating to aeronautical services.

Aeronautical fixed telecommunications network (AFTN). A world-wide system of aeronautical fixed circuits provided, as part of the aeronautical fixed service, for the exchange of messages and/or digital data between aeronautical fixed stations having the same or compatible communications characteristics.

Aeronautical industry service communication (AINSC). Communication related to aeronautical industry services including aeronautical operational control communication, aeronautical administrative communication, and aeronautical passenger communication. This communication involves one or more aeronautical industry service administrations. This term is used for purposes of address administration.

Aeronautical information service (AIS) messages. Messages concerning the aeronautical information service defined in ANNEX 15.

Aeronautical mobile-satellite service (AMSS). The AMSS comprises satellites, aeronautical earth stations (AESs), ground earth stations (GESs) and associated ground facilities such as a network coordination center. It uses the satellite subnetwork to provide aeronautical communication services between aircraft and ground users. Technical requirements for the AMSS are contained in Annex 10, Volume III, Part I, Chapter 4. The ATN supports the packet-mode data exchange provided by the AMSS.

Aeronautical operational control (AOC). Communication required for the exercise of authority over the initiation, continuation, diversion or termination of flight for safety, regularity and efficiency reasons.

Aeronautical passenger communication (APC). Communication relating to the non-safety voice and data services to passengers and crew members for personal communication.

AFTN. The symbol used to designate aeronautical fixed telecommunication network

AFTN/AMHS gateway. An end system which provides bi-directional interworking between users of the ATS message service and users connected to the AFTN.

AFTN/ATN Type A gateway. An end system which provides a bi-directional interface between the ATN and the AFTN for the purpose of conveying AFTN messages over the ATN by implementation of the ATN pass-through service.

AFTN form address (AF-address). Either an AFTN addressee indicator as specified in Annex 10, Volume II, paragraphs 4.4.3.1.2 and 4.4.16.2.1.3 which is used to locate AMHS users, either direct or indirect, in the AFTN address space or a predetermined distribution addressee indicator (PDAI) as specified in Annex 10, Volume II, 4.4.14.

Note.— An AF-address (AFTN-form) is an ICAO AFTN 8-letter addressee indicator.

AIDC. The symbol used to designate ATS interfacility data communication.

AINSC. The symbol used to designate aeronautical industry service communication.

Air application service element (air-ASE). An abstract part of the aircraft system that performs the communication related functions of the application.

Airborne collision avoidance system (ACAS). An aircraft system based on secondary surveillance radar (SSR) transponder signals which operates independently of ground-based equipment to provide advice to the pilot on potential conflicting aircraft that are equipped with SSR transponders.

Aircraft address. A unique combination of twenty-four bits available for assignment to an aircraft for the purpose of air-ground communications, navigation and surveillance.

Aircraft flight identification. A group of letters, figures or a combination thereof which is either identical to, or the coded equivalent of, the aircraft call sign to be used in air-ground communication and which is used to identify the aircraft in ground-ground air traffic services communication.

Air-ground application. An application that has one peer application on an aircraft and its other peer application on the ground. An air-ground application may require the use of ground-ground subnetworks.

Air traffic control (ATC) clearance. Authorization for an aircraft to proceed under conditions specified by an air traffic control unit.

Note 1.— For convenience the term “air traffic control clearance” is frequently abbreviated to “clearance” when used in appropriate contexts.

Note 2.— The abbreviated term “clearance” may be prefixed by the words “taxi”, “take-off”, “departure”, “en-route”, “approach” or “landing” to indicate the particular portion of flight to which the air traffic control clearance relates.

Air traffic control (ATC) instruction. Directives issued by air traffic control for the purposes of requiring a pilot to take specific action.

Air traffic control (ATC) service. A service provided for the purposes of:

- a) preventing collisions:
 - 1) between aircraft, and
 - 2) on the manoeuvring area between aircraft and obstructions; and
- b) expediting and maintaining an orderly flow of traffic.

Air traffic services (ATS). A generic term meaning variously, flight information service, alerting service, air traffic advisory service, air traffic control service (area control service, approach control service or aerodrome control service).

Air user (air-user). The abstract part of the aircraft system that performs the non communication related functions of the application.

AMHS. The symbol used to designate ATS message handling system.

AMHS management domain. An AMHS management domain formed by an ATS organization for the management of that part of the AMHS which is under its responsibility.

AMHS message. An instance of the category of information object defined as message in ISO/IEC 10021-2 and conveyed in the AMHS. It is composed of an envelope and of a content.

AMHS probe. An instance of the category of information object defined as probe in ISO/IEC 10021-2 and conveyed in the AMHS. It is a class of message containing only an envelope which is conveyed by the message transfer agents (MTAs) from one user up to the MTA serving other users, used to determine the deliverability of messages.

AMHS report. An instance of the category of information object defined as report in ISO/IEC 10021-2 and conveyed in the AMHS. It is generated by a message transfer agent (MTA) in order to report on the outcome or progress of a message or probe in the set of interconnected MTAs pertaining to the AMHS.

Application. The ultimate use of an information system, as distinguished from the system itself.

Application entity (AE). Part of an application process that is concerned with communication within the OSI environment. The aspects of an application process that need to be taken into account for the purposes of OSI are represented by one or more AEs.

Application entity (AE) qualifier. That part of the AE title that unambiguously identifies the particular application entity.

Application entity (AE) service interface. The interface between the application users and the application service provider.

Application entity (AE) title. An unambiguous name for an application entity.

Application layer. The seventh layer of the OSI reference model that controls application user access to the communication system and provides services to perform a logical association to other applications.

Application layer structure (ALS). The application layer structure refers to the internal architecture of the OSI application layer as described in ISO/IEC 9545.

Application process (AP). A set of resources, including processing resources, within a real open system which may be used to perform a particular information processing activity.

Application protocol data unit (APDU). An Application protocol data unit is an (N) PDU where N refers to the application layer. An APDU is the basic unit of information exchanged between the airborne application and the ground application.

Application service. The abstract interface between the (N) service and the (N) service user, where N refers to the application layer; thus it is the boundary between the AE and the application user.

Application service element (ASE). The element in the communication system which executes the application specific protocol. In other words, it processes the application specific service primitive sequencing actions, message creation, timer management, error and exception handling. The application's ASE interfaces only with the application's CF.

Application service element (ASE) service interface. The abstract interface through which the ASE service is accessed.

Note.— In version 1 of the ADS application, the ADS-ASE service interface coincides with the ADS-AE abstract service interface.

Application service object (ASO). An active element within (or equivalent to the whole of) the application-entity embodying a set of capabilities defined for the application layer that corresponds to a specific ASO-type (without any extra capabilities being used). An ASO is a combination of application service elements (ASEs) and ASOs that perform a specific function. An ASO that provides the functions of the establishment and data transfer phases is considered a complete protocol.

Application user. That abstract part of the aircraft or ground system that performs the non-communication related functions of the application.

Association control service element (ACSE). The association control service element is the common mechanism in the application layer structure (ALS) for establishing and releasing application service object (ASO) associations.

ATIS. The automatic provision of current, routine information to arriving and departing aircraft throughout 24 hours or a specified portion thereof:

Data link — automatic terminal information service (D-ATIS). The provision of ATIS via data link.

Voice — ATIS (Voice -ATIS). The provision of ATIS by means of continuous and repetitive voice broadcast.

ATIS application. An ATN application that supports the ATIS.

ATN. The symbol used to designate the aeronautical telecommunication network.

ATN application. Refers to an application that is designed to operate over ATN communication services.

ATN communication services. Composed of the internet communications service and the upper layers communications service.

ATN environment. The environment that relates to functional and operational aspects of the ATN as a complete end-to-end communication system.

ATN profile requirement list (APRL). APRLs identify, in a tabular form, requirements together with the options and parameters for protocols used in the ATN. The supplier of an ATN protocol implementation claiming to conform to the ATN technical requirements must indicate conformance to those requirements by preparing a protocol implementation conformance statement (PICS) based on the set of APRLs.

ATS. The symbol used to designate air traffic services.

ATSC. The symbol used to designate air traffic services communication.

ATSC class. The ATSC class parameter enables the ATSC user to specify the quality of service expected for the offered data. The ATSC class value is specified in terms of ATN end-to-end transit delay at 95% probability.

ATS communication (ATSC). Communication related to air traffic services including air traffic control, aeronautical and meteorological information, position reporting and services related to safety and regularity of flight. This communication involves one or more air traffic service administrations. This term is used for purposes of address administration.

ATS interfacility data communication (AIDC). Automated data exchange between air traffic services units, particularly in regard to co-ordination and transfer of flights.

AIDC application. An ATN application dedicated to exchanges between ATS units (ATSUs) of air traffic control (ATC) information in support of flight notification, flight coordination, transfer of control, transfer of communication, transfer of surveillance data and transfer of general data.

ATS message. A unit of user-data, coded in binary form, which is conveyed from an originator of the data to one or more recipients of the data. It is possible to associate a unique message identifier and a priority with each ATS message.

ATS message handling services (ATSMHS). Procedures used to exchange ATS messages over the ATN such that the conveyance of an ATS message is in general not correlated with the conveyance of another ATS message by the service provider. There are two ATS message handling services. They are the ATS message service and the ATN pass-through service.

ATS message protocol stack Type A. The protocol implemented between two ATN end systems which support the ATN pass-through service.

ATS message server. An ATN end system which provides the relay function included in the ATS message service. It may also optionally provide the storage function included in the ATS message service.

ATS message handling system (AMHS). The set of computing and communication resources implemented by ATS organizations to provide the ATS message service.

ATS message user agent. An ATN end system which provides an interface to the ATS message service for an ATS message service user.

ATSMHS. The symbol used to designate ATS message handling services.

ATS organization. An ICAO State or organization which administers one or more ATS end and/or intermediate systems.

ATS unit (ATSU). A generic term meaning variously, air traffic control unit, flight information centre or air traffic services reporting office.

Authorized path. A communication path that the administrator(s) of the routing domain(s) has pre-defined as suitable for a given traffic type and category.

Automatic dependent surveillance (ADS). A surveillance technique in which aircraft automatically provide, via a data link, data derived from on-board navigation and position-fixing systems, including aircraft identification, four-dimensional position, and additional data as appropriate.

Automatic terminal information service (ATIS). The provision of current, routine information to arriving and departing aircraft throughout the day or a specified portion of the day, via a data link or a continuous and repetitive voice broadcast.

Boundary intermediate system (BIS). An intermediate system that is able to relay data between two separate routing or administrative domains.

CIDIN. The symbol used to designate common ICAO data interchange network.

CM. The symbol used to designate context management.

Connectionless network protocol (CLNP). The protocol responsible for forwarding packets through the ATN internet communications service.

Context management (CM) application. An ATN application that provides a logon service allowing initial aircraft introduction into the ATN and a directory of all other data link applications on the aircraft. It also includes functionality to forward addresses between ATS units.

Note.— Context management is a recognized OSI presentation layer term. The OSI use and the ATN use have nothing in common.

Controller pilot communication (CPC). In a controlled airspace, continuous listening watch on the appropriate radio frequency (either manual or automatic with signaling devices) and establishment of two-way communication with the appropriate air traffic control (ATC) unit.

Controller pilot data link communication (CPDLC). A means of communication between controller and pilot, using data link for ATC communications.

CPDLC application. An ATN application that provides a means of ATC data communication between controlling, receiving or downstream ATS units and the aircraft, using air-ground and ground-ground subnetworks, and which is consistent with the ICAO phraseology for the current ATC voice communication.

Control function (CF). That abstract part of the AE that performs the mapping between the ASE service primitives, the association control service element (ACSE) service primitives and other elements within the application entity.

Controlling ATSU (C-ATSU). The air traffic control unit exercising legal authority over the initiation, continuation, diversion or termination of flights and providing air traffic control service to controlled flights in the control area under its jurisdiction.

CPDLC. The symbol used to designate controller pilot data link communication.

Current data authority. The ground system that provides for the establishment and maintenance of a transport connection for the purposes of conducting a CPDLC dialogue pertaining to the services of the C-ATSU.

Data authority. A ground system that provides for the establishment and maintenance of a CPDLC transport connection with an aircraft. The transfer of communication from the current data authority to the next data authority is prepared prior to the actual data link switch by designating a next data authority in a specific CPDLC message.

Data communications equipment (DCE). An interface between data terminal equipment and the transmission mechanism.

Data link layer. The second layer of the OSI reference model that manages the operations of the physical layer and may utilize special error detection or retransmission techniques to achieve acceptable error rates.

Demand contract (DC). A contract between a requestor and a provider of information service, such as ADS or FIS, to provide a single report to the requestor (vs. Continual reports to one request).

Dialogue. A co-operative relationship between elements which enables communication and joint operation.

Dialogue service (DS). The lower service boundary of an ASE; the service allows two ASEs to communicate, such as a CM ground-ASE to communicate with a CM air-ASE.

Directory. A facility that supports on request the retrieval of address information or the resolution of application names.

Distinguishing path attribute (DPA). Used to discriminate among multiple routes to a destination, based on differences in the quality of service between the routes (for example, expense, transit delay or residual error probability.)

Domain. A set of end systems and intermediate systems that operate according to the same routing procedures and that is wholly contained within a single administrative domain.

Domain specific part (DSP). An addressing authority is responsible for its own addressing subdomain and network service access point (NSAP) addresses within that addressing domain are distinguished, where necessary, by the value of the DSP.

Downstream ATSU (D-ATSU). D-ATSU handles the coordination of the conditions of transfer for a flight from the controlling ATSU (C-ATSU) which may notify the D-ATSU of a flight's cleared profile prior to its effective transfer to the receiving ATSU (R-ATSU).

Downstream clearance (DSC). Specific clearance request by an aircraft to an ATSU which is not the controlling ATSU. The initiation of the DSC service can only be initiated by an aircraft.

Downstream data authority. The ground system that is permitted to conduct a downstream CPDLC downstream clearance (DSC) dialogue with an aircraft.

DSC. The symbol used to designate downstream clearance.

Emergency contract. A contract to provide ADS reports at regular intervals during an emergency situation.

End routing domain (ERD). A routing domain (RD) that only routes protocol data units (PDUs) from/to its own RD.

End system (ES). A system that contains the OSI seven layers and contains one or more end user application processes.

End-to-end. Pertaining or relating to an entire communication path, typically from (1) the interface between the information source and the communication system at the transmitting end to (2) the interface between the communication system and the information user or processor or application at the receiving end.

End user. An ultimate source and/or consumer of information.

Entity. An active element in any layer which can either be a software entity (such as a process) or a hardware entity (such as an intelligent I/O chip).

Estimated time of arrival (ETA). For IFR flights, the time at which it is estimated that the aircraft will arrive over that designated point, defined by reference to navigation aids, from which it is intended that an instrument approach procedure will be commenced, or if no navigation aid is associated with the aerodrome, the time at which the aircraft will arrive over the aerodrome. For VFR flights, the time at which it is estimated that the aircraft will arrive over the aerodrome.

Ethernet. Based on the local area network standard, ISO/IEC 8802-3, carrier sense multiple access with collision detection (CSMA/CD) access method, and physical layer specifications using broadcast technology which may connect as an ATN subnetwork.

Expense. The cost to perform some task. In the context of internetworking, expense is defined in terms of the incremental expense incurred for transfer of a single network service data unit (NSDU) of 512 octets in size.

Extended projected profile. A projected profile extended up to a number of way points.

Fast byte. The capability at any layer of the OSI reference model to negotiate out the capabilities of the base protocol.

Figure of merit (FOM). An indication of the level of accuracy of positional information given in an ADS report.

FIS. The symbol used to designate flight information service.

FIS application. An ATN application that provides to aircraft information and advice useful for safe and efficient conduct of flight.

FIS contract. An agreement between a FIS air-user and a FIS ground-user that the latter will provide FIS reports under the conditions specified in the FIS contract.

Flight information region (FIR). An airspace of defined dimensions within which flight information service and alerting service are provided.

Flight information service (FIS). A service provided for the purpose of giving advice and information useful for the safe and efficient conduct of flights.

Flight plan. Specified information provided to air traffic services units, relative to an intended flight or portion of a flight of an aircraft.

Note.— Specifications for flight plans are contained in Annex 2. A model Flight Plan Form is contained in Appendix 2 to PANS-RAC (Doc 4444).

Flow control. A function that controls the flow of data to perform buffer management within a layer or between adjacent layers.

Forward contract. A contract to provide a ground ADS system with ADS reports.

Forwarding information base (FIB). The information base that is maintained by each router and contains the set of forwarding paths reflecting the various policy and QoS rankings available to reach each known destination.

Function. A coherent set of activities which fulfils, by itself or together with other functionality, a concept. Examples of functions: conflict free planning; electronic representation of the flight.

Functional requirements. Requirements that determine what function a system should perform. They can usually be expressed by a verb applying to a type of data, e.g., display aircraft position.

Gateway. A system used to interconnect dissimilar networks. A gateway may contain all seven layers of the OSI reference model.

General communication. A category of communications which includes APC, public correspondence and other non-operational and non-administrative communication.

Ground application service element (ground-ASE). An abstract part of the ground system that performs the communication related functions of the application.

Ground user (ground-user). The abstract part of the ground system that performs the non-communication related functions of the application.

Ground earth station (GES). An earth station in the fixed satellite service, or, in some cases, in the aeronautical mobile-satellite service, located at a specified fixed point on land to provide a feeder link for the aeronautical mobile-satellite service.

Note.— This definition is used in the ITU's Radio Regulations under the term "aeronautical earth station." The definition herein as "GES" for use in the SARPs is to clearly distinguish it from an aircraft earth station (AES), which is a mobile station on an aircraft.

Ground forwarding function. The capability for a ground system to forward a CPDLC message to another ground system via a CPDLC message with an indication of success, failure or non-support from the receiving ground system. This function may be invoked by the current data authority in order to avoid retransmission of a request by an aircraft by forwarding the information to the next data authority. The

downstream data authority may use this function in order to relay a message to the current data authority which then performs the actual transmission to the aircraft.

Ground-ground application. An application that has one both of its peer applications on the ground.

ICAO Facility Designator (ICAO AFTN Addressee Indicator). An eight-letter code group formulated in accordance with rules prescribed by ICAO and assigned to the ATS end system executing an application process.

ICC. The symbol used to designate inter-centre communication.

ICS. The symbol used to designate the internet communication services.

Initial domain part (IDP). The addressing authority responsible for an addressing subdomain that assigned the network service access point (NSAP) address and that specified the abstract syntax and structure of the remainder of the NSAP address.

Inter-centre communication (ICC). ICC is data communication between ATS units to support ATS, such as notification, coordination, transfer of control, flight planning, airspace management and air traffic flow management.

Intermediate system (IS). A system which performs relaying and routing function and comprises the lowest three layers of the OSI reference model.

International Alphabet No. 5 (IA5). International Alphabet Number 5 defined by ITU-T.

Note.— ATN uses the “6 bit ASCII” subset of IA5, as used in SSR Mode S specifications.

Internet communications service (ICS). The internet communications service is an internetwork architecture which allows ground, air-to-ground and avionics data subnetworks to interoperate by adopting common interface services and protocols based on the ISO OSI reference model.

Internetwork. A set of interconnected, logically independent heterogeneous subnetworks. The constituent subnetworks are usually administrated separately and may employ different transmission media.

Internetwork protocol (IP). A protocol that performs the basic end-to-end mechanism for the transfer of data packets between network entities. In the ATN internet communications service, the ISO/IEC 8473 internetwork protocol is used.

Interoperability. Describes the ability of the ATN to provide, as a minimum, a transparent data transfer service between end systems even though the ATN comprises various ground, air-to-ground and avionics subnetworks. The ability to interoperate between end systems can be extended to include commonality of upper layer protocols.

ISO. The symbol used to designate International Organization for Standardization.

ITU-T. The symbol used to designate International Telecommunication Union-Telecommunication Standardization Sector.

IETF. The symbol used to designate Internet Engineering Task Force.

Long transport service access point (TSAP). Composed of the router domain part (RDP) and the short TSAP.

Lower layers. The physical, data link, network and transport layers of the OSI reference model.

Managed object. Data processing and data communication resources that may be managed through the use of the OSI management protocol.

Management agent. Performs management operations on managed objects within its local environment as a consequence of management operations communicated from a manager. A management agent may also forward notifications emitted by managed objects to a manager.

Management domain (MD). Resources that for systems management purposes are represented by managed objects. A management domain possesses at least the following quantities: a name that uniquely identifies that management domain, identification of a collection of managed objects that are members of the domain and identification of any inter-domain relationships between this domain and other domains.

Manager. The term given to a system that requests or otherwise receives information about managed objects.

Message. Basic unit of user information exchanged between an airborne application and its ground counterpart or between two ground applications. Messages are passed in one or more data blocks from one end user to another through different subnetworks.

Message element. A component of a message used to define the context of the information exchanged.

Message element identifier. The ASN.1 tag of the ATCUplinkMsgElementId or the ATCDownlinkMsgElementId.

Message handling system (MHS)-form address. An instance of the AMHS address form which is used to locate a direct or indirect AMHS user in the AMHS address space.

Message header. The control information used to maintain synchronization between the two end systems.

Mobile routing domains. Formed from ATSC and AINSC systems onboard an aircraft (or any other mobile platform), within the aircraft operator's administrative domain. A mobile RD is characterized as an end routing domain (ERD).

Mobile subnetwork. A subnetwork connecting a mobile system with another system not resident in the same mobile platform. These subnetworks tend to use free-radiating media (e.g. VHF/UHF radio, D band satellite or D band secondary surveillance radar) rather than contained media (e.g. wire or coaxial cable); thus they exhibit broadcast capabilities in the truest sense.

Mode select (Mode S). An enhanced mode of secondary surveillance radar (SSR) which permits the selective interrogation of Mode S transponders, the two-way exchange of digital data between Mode S interrogators and transponders and also the interrogation of Mode A or Mode C transponders.

Naming plan. A plan that provides common naming conventions and designations for the unambiguous identification of all end and intermediate systems in accordance with the rules prescribed in ISO/IEC 7498-3, ISO/IEC TR 10730 and ISO/IEC 9545.

Network addressing domain. A subset of the global addressing domain consisting of all the NSAP addresses allocated by one or more addressing authorities.

Network entity (NE). A functional portion of an internetwork router or host computer that is responsible for the operation of internetwork data transfer, routing information exchange and network layer management protocols.

Network entity title (NET). The global address of a network entity.

Network layer (NL). Provides a uniform service interface for the transfer of data among end systems and intermediate systems (ISs) utilizing the ISO protocol architecture.

Network management (NM). The set of functions related to the management of various OSI resources and their status across the Network Layer of the OSI architecture.

Network service access point (NSAP). Point within the ISO protocol architecture at which global end users may be uniquely addressed on an end-to-end basis.

Network service access point (NSAP) address. A hierarchically organized global address, supporting international, geographical and telephony-oriented formats by way of an address format identifier located within the protocol header. Although the top level of the NSAP address hierarchy is internationally administered by ISO, subordinate address domains are administered by appropriate local organizations.

Network service access point (NSAP) address prefix. Used to identify groups of systems that reside in a given routing domain or confederation. An NSAP prefix may have a length that is either smaller than or the same size as the base NSAP address.

Network topology map. Provides an overall view of the global network connectivity and is used in path computations by the operative routing algorithm.

Next data authority. The ground system that provides for the establishment and maintenance of a transport connection for the purposes of conducting a CPDLC dialogue pertaining to the services of the receiving ATS unit (R-ATSU).

NOTAM. A notice containing information concerning the establishment, condition or change in any aeronautical facility, service, procedure or hazard, the timely knowledge of which is essential to personnel concerned with flight operations.

Open systems interconnection (OSI) protocol architecture. A set of protocols used to implement the OSI reference model.

Open systems interconnection (OSI) reference model. A model providing a standard approach to network design introducing modularity by dividing the complex set of functions into seven more manageable, self-contained, functional layers. By convention these are usually depicted as a vertical stack.

Operational requirement. A statement of the operational attributes of a system needed for the effective and/or efficient provision of air traffic services to users.

OSI. The symbol used to designate open systems interconnection.

Packed encoding rules (PER). Encoding rules as defined in ISO/IEC 8825-2 which have been designed to minimize the number of bits transmitted.

Performance management. Enables the behavior of resources and the effectiveness of communication activities to be evaluated. Includes functions to gather statistical information, maintain and examine logs of system state histories, determine system performance under natural and artificial conditions and alter system modes of operation.

Performance requirements. Requirements that define a function's characteristics, such as reliability, availability, response time, processing delay, integrity, that are necessary to meet the operational requirements for a specific application of the function.

Periodic contract (PC). A contract to provide ADS reports at regular intervals.

Physical layer. The layer of the OSI reference model that controls access to the transmission medium which forms the basis for the communication system.

Presentation address (PA). The presentation address must, as a minimum, include a network service access point (NSAP) address and a transport service access point (TSAP) selector and may include a presentation service access point (PSAP) selector and session service access point (SSAP) selector based on the addressing structure adopted within the end system (ES) and whether the application requires the OSI session or presentation protocol.

Presentation data value (PDV). The unit of information specified in an abstract syntax, which is transferred by the OSI presentation-service (ISO/IEC 8822).

Presentation layer. The layer of the OSI reference model that controls the coding, format and appearance of the data transferred to and from the application layer.

Presentation service access point (PSAP) selector. The element of the presentation address that identifies the user of the presentation protocol entity.

Priority (P). The relative importance of a particular protocol data unit (PDU) relative to other PDUs in transit and used to allocate resources which become scarce during the transfer process.

Profile. Defines implementation conformance constraints on a set of reference specifications.

Projected profile. An indication of where and when the aircraft anticipates it will be at the following two way-points.

Protocol. A set of rules and formats (semantic and syntactic) which determines the communication behavior between peer entities in the performance of functions at that layer.

Protocol control information (PCI). Information included in a layer header which contains service primitives specific to that layer.

Protocol data unit (PDU). (1) A unit of data transferred between peer entities within a protocol layer consisting of protocol control information and higher layer user data (i.e. service data units). (2) A unit of data specified in an (N) protocol and consisting of (N) protocol control information and possibly (N) user data, where N indicates the layer.

Protocol implementation conformance statement (PICS). A protocol implementation conformance statement enables conformance testing of protocols. As recommended by ISO/IEC 9646-2, PICS proforma, tailored to ATN context, have been developed as ATN profile requirement list (APRLs) to provide an effective basis for conformance testing of implementations.

Quality of service (QoS). Information relating to data transfer characteristics (for example, requested throughput and priority) used by a router to perform relaying and routing operations across the subnetworks which make up a network.

RFC. The symbol to designate Request for Comments.

Receiving ATSU (R-ATSU). The next air traffic control unit which is the process of accepting the control authority and communication responsibility for a flight transferred by the controlling ATSU (C-ATSU).

Relaying. The process of transferring packets across subnetworks including any necessary packet conversion.

Requested QoS. The service characteristics desired by the service user.

Reserved value. Legal values for the respective fields (have not yet been assigned specific meanings by ICAO). These values should be processed normally in order to allow future assignment. Meanings may be assigned in the future and are not available for local use. The allocation of these values requires no change in the version identifier.

Residual error probability. Indicates the likelihood that a protocol data unit (PDU) will be lost, duplicated or corrupted. This probability is defined as the ratio of lost, duplicated or corrupted network service data units (NSDUs) to the total number of NSDUs transmitted by an ATN network service (NS) provider, normalized for an NSDU size of 512 octets.

Residual error rate (RER). The ratio of messages mis-delivered, non-delivered or delivered with an error undetected by the system, to the total number of messages delivered to the system during a measurement period (adapted from ISO/IEC 8072).

Note.— For the ATN, detected mis-delivered and non-delivered messages are not included in the ratio.

Route. The set of addresses that identifies the destinations reachable over the router and information about the route's path including the QoS and security available over the route.

Router. The communication element that manages the relaying and routing of data while in transit from an originating end system to a destination end system. A router comprises an OSI intermediate system and end system supporting a systems management agent.

Routing. A function within a layer that uses the address to which an entity is attached in order to define a path by which that entity can be reached.

Routing area (RA). A routing subdomain comprising one or more intermediate systems (ISs) and optionally one or more end systems (ESs).

Routing domain (RD). A set of end systems and intermediate systems that operate the same routing protocols and procedures and that are wholly contained within a single administrative domain. A routing domain may be divided into multiple routing subdomains.

Routing domain confederation (RDC). A set of routing domains and/or RDCs that have agreed to join together. The formation of a RDC is done by private arrangement between its members without any need for global coordination.

Routing domain identifier (RDI). A generic network entity title (NET) as described in ISO/IEC 7498 and is assigned statically in accordance with ISO/IEC 8348. An RDI is not an address and cannot be used as a valid destination of an ISO/IEC 8473 PDU. However, RDIs are, like ordinary NETs, assigned from the same addressing domain as network service access point (NSAP) addresses.

Routing information base (RIB). A data base that is maintained by each router and comprises the information regarding the connectivity and topology of the end systems (ESs) and intermediate systems (ISs) within a particular routing domain and path information pertinent to paths interconnecting routing domains. It is maintained by way of the information received by a routing information exchange protocol. Each routing information exchange protocol has its own RIB specification.

Routing information exchange protocol. The protocol used to exchange subnetwork connectivity information between end systems and intermediate systems and between intermediate systems and intermediate systems.

Routing policy. A set of rules that control the selection of routes and the distribution of routing information by boundary intermediate systems (BISs). These rules are based on policy criteria rather than on performance metrics such as hop count, capacity, transit delay, cost, etc. which are usually applied for routing. There are two groups of routing policy in the ATN:

- a) general routing policy to ensure necessary connectivity at a reasonable routing information update rate, and
- b) user specified routing policy, i.e. individual policy rules which may be additionally implemented in BISs by administrations and organizations to meet their specific operational and policy needs.

Runway visual range (RVR). The range over which the pilot of an aircraft on the centre line of a runway can see the runway surface markings or the lights delineating the runway or identifying its centre line.

Secondary surveillance radar (SSR). A surveillance radar system which uses transmitters/receivers (interrogators) and transponders.

Security label. May indicate requirements for protection of a protocol data unit (PDU) and provide information used by network layer access control functions.

Service data unit (SDU). A unit of data transferred between adjacent layer entities, which is encapsulated within a protocol data unit (PDU) for transfer to a peer layer.

Service primitive. A function of an application service element (ASE) that is not broken down further into subfunctions and is presented as part of the abstract service interface (i.e. request, indication, response or confirmation).

Service provider. The ground and airborne application entities (AEs) for the application, all underlying data communication protocol entities and the physical media. As a consequence, it encompasses everything between the application-AE service interfaces of the end users of the application.

Session layer. The layer of the OSI reference model that establishes the rules of dialogue between two end user entities.

Session service access point (SSAP) selector. The element of the session address that identifies the user of the session protocol entity.

Short transport service access point (TSAP). Composed of the administrative region selector (ARS), (Optional), the location identifier (LOC), the system identifier (SYS), the network selector (SEL), and the transport selector (TSAP selector).

Stack (or protocol stack). A set of cooperating OSI protocols selected from different layers of the basic reference model. Hence, upper layer stack refers to session, presentation and application protocols, while lower layer stack refers to physical, data link, network and transport protocols.

Subnetwork (SN). An actual implementation of a data network that employs a homogeneous protocol and addressing plan and is under control of a single authority.

Subnetwork access protocol (SNACp). The actual protocol used to receive services for a particular sub-network. For example, the subnetwork access protocol to many public data networks is X.25.

Subnetwork dependent convergence function (SND CF). The set of rules and procedures needed to convert the data transfer needs of the subnetwork independent convergence protocol to the actual services provided by a subnetwork.

Subnetwork (SN) domain. The set of end systems and intermediate systems connected to the same physical network.

Subnetwork independent convergence function (SNICF). The common protocol for all host computers and routers that is used for the transfer of data. The SNICF is the connectionless network protocol defined by ISO/IEC 8473.

Subnetwork point of attachment (SNPA). The point at which a real end system, interworking unit or real subnetwork is attached to a real subnetwork and is a conceptual point within an end or intermediate system at which the subnetwork service is offered.

Subnetwork point of attachment (SNPA) address. Provides information used in the context of a particular real subnetwork to identify a SNPA. An SNPA address is a subnetwork address such as X.25 data terminal equipment (DTE) addresses, ethernet MAC addresses, etc.

Subset. An implementation of an application air or ground service conforming to the application SARPs which supports a defined, technically acceptable but not complete application functionality.

Subsetting rules. Formal instructions relating to the requirement for combinations of elements within an application SARPs, constituting limited application functionality.

System application. An application supports the operation of the air-ground applications, ground-ground applications, or communication services. A system application can take the form of either an air-ground application or a ground-ground application.

System level requirement. The system level requirement is a high-level technical requirement that has been derived from operational requirements, technological constraints and regulatory constraints (administrative and institutional). The system-level requirements are the basis for the functional requirements and lower level requirements.

Systems management (SM). ATN systems management gives deterministic and controllable behaviour in support of the required communications service levels by providing facilities to control, co-ordinate and monitor the resources which allow communications to take place in the ATN environment. These facilities include fault management, accounting management, configuration management, performance management and security management.

Traffic category. A subdivision of the operational communication traffic type used to distinguish between ATS communication and aeronautical operational control (AOC).

Traffic type. A means used to distinguish different types of message traffic for the purposes of establishing communication paths to support operational and legal requirements. There are four traffic types:

- a) the operational communication traffic type is subdivided into two categories representing safety and regularity of flight communication:
 - 1) ATS communication
 - 2) Aeronautical operational control
- b) administrative communication representing non-safety and regularity of flight communication sent by aircraft operating agencies and ATS administrations
- c) general communication, representing APC, public correspondence and other non-operational and non administrative communication, and

- d) systems management communication representing systems management information that is critical for support of network operations.

Note.— The differentiation of traffic types is required because different data traffic may have different access to subnetworks. The traffic type is conveyed in the ATN security label of ISO/IEC 8473 and ISO/IEC 10747. It is used to qualify connectionless mode network protocol (CLNP) data packets and (inter-domain) routes according to the class of traffic that they carry. Based on this qualification, access of subnetworks is controlled by the ATN internet communications service.

Transit delay. In packet data systems, the elapsed time between a request to transmit an assembled data packet and an indication at the receiving end that the corresponding packet has been received and is ready to be used or forwarded.

Transit routing domain (TRD). A domain whose policies permit its boundary intermediate systems (BISs) to provide relaying for protocol data units (PDUs) whose source is located in either the local routing domain or in a different routing domain.

Transport layer. The fourth layer of the OSI reference model which ensures that the data are reliably delivered to the correct destination regardless of which network layer protocol and underlying subnetworks are being used.

Transport protocol class 4 (TP-4). Transport protocol class 4 is defined in ISO/IEC 8073 and profiled for ATN context to provide the connection mode transport service as described in ISO/IEC 8072.

Transport service access point (TSAP). The logical access point to the transport layer.

Transport service access point (TSAP) address. The complete communication address which unambiguously defines a transport service user. The TSAP address comprises the NSAP address and a TSAP selector.

Transport service data unit (TSDU). The data presented to the transport layer for transmission over the ATN internet communications service.

Update contract (UC). A contract to provide a piece of FIS information and any update of this information.

Upper layer (UL) communications service. A term pertaining to the session, presentation and application layers of the OSI reference model.

User. That abstract part of the aircraft or ground system that performs the non-communication related functions of the application. The direct user of the ATN is an application within an end system supporting ATS or aeronautical industry services. The air traffic controller, other ground staff or the pilot are users of the ATN. The user may also be seen more on the abstract level as an organization, e.g. airline or service provider

User requirements. Requirements that are allocated to the user to ensure the interoperability of the communication services and application entities.

UTC. The symbol used to designate coordinated universal time.

Very high frequency (VHF) digital link (VDL). Packet data communication to aircraft and ground users comprised of airborne VHF data radios (VDRs), VHF ground stations and connectivity to routers on the aircraft and the ground.

X.25 packet switched data network (PSDN). A communication network that provides a network access service in compliance with CCITT recommendation X.25.

1.1.2 REFERENCES

When the following reference designators are cited in the Standards and Recommended Practices (SARPs) for the ATN they are referring to the following editions and/or versions:

Note 1.— The cited references were used in the preparation of Doc 9705. In the course of the normal progression of ISO and ITU-T standards, new editions are released. New editions to the referenced documents can be safely used in place of the referenced documents with the understanding that new functions introduced in those editions might not be supported by other implementations. Additionally, Amendments to ISO standards are incorporated into the following editions of the base standard and therefore information can be found there.

Note 2.— New versions of the ISO standards are issued whenever backwards compatibility is not ensured. For that reason use of different versions of ISO and ITU standards from those cited is undesirable.

CCITT Rec X.121 (1992). International numbering plan for public data networks.

CCITT Rec X.400 (1992). Message handling system and service overview.

CCITT Rec X.402 (1992). Message handling systems: Overall architecture.

CCITT Rec X.408 (1988). Message handling systems: Encoded information type conversion rules.

CCITT Rec X.411 (1992). Message handling systems: Message transfer system: Abstract service definition and procedures.

CCITT Rec X.413 (1992). Message handling systems: Message store: Abstract service definition.

CCITT Rec X.419 (1992). Message handling systems: Protocol specifications.

CCITT Rec X.420 (1992). Message handling systems: Interpersonal messaging system.

ISO/IEC 646:1991. Information Technology — ISO 7-bit coded character set for information interchange.

ISO/IEC 3166:1993. Codes for the representation of names and countries.

ISO/IEC 6523:1994. Data interchange — Structures for the identification of organizations (Registration of International Code Designators).

ISO/IEC 7498-1:1994. Information technology — Open Systems Interconnection — Basic Reference Model. Reference: ITU-T Rec. X.200 (1994)

ISO/IEC 7498-2:1989. Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 2: Security Architecture.

ISO/IEC 7498-3:1989. Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 3: Naming and Addressing. Reference: ITU-T Rec. X.650 (1992).

ISO/IEC 7498-4:1989. Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 4: Management framework.

ISO/IEC 8072:1994. Information technology — Open Systems Interconnection — Transport service definition (second edition). Reference: ITU-T Rec. X.214 (1993).

ISO/IEC 8073:1992. Information technology — Telecommunications and information exchange between systems — Open Systems Interconnection — Protocol for providing the connection-mode transport service.

ISO/IEC 8073/PDAM5:1992. Information technology — Telecommunications and information exchange between systems — Open Systems Interconnection — Protocol for providing the connection-mode transport service — Amendment 5: Provision of Non-blocking Expedited Service.

ISO/IEC 8208:1995. Information technology — Data communications — X.25 Packet Layer Protocol for Data Terminal Equipment (Revision of ISO/IEC 8208:1990).

ISO/IEC 8326:1994. Information technology — Open Systems Interconnection — Basic Session Service Definition (second edition). Reference: ITU-T Rec. X.215 (1994).

ISO/IEC 8326:1994/Amd. 1:1997. Information Technology — Open Systems Interconnection — Basic Session Service Definition — Amendment 1: efficiency enhancements.

ISO/IEC 8327-1:1994. Information Technology — Open Systems Interconnection — Basic Connection Oriented Session Protocol: Part 1 — Protocol Specification (second edition). Reference: ITU-T Rec. X.225 (1994).

ISO/IEC 8327-1:1995/Amd. 1:1997. Information Technology — Open Systems Interconnection — Basic Connection Oriented Session Protocol: Part 1 — Protocol Specification — Amendment 1: efficiency enhancements.

ISO/IEC 8327-2:1994. Information technology — Open Systems Interconnection — Basic connection oriented session protocol specification — Part 2: Protocol Implementation Conformance Statement (PICS) Proforma.

ISO/IEC 8348:1993. Information technology — Open Systems Interconnection — Network Service Definition.

ISO/IEC 8473-1:1994. Information technology — Protocol for providing the connectionless-mode network service: Protocol specification.

ISO/IEC 8473-2:1994. Information technology — Protocol for providing the connectionless-mode network service — Part 2: Provision of the underlying service by an ISO/IEC 8802 subnetwork.

ISO/IEC 8473-3:1995. Information technology — Protocol for providing the connectionless-mode network service — Part 3: Provision of the underlying service by an X.25 subnetwork.

ISO/IEC 8473-4:1995. Information technology — Protocol for providing the connectionless-mode network service — Part 4: Provision of the underlying service by a subnetwork that provides the OSI data link service.

ISO/IEC TR 8509:1987. Information processing systems — Open Systems Interconnection — OSI Service Conventions.

ISO/IEC 8602:1995. Information technology — Protocol for providing the OSI connectionless-mode transport service.

ISO/IEC 8648:1988. Information processing systems — Open Systems Interconnection — Internal organization of the Network Layer.

ISO/IEC 8649:1994. Information processing systems — Open Systems Interconnection — Service definition for the Association Control Service Element (second edition). Reference: ITU-T Rec. X.217 (1992).

ISO/IEC 8649/PDAM 1:1995. Information technology — Open Systems Interconnection — Service definition for the Association Control Service Element — Amendment 1: Fast associate mechanism.

ISO/IEC 8650-1:1996. Information processing systems — Open Systems Interconnection — Protocol specification for the Association Control Service Element (second edition). Reference: ITU-T Rec. X.227 (1994).

ISO/IEC 8650-1/DAM 1:1995. Information processing systems — Open Systems Interconnection — Protocol specification for the Association Control Service Element — Amendment 1: Incorporation of extensibility markers.

ISO/IEC 8650-1/DAM 2:1995. Information processing systems — Open Systems Interconnection — Protocol specification for the Association Control Service Element — Amendment 2: Fast associate mechanism.

ISO/IEC 8650-2:1995. Information technology — Open Systems Interconnection — Protocol specification for the Association Control Service Element — Part 2: Protocol Implementation Conformance Statement (PICS) proforma.

ISO/IEC 8802-2:1990. Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part 2: Logical Link Control

ISO/IEC 8802-3:1989. Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part 3: Carrier Sense Multiple Access with Collision Detection — Access Method and Physical Layer Specifications.

ISO/IEC 8822:1994. Information technology — Open Systems Interconnection — Presentation service definition (Second edition). Reference: ITU-T Rec. X.216 (1994).

ISO/IEC 8822:1994/Amd. 1:1997. Information technology — Open Systems Interconnection — Presentation service definition — Amendment 1: efficiency enhancements.

ISO/IEC 8823-1:1994. Information technology — Open Systems Interconnection — Basic connection-oriented presentation protocol — Part 1: Protocol specification (second edition). Reference: ITU-T Rec. X.226 (1994).

ISO/IEC 8823-1:1994/Amd. 1:1997. Information technology — Open Systems Interconnection — Basic connection-oriented presentation protocol — Part 1: Protocol specification — Amendment 1: Efficiency enhancements.

ISO/IEC 8823-2:1995. Information technology — Open Systems Interconnection — Basic connection-oriented presentation protocol — Part 2: Protocol Implementation Conformance Statement (PICS) proforma.

ISO/IEC 8824-1:1994. Information Technology — OSI Abstract Syntax Notation One (ASN.1). — Specification of basic notation. Reference: ITU-T Rec. X.682 (1994).

ISO/IEC 8824-1/Amd.1:1995. Information Technology — Open Systems Interconnection — Abstract Syntax Notation One (ASN.1) — Specification of Basic Notation — Amendment 1: Rules for Extensibility.

ISO/IEC 8824-2:1995. Information technology Open Systems Interconnection Abstract Syntax Notation One (ASN.1) Part 2: Information object specification.

ISO/IEC 8824-3:1995. Information technology Open Systems Interconnection Abstract Syntax Notation One (ASN. 1) Part 3: Constraint specification.

ISO/IEC 8824-4:1995. Information technology Open Systems Interconnection Abstract Syntax Notation One (ASN. 1) Part 4: Parameterization of ASN.1 Specifications.

ISO/IEC 8825-1:1995. Information technology — ASN.1 encoding rules — Part 1: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER) Reference: ITU-T Rec. X.691 (1993).

ISO/IEC 8825-2:1996. Information technology — Open Systems Interconnection — Encoding Rules for Abstract Syntax Notation One (ASN.1) — Part 2: Packed encoding rules. Reference: ITU-T Rec. X.691 (1995).

ISO/IEC 8859-1:1987. Information processing — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1.

ISO/IEC 8878:1992. Information technology — Telecommunications and information exchange between systems — Use of X.25 to provide the OSI Connection-mode Network Service.

ISO/IEC 9542:1988. Information processing systems — Telecommunications and information exchange between systems — End system to Intermediate system routing exchange protocol for use in conjunction with the Protocol for providing the connectionless-mode network service (ISO/IEC 8473).

ISO/IEC 9542/DAM1:1988. Information processing systems — Telecommunications and information exchange between systems — End system to Intermediate system routing exchange protocol for use in

conjunction with the Protocol for providing the connectionless-mode network service (ISO/IEC 8473)—
Amendment 1: Dynamic Discovery of OSI NSAP Addresses by End Systems.

ISO/IEC 9545:1994. Information technology — Open Systems Interconnection — Application Layer structure (second edition).

ISO/IEC 9548-1:1993. Information Technology — Open Systems Interconnection — Connectionless Session Protocol Specification Part 1.

ISO/IEC 9548-2:1995. Information technology — Open Systems Interconnection — Connectionless session protocol specification — Part 2: Protocol implementation conformance statement (PICS) proforma.

ISO/IEC TR 9575:1995. Information technology — Telecommunications and information exchange between systems — OSI Routing Framework.

ISO/IEC 9576-1:1995. Information technology — Open Systems Interconnection — Connectionless presentation protocol specification.

ISO/IEC 9576-2:1995. Information technology — Open Systems Interconnection — Connectionless presentation protocol specification — Part 2: Protocol Implementation Conformance Statement (PICS) proforma.

ISO/IEC TR 9577:1993. Information technology — Telecommunications and information exchange between systems — Protocol identification in the network layer.

ISO/IEC 9595:1991. Information technology — Open Systems Interconnection — Common management information service definition.

ISO/IEC 9596-1:1991. Information technology — Open Systems Interconnection — Common management information protocol — Part 1: Specification.

ISO/IEC 9596-2:1993. Information technology — Open Systems Interconnection — Common management information protocol — Part 2: Protocol Implementation Conformance Statement (PICS) proforma.

ISO/IEC 9646-1:1994. Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 1: General concepts.

ISO/IEC 9646-2:1994. Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 2: Abstract Test Suite specification.

ISO/IEC 9646-4:1994. Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 4: Test realization.

ISO/IEC 9646-5:1994. Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 5: Requirements on test laboratories and clients for the conformance assessment process.

ISO/IEC 9646-7:1995. Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 7: Implementation Conformance Statements.

ISO/IEC 9834-1:1993. Information technology — Open Systems Interconnection — Procedures for the operation of OSI Registration Authorities — Part 1: General procedures.

ISO/IEC 9834-2:1993. Information Technology — Open Systems Interconnection — Procedures for specific OSI Registration Authorities — Part 2: Registration Procedures for OSI Document Types.

ISO/IEC 9834-6:1993. Information technology — Open Systems Interconnection — Procedures for the operation of OSI Registration Authorities — Part 6: Application processes and application entities.

ISO/IEC TR 10000-1:1995. Information technology — Framework and taxonomy of International Standardized Profiles — Part 1: General principles and documentation framework.

ISO/IEC TR 10000-2:1995. Information technology — Framework and taxonomy of International Standardized Profiles — Part 2: Principles and Taxonomy for OSI profiles.

ISO/IEC 10021-1:1990. Information Technology — Text Communication — Message-Oriented Text Interchange System (MOTIS) — Part 1: System and Service Overview.

ISO/IEC 10021-1/Amd.2:1994. Information Technology — Text Communication — Message-Oriented Text Interchange System (MOTIS) — Part 1: System and Service Overview.

ISO/IEC 10021-2:1990. Information Technology — Text Communication — Message-Oriented Text Interchange System (MOTIS) — Part 2: Overall Architecture.

ISO/IEC 10021-2/Amd.1:1993. Information Technology — Text Communication — Message-Oriented Text Interchange System (MOTIS) — Part 2: Overall Architecture.

ISO/IEC 10021-2/Amd.2:1994. Information Technology — Text Communication — Message-Oriented Text Interchange System (MOTIS) — Part 2: Overall Architecture.

ISO/IEC 10021-3:1990. Information Technology — Text Communication — Message-Oriented Text Interchange System (MOTIS) — Part 3: Abstract Service Definition Conventions.

ISO/IEC 10021-4:1990. Information Technology — Text Communication — Message-Oriented Text Interchange System (MOTIS) — Part 4: Message Transfer System: Abstract Service Definition and Procedures.

ISO/IEC 10021-4/Amd. 1:1994. Information Technology — Text Communication — Message-Oriented Text Interchange System (MOTIS) — Part 4: Message Transfer System: Abstract Service Definition and Procedures.

ISO/IEC 10021-5:1990. Information Technology — Text Communication — Message-Oriented Text Interchange System (MOTIS) — Part 5: Message Store: Abstract Service Definition.

ISO/IEC 10021-5/Amd. 1:Date. Message Store Extensions and Message Store Logs.

ISO/IEC 10021-6:1990. Information Technology — Text Communication — Message-Oriented Text Interchange System (MOTIS) — Part 6: Protocol Specifications.

ISO/IEC 10021-7:1990. Information Technology — Text Communication — Message-Oriented Text Interchange System (MOTIS) — Part 7: Interpersonal Messaging System.

ISO/IEC 10028:1993. Information technology — Telecommunications and information exchange between systems — Definition of the relaying functions of a Network layer intermediate system.

ISO/IEC 10035-1:1995. Information Technology Open Systems Interconnection Connectionless ACSE Protocol to Provide the Connectionless Mode ACSE Service.

ISO/IEC 10035-2:1995. Information technology — Open Systems Interconnection — Connectionless ACSE protocol to provide the Connectionless-Mode ACSE service — Part 2: Protocol implementation conformance statement (PICS) proforma.

ISO/IEC 10169-1:1991. Information technology — Open Systems Interconnection — Conformance test suite for the ACSE protocol — Part 1: Test suite structure and test purposes.

ISO/IEC 10589:1992. Information technology — Telecommunications and information exchange between systems — Intermediate system to intermediate system intra-domain-routing routine information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO/IEC 8473).

ISO/IEC ISP 10611-1:1994. Information technology — International Standardized Profiles AMH1n — Message Handling Systems — Common Messaging — Part 1: MHS Service Support.

ISO/IEC ISP 10611-2:1994. Information technology — International Standardized Profiles AMH1n — Message Handling Systems — Common Messaging — Part 2: Specification of ROSE, RTSE, ACSE, Presentation and Session Protocols for use by MHS.

ISO/IEC ISP 10611-3:1994. Information technology — International Standardized Profiles AMH1n — Message Handling Systems — Common Messaging — Part 3: AMH11-Message Transfer (P1).

ISO/IEC ISP 10611-4:1994. Information technology — International Standardized Profiles AMH1n — Message Handling Systems — Common Messaging — Part 4: AMH12-MTS Access (P3).

ISO/IEC ISP 10611-5:1994. Information technology — International Standardized Profiles AMH1n — Message Handling Systems — Common Messaging — Part 5: AMH13-MS Access (P7).

ISO/IEC TR 10730:1993. Information technology — Open systems Interconnection Tutorial on naming and addressing.

ISO/IEC 10731:1994. Information technology — Open Systems Interconnection — Conventions for the definition of OSI services. Reference: ITU-T Rec. X.210 (1993).

ISO/IEC 10747:1994. Information technology — Telecommunications and information exchange between systems — Protocol for exchange of inter-domain routing information among intermediate systems to support forwarding of ISO/IEC 8473 PDUs.

ISO/IEC ISP 11183-1:1992. Information technology — International Standardized Profiles AOM1n OSI Management — Management Communications — Part 1: Specification of ACSE, presentation and session protocols for the use by ROSE and CMISE.

ISO/IEC ISP 11183-2:1992. Information technology — International Standardized Profiles AOM1n OSI Management — Management Communications — Part 2: CMISE/ROSE for AOM12 — Enhanced Management Communications.

ISO/IEC ISP 11183-3:1992. Information technology — International Standardized Profiles AOM1n OSI Management — Management Communications — Part 3: CMISE/ROSE for AOM11 — Basic Management Communications.

ISO/IEC ISP 11188-1:1995. Information Technology — International Standardized Profile — Common upper layer requirements — Part 1: Basic connection oriented requirements.

ISO/IEC 11570:1992. Information technology — Telecommunications and information exchange between systems — Open Systems Interconnection — Transport protocol identification mechanism.

ISO/IEC ISP 12062-1:1995. Information technology — International Standardized Profiles AMH2n — Message Handling Systems — Interpersonal Messaging — Part 1: IPM MHS Service Support.

ISO/IEC ISP 12062-2:1995. Information technology — International Standardized Profiles AMH2n — Message Handling Systems — Interpersonal Messaging — Part 2: AMH21 — IPM Content.

ISO/IEC ISP 12062-3:1995. Information technology — International Standardized Profiles AMH2n — Message Handling Systems — Interpersonal Messaging — Part 3: AMH22 — IPM Requirements for Message Transfer (P1).

ISO/IEC ISP 12062-4:1995. Information technology — International Standardized Profiles AMH2n — Message Handling Systems — Interpersonal Messaging — Part 4: AMH23 — IPM Requirements for MTS Access (P3).

ISO/IEC ISP 12062-5:1995. Information technology — International Standardized Profiles AMH2n — Message Handling Systems — Interpersonal Messaging — Part 5: AMH24 — IPM Requirements for Enhanced MS Access (P7).

ITU-T Rec X.25 (1996). Interface between DTE and DCE terminals Operating in Packet Mode.

ITU-T Rec. X.215 Addendum 1 (1995). Information processing systems — Open Systems Interconnection — Service Definition for Session Layer Efficiency Enhancements.

ITU-T Rec. X.216 Addendum 1 (1995). Information processing systems — Open Systems Interconnection — Service Definition for Presentation Layer Efficiency Enhancements.

ITU-T Rec. X.225 Addendum 1 (1995). Information processing systems — Open Systems Interconnection — Protocol Specification for Session Layer Efficiency Enhancements.

ITU-T Rec. X.226 Addendum 1 (1995). Information processing systems — Open Systems Interconnection — Protocol Specification for Presentation Layer Efficiency Enhancements.

ITU-T Rec X.666 (1995). Procedures for registration of international and multinational organization names.

IETF RFC 1320. The MD4 Message — Digest Algorithm, R. Rivest, April 1992.

1.2 GENERAL

1.2.1 The aeronautical telecommunication network (ATN) shall provide data communication services and application entities in support of:

- a) the delivery of air traffic services (ATS) to aircraft;
- b) the exchange of ATS information between ATS units; and
- c) other applications such as aeronautical operational control (AOC) and aeronautical administrative communication (AAC).

Note 1.— The conceptual model of the ATN is as shown in Figure 1.2.

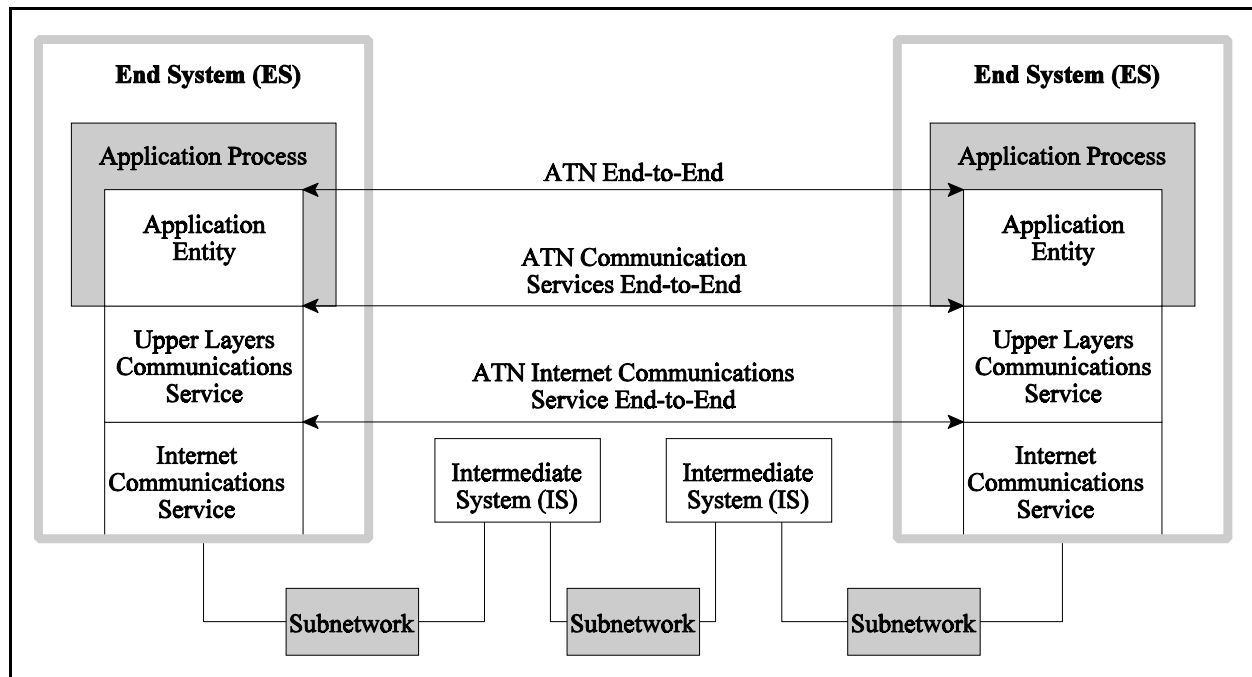
Note 2.— Provisions have been made to accommodate the exchange of information between aircraft operator ground based systems and ATS units, such as weather, flight plans, notices to airmen and dynamic real time air traffic flow management.

Note 3.— Provisions have also been made to accommodate aeronautical passenger communication (APC).

1.2.2 When the aeronautical telecommunication network is used in support of air traffic services, it shall conform with the provisions of this document.

1.2.3 Requirements for use of the ATN shall be made on the basis of regional air navigation agreements.

1.2.4 **Recommendation.**— *Civil aviation authorities should co-ordinate, with national authorities and aeronautical industry, those implementation aspects of the ATN which will permit its world-wide safety, interoperability and efficient use, as appropriate.*



Note 1.— Shading indicates elements outside the scope of these SARPs. User requirements define the interface between the application entity and the user and ensure the functionality and interoperability of the ATN.

Note 2.— The figure represents a simplified model of the ATN and does not depict all of its capabilities (e.g. the store and forward capability which is provided for ATS message handling service).

Note 3.— Various end-to-end points have been defined within the ATN to specify certain end-to-end performance requirements. It may be necessary, however, to define different end-to-end points to facilitate the qualification of implementations to those performance requirements. In such cases, the end-to-end points should be clearly defined and correlated with the end-to-end points shown in the figure.

Figure 1.2. Conceptual model of the ATN

1.3 SYSTEM LEVEL REQUIREMENTS

Note.— The system level requirements are high-level technical requirements that have been derived from operational requirements, technological constraints and regulatory constraints (administrative and institutional). These system-level requirements are the basis for the functional requirements and lower level requirements.

1.3.1 The ATN shall use International Organization for Standardization (ISO) communication standards for open systems interconnection (OSI).

1.3.2 The ATN shall provide a means to facilitate migration to future versions of application entities and/or the communication services.

Note.— It is an objective that the evolution towards future versions facilitates the backward compatibility with previous versions.

1.3.3 The ATN shall enable the transition of existing AFTN users and systems into the ATN architecture.

1.3.4 The ATN shall make provisions whereby only the controlling ATS unit may provide ATC instructions to aircraft operating in its airspace.

Note.— This is achieved through the current and next data authority aspects of the CPDLC application entity.

1.3.5 The ATN shall accommodate routing based on a pre-defined routing policy.

1.3.6 The ATN shall provide means to define data communication that can be carried only over authorized paths for the traffic type and category specified by the user.

1.3.7 The ATN shall offer ATSC classes in accordance with the criteria in Table 1-1.

Table 1-1. Transit delays for ATSC Classes

Maximum One way ATN End-to-End Transit Delay at 95% probability (seconds)	ATSC Class
Reserved	A
4.5	B
7.2	C
13.5	D
18	E
27	F
50	G
100	H
No value specified	no preference
<i>Note 1.— The value for the ATN end-to-end transit delay represents approximately 90% of the value for the total end-to-end transit delay between the ultimate users of the system.</i> <i>Note 2.— The 95% probability is based on the availability of a route conforming to the requested ATSC class.</i>	

Note 1.— When ATSC class is specified by an application process, packets will be forwarded in the ATN internet communications service on a best effort basis. Best effort basis means that when a route is available of the requested ATSC class the packet is forwarded on that route. When no such route is available, the packet will be forwarded on the first known route of ATSC class higher than that requested, or if there is no such route, first known route of lower ATSC class than that requested.

Note 2.— The ATN communication services will not inform application entities if the requested ATSC class was not achieved. It is the responsibility of the application entity to determine the actual transit delay achieved by local means such as time stamping.

1.3.8 The ATN shall operate in accordance with the communication priorities defined in Table 1-2 and Table 1-3.

Table 1-2. Mapping of ATN Communication Priorities

Message Categories	ATN Application	Corresponding Protocol Priority	
		Transport Layer Priority	Network Layer Priority
Network/Systems Management		0	14
Distress Communications		1	13
Urgent Communications		2	12
High Priority Flight Safety Messages	CPDLC, ADS	3	11
Normal Priority Flight Safety Messages	AIDC	4	10
Meteorological Communications		5	9
Flight Regularity Communications	CM, ATSMHS	6	8
Aeronautical Information Service Messages	ATIS	7	7
Network/Systems Administration		8	6
Aeronautical Administrative Messages		9	5
<unassigned>		10	4
Urgent Priority Administrative and U.N. Charter Communications		11	3
High Priority Administrative and State/Government Communications		12	2
Normal Priority Administrative		13	1
Low Priority Administrative		14	0
<i>Note 1.— Priorities above the bold line are for communications related to safety and regularity of flight.</i> <i>Note 2.— The network layer priorities shown in the table apply only to connectionless network priority and do not apply to subnetwork priority.</i>			

Table 1-3. Mapping of ATN Network Priority to Mobile Subnetwork Priority

Message Categories	ATN Network Layer Priority	Corresponding Mobile Subnetwork Priority		
		AMSS	VDL (Mode 1 and Mode 2)	SSR Mode S
Network/Systems Management	14	14	see note 2	high
Distress Communications	13	14	see note 2	high
Urgent Communications	12	14	see note 2	high
High Priority Flight Safety Messages	11	11	see note 2	high
Normal Priority Flight Safety Messages	10	11	see note 2	high
Meteorological Communications	9	8	see note 2	low
Flight Regularity Communications	8	7	see note 2	low
Aeronautical Information Service Messages	7	6	see note 2	low
Network/Systems Administration	6	5	see note 2	low
Aeronautical Administrative Messages	5	5	see note 2	not allowed
<unassigned>	4	not applicable	see note 2	not allowed
Urgent Priority Administrative and U.N. Charter Communications	3	3	see note 2	not allowed
High Priority Administrative and State/Government Communications	2	2	see note 2	not allowed
Normal Priority Administrative	1	1	see note 2	not allowed
Low Priority Administrative	0	0	see note 2	not allowed
<i>Note 1.— Priorities above the bold line are for communications related to safety and regularity of flight.</i> <i>Note 2.— VDL Mode 1 and Mode 2 have no specific subnetwork priority mechanisms.</i> <i>Note 3.— The AMSS SARPs specify mapping of message categories to subnetwork priority without explicitly referencing ATN network layer priority.</i> <i>Note 4.— The term “not allowed” means that only communications related to safety and regularity of flight are authorized to pass over this subnetwork as defined in the subnetwork SARPs.</i>				

1.3.9 The ATN shall enable exchange of application information when one or more authorized paths exist.

1.3.10 The ATN shall notify the appropriate application processes when no authorized path exists.

1.3.11 The ATN shall provide means to unambiguously address all ATN end and intermediate systems.

1.3.12 The ATN shall enable the recipient of a message to identify the originator of that message.

1.3.13 The ATN addressing and naming plans shall allow States and organizations to assign addresses and names within their own administrative domains.

- 1.3.14 The ATN shall support data communication to fixed and mobile systems.
- 1.3.15 The ATN shall accommodate ATN mobile subnetworks as defined in this Annex.
- 1.3.16 The ATN shall make provisions for the efficient use of limited bandwidth subnetworks.
- 1.3.17 The ATN shall enable an aircraft intermediate system to be connected to a ground intermediate system via concurrent mobile subnetworks.
- 1.3.18 The ATN shall enable an aircraft intermediate system to be connected to multiple ground intermediate systems.
- 1.3.19 The ATN shall enable the exchange of address information between application entities.
- 1.3.20 The ATN shall support the context management (CM) application when any of the other air-ground applications are supported.
- 1.3.21 The ATN shall be capable of establishing, maintaining, releasing and aborting peer to peer application associations for the context management (CM) application.
- 1.3.22 The ATN shall be capable of establishing, maintaining, releasing and aborting peer to peer application associations for the automatic dependent surveillance (ADS) application.
- 1.3.23 The ATN shall be capable of establishing, maintaining, releasing and aborting peer to peer application associations for the controller pilot data link communication (CPDLC) application.
- 1.3.24 The ATN shall be capable of establishing, maintaining, releasing and aborting peer to peer application associations for the automatic terminal information service (ATIS) application.
- 1.3.25 The ATN shall be capable of establishing, maintaining, releasing and aborting application associations for the ATS message handling services (ATSMHS) application.
- 1.3.26 The ATN shall be capable of establishing, maintaining, releasing and aborting peer to peer application associations for the ATS interfacility data communication (AIDC) application.
- 1.3.27 Where the absolute time of day is used within the ATN, it shall be based on coordinated universal time (UTC).
- 1.3.28 The end system shall make provisions to ensure that the probability of not detecting a 255-octet message being mis-delivered, non-delivered or corrupted by the internet communications service is less than or equal to 10^{-8} per message.

Note.— It is assumed that ATN subnetworks will ensure data integrity consistent with this system level requirement.

Sub-Volume II

Air-Ground Applications

NOTE ON THE SECOND EDITION

The list below shows the parts of this sub-volume that are different from similar parts of the first edition. It also shows the parts of the first edition that have been deleted and thus no longer appear in this edition.

Reference	Nature of change
2.1.4.2.1	modification
2.1.5.1.1 (Note 4)	modification
Figure 2.1.5-6	modification
Figure 2.1.5-8	modification
Figure 2.1.5-10	modification
Table 2.1.5-1	modification
2.1.5.3.1.2	modification
2.1.5.3.2.2.1	modification
2.1.5.3.3.2.1	modification
2.1.5.3.3.2.2	modification
2.1.5.3.3.4.1	modification
2.1.5.3.3.8.2	modification
2.1.5.4.3.2	modification
2.1.5.4.4.	modification
2.1.5.4.4.1	modification
2.1.5.4.4.2	modification
2.1.5.4.4.3	addition
2.1.5.4.5.2	modification
2.1.5.4.7.1	modification
Table 2.1.5-2	modification
2.1.7.1.1.1	modification
2.1.7.1.1.2	addition
2.1.7.1.1.7	addition
2.1.7.1.2 (Note)	addition
2.1.7.1.3.4	modification
2.1.7.2.2.6	addition
2.1.7.2.2.7	addition
2.1.7.2.3.1 (Note)	addition
2.1.7.2.6.5	modification
Table 2.1.8-2	modification
Table 2.1.8-3	modification
Table 2.1.8-4	modification
Table 2.1.8-5	deletion
Table 2.1.8-5	deletion

2.2.1.1.1 (Note 4)	modification
2.2.1.1.1 (Note 5)	modification
2.2.1.1.1 (Note 6)	modification
2.2.1.1.1 (Note 7)	modification
2.2.1.3.2.1 (Note 5)	modification
Table 2.2.1.3-1	modification
Table 2.2.1.3-2	modification
2.2.1.3.4.3	modification
2.2.1.3.4.3.1	modification
2.2.1.3.4.7 (Note)	modification
2.2.1.3.5 (Note)	modification
Table 2.2.1.3-3	modification
Table 2.2.1.3-4	modification
2.2.1.3.5.3	modification
2.2.1.3.5.3.1	modification
2.2.1.3.5.6.1	modification
Table 2.2.1.3-5	modification
Table 2.2.1.3-6	modification
2.2.1.3.6.3	modification
2.2.1.3.6.3.1	modification
2.2.1.3.7.3.1	modification
2.2.1.3.7.4.1	modification
2.2.1.3.13 (Note)	modification
2.2.1.3.14 (Note)	modification
2.2.1.3.14.2.1	modification
2.2.1.4	modification
2.2.1.4.1	modification
2.2.1.4.1.1	modification
2.2.1.4.1.2	modification
2.2.1.4.2.1	modification
2.2.1.5.1.1 (Note 3)	modification
Table 2.2.1.5-1	modification
2.2.1.5.3.3	modification
2.2.1.5.3.3.2	modification
2.2.1.5.3.6.2.1	modification
Table 2.2.1.5-7	modification
Table 2.2.1.5-8	modification
2.2.15.3.8.2.1	modification
2.2.1.5.3.8.8	modification
2.2.1.5.3.9.5	modification

Table 2.2.1.5-31	modification
Table 2.2.1.5-32	modification
2.2.1.5.3.10.3.1	modification
2.2.1.5.3.10.3.2	modification
2.2.1.5.3.10.4.1	modification
2.2.1.5.3.10.9	modification
Table 2.2.1.5-40	modification
2.2.1.5.3.14.4	modification
Table 2.2.1.5-61	modification
2.2.1.5.3.15.4	modification
2.2.1.5.3.15.4.1	modification
2.2.1.5.3.15.5.1	modification
Table 2.2.1.5-62	modification
2.2.1.5.3.16.4	modification
2.2.1.5.3.16.4.1	modification
2.2.1.5.3.16.5.1	modification
2.2.1.5.4.3.3	modification
2.2.1.5.4.4.2	addition
2.2.1.5.4.5.2	modification
2.2.1.5.4.6.1	modification
2.2.1.5.4.8.1	modification
2.2.1.5.5.1.1 (Note 1)	modification
Table 2.2.1.5-73	modification
2.2.1.7.1.5.3	modification
2.2.1.7.1.5.5	modification
2.2.1.7.2.4.3	modification
2.2.1.7.2.4.4.	addition
2.2.1.7.3.4	modification
2.2.1.7.3.5	modification
2.2.1.7.3.5.1	modification
2.2.1.7.3.5.2	modification
2.2.1.7.3.5.3	modification
2.2.1.7.3.5.4	modification
2.2.1.7.3.5.5	modification
2.2.1.7.3.5.6	modification
2.2.1.7.3.5.7	modification
2.2.1.7.3.5.8	modification
2.2.1.7.3.5.9	modification
2.2.1.7.3.5.10	modification
2.2.1.7.3.5.11	modification
2.2.1.7.3.7.12	modification

2.2.1.7.3.7.13	modification
2.2.1.7.3.7.14 (Note 2)	modification
2.2.1.7.4.1	modification
2.2.1.7.4.1 (Note)	modification
Table 2.2.1.7-1	modification
2.2.1.7.4.4.7	modification
2.2.1.7.4.4.8	addition
2.2.1.7.7.1	modification
2.2.1.7.7.1 (Note)	modification
2.2.1.7.7.1.1	modification
2.2.1.7.7.2.1	modification
2.2.1.7.7.2.1 (Note 2)	modification
2.2.1.7.9 (Note 2)	modification
Table 2.2.1.8-2	modification
Table 2.2.1.8-3	modification
Table 2.2.1.8-4	modification
Table 2.2.1.8-5	deletion
Table 2.2.1.8-6	deletion
2.2.2.1.1	modification
2.2.2.1.1 (Note 2)	modification
2.2.2.3.2 (Note)	modification
2.2.2.3.2.1	modification
2.2.2.3.8.2 (Note)	modification
2.2.2.5.2.1	modification
2.2.2.5.3.2	modification
Table 2.2.2.5-2	modification
2.2.2.5.3.4.10.1	modification
2.2.2.5.3.4.11.1	modification
2.2.2.5.3.5.3.1	modification
2.2.2.5.3.5.3.1 (Note)	addition
2.2.2.5.3.5.3.2	modification
Table 2.2.2.5-18	modification
2.2.2.5.4.1.1	modification
2.2.2.5.4.2.1	modification
2.2.2.5.4.3.1	modification
2.2.2.5.4.4.1	modification
2.2.2.5.4.4.2	addition
2.2.2.5.4.6.1	modification
2.2.2.5.4.7.1	modification
Table 2.2.2.5-19	modification

Table 2.2.2.8-4	deletion
Table 2.2.2.9-4	deletion
2.3.3.3.7 (Note 3)	modification
2.3.3.3.7.2	modification
2.3.3.4.7 (Note 2)	modification
2.3.3.4.7 (Note 3)	modification
2.3.3.4.7.2	modification
2.3.4.2.1	modification
2.3.5.3.2.1	modification
2.3.5.4.4	modification
2.3.5.4.4.1	modification
2.3.5.4.4.2	modification
2.3.5.4.4.3	modification
2.3.5.4.4.4	addition
2.3.5.4.6.1	modification
2.3.5.4.7.1	modification
2.3.5.6.4	modification
2.3.5.6.4.1	modification
2.3.5.6.4.2	modification
2.3.5.6.4.3	modification
2.3.5.6.4.4	addition
2.3.5.6.6.1	modification
Table 2.3.6-1	modification
2.3.7.1 (Note 5)	deletion
2.3.7.2.1.6	modification
2.3.7.2.2 (Note 2)	modification
2.3.7.2.2 (Note 4)	modification
Table 2.3.7-3	modification
Table 2.3.7-4	modification
2.3.7.3.5.1	modification
2.3.7.3.6.1	modification
2.3.7.3.9.6	modification
2.3.7.3.9.8	modification
2.3.7.3.9.9	modification
2.3.7.3.9.10 (Note)	addition
2.3.7.3.9.10	modification
2.3.7.3.9.11	modification
2.3.7.3.9.12	modification
2.3.7.3.9.13	modification
2.3.7.3.9.14	modification

2.3.7.3.9.15	modification
2.3.7.3.9.20	modification
2.3.7.3.9.21	modification
2.3.7.3.11.2.1	modification
2.3.7.3.11.5.1	modification
2.3.7.3.11.7	modification
2.3.7.3.11.7.1	modification
2.3.7.3.11.7.2	addition
2.3.7.3.11.7.3	modification
2.3.7.3.11.7.4	addition
2.3.7.3.11.8.1	modification
2.3.7.3.11.8.2	modification
2.3.7.3.11.8.3	modification
2.3.7.3.11.8.4	modification
2.3.7.3.11.8.5	addition
2.3.7.4.1.2.1	modification
2.3.7.4.1.2.2	modification
2.3.7.4.1.2.4	modification
2.3.7.4.1.2.5	modification
2.3.7.4.1.2.6	modification
2.3.7.4.1.2.7	modification
2.3.7.4.1.2.8	modification
2.3.7.4.1.3.1	modification
2.3.7.4.2.2.1	modification
2.3.7.4.4.2.1 (Note 2)	modification
2.3.7.4.4.2.1 (Note 4)	modification
2.3.7.4.4.2.1 (Note 5)	modification
2.3.7.4.4.2.1 (Note 6)	modification
2.3.7.4.4.2.2.1	modification
2.3.7.4.4.2.8.2	modification
2.3.7.4.4.2.9	modification
2.3.7.4.5.1.2	modification
2.3.7.4.6.2.1	modification
2.3.7.4.6.3.1	modification
2.3.7.5.1.2.3	modification
2.3.7.5.1.2.5	modification
2.3.7.5.1.2.6	modification
2.3.7.5.1.2.7	modification
2.3.7.5.1.2.8	addition
2.3.7.5.2.1.3	modification
2.3.7.5.2.2.2.	modification

2.3.7.5.3.1.2	modification
2.3.7.5.4.1 (Note 5)	modification
2.3.7.5.4.1 (Note 6)	modification
2.3.7.5.4.1.2.1	modification
2.3.7.5.4.1.3.1	modification
2.3.7.5.4.1.4.2	modification
2.3.7.5.4.1.5.2	modification
2.3.7.6.1 (Note 2)	addition
Table 2.3.7-5 to 28	modification
Table 2.3.7-5 to 28 (Notes)	addition
Table 2.3.8-2	modification
Table 2.3.8-3	modification
Table 2.3.8-4	modification
Table 2.3.8-5	deletion
Table 2.3.8-6	deletion
2.4.3.9.2.1	modification
2.4.4.2.1	modification
2.4.5.1.1 (Note 3)	modification
Figure 2.4.5-18	addition
Table 2.4.5-1	modification
2.4.5.3.5.7	modification
2.4.5.3.6.4	modification
2.4.5.3.6.4.1	modification
2.4.5.3.6.5	modification
2.4.5.3.6.5.1	modification
2.4.5.3.6.7	modification
2.4.5.3.7.9.3	modification
2.4.5.3.7.10.1	modification
2.4.5.3.7.11	modification
2.4.5.3.8.8.3	modification
2.4.5.3.8.10	modification
2.4.5.3.9.1	modification
2.4.5.3.12 (Note 1)	modification
2.4.5.3.12.5.1	modification
2.4.5.3.12.5.2	deletion
2.4.5.3.12.6	modification
2.4.5.3.12.6.1	modification
2.4.5.3.12.7.1	modification
2.4.5.3.12.8.1	modification
2.4.5.3.12.9.1	modification

2.4.5.3.12.10.1	modification
2.4.5.3.12.11.1	modification
2.4.5.3.12.12	modification
2.4.5.3.12.13	modification
2.4.5.3.12.14	modification
2.4.5.3.12.15	modification
2.4.5.3.12.16.1	modification
2.4.5.3.12.17	modification
2.4.5.3.12.17.1	modification
2.4.5.4.2.2	addition
2.4.5.4.6.1	modification
Table 2.4.5-8/b	modification
Table 2.4.5-9/b	modification
Table 2.4.5-10/a	modification
Table 2.4.5-10/b	modification
Table 2.4.8-2	modification
Table 2.4.8-3/a	modification
Table 2.4.8-3/b	modification
Table 2.4.8-4	deletion
Table 2.4.8-5	deletion

2.1 CONTEXT MANAGEMENT APPLICATION

2.1.1 INTRODUCTION

The CM application allows addressing capability for data link applications. The CM application provides the capability to establish a logon between peer ATS ground systems and ATS ground and aircraft systems. Once an appropriate connection is established, CM provides data link application information, the capability to log-on to another ground system, and the capability to update log-on information.

Note 1.— Structure

- a) 2.1.1: *INTRODUCTION* contains the purpose, structure, and a summary of the functions of CM.
- b) 2.1.2: *GENERAL REQUIREMENTS* contains CM ASE Version Number and error processing requirements.
- c) 2.1.3: *ABSTRACT SERVICE DEFINITION* contains the description of the abstract service provided by the CM Application Service Element (CM-ASE).
- d) 2.1.4: *FORMAL DEFINITION OF MESSAGES* contains the formal definition of messages exchanged by CM-ASEs using Abstract Syntax Notation Number One (ASN.1).
- e) 2.1.5: *PROTOCOL DEFINITION* describes the exchanges of messages allowed by the CM protocol, as well as time constraints and CM-ASE protocol descriptions. State tables are also included.
- f) 2.1.6: *COMMUNICATION REQUIREMENTS* contains the requirements that the CM application imposes on the underlying communication system.
- g) 2.1.7: *CM USER REQUIREMENTS* contains requirements imposed on the user of the CM-ASE service.
- h) 2.1.8: *SUBSETTING RULES* contains the conformance requirements which all implementations of the CM protocol obey.

Note 2.— Functional Descriptions

- a) *Logon Functional Description*
 - 1) *The Logon function can only be air initiated. The aircraft system can use the logon function to provide an application name and version number for each air-only initiated application, and an application name, address, and version number for each application that the aircraft wishes to use that can be ground initiated, along with flight plan information as required by the ground system. In response, the ground provides an application name for each ground-only initiated requested application and an application name, address and version number for each requested application that can be air initiated and that the ground can support.*

- 2) Up to a maximum of 256 applications can be supported.
 - 3) Each time a logon is accomplished between a given aircraft and a ground system, the latest exchanged information replaces any previous information for each indicated application.
 - 4) The CM Logon Request message provides required flight plan information, the aircraft's CM application name and address, and information for each application for which data link services are desired. For each application that can be ground initiated the aircraft must provide the application name, version number and address. For each application which is only air initiated the aircraft must provide the application name and version number.
 - 5) The CM Logon Response message provides information for the logon-indicated air-initiated applications. For each desired air-initiated application the ground provides the application name, version number, and address.
- b) *Update Functional Description*
- 1) This function provides a method for the ground system to update application information. This function assumes that the logon function has been accomplished.
 - 2) The CM Update message can provide updated ground information for up to 256 applications. For each updated application the ground provides the application's name, version number and address.
- c) *Contact Functional Description*
- 1) This function provides a method for the ground system to request the aircraft system to initiate the logon function with a designated ground system. It is expected that the contact function will only be used when ground connectivity is not available between respective ground system applications. This function assumes that the logon function has been accomplished with the ground system initiating the contact function. The ground initiates this function with a contact request specifying which ground system to logon with. The aircraft initiates a logon as specified above and indicates the success or lack thereof of the logon.
 - 2) The CM Contact Request message provides the ground system CM application address that the initiating ground system is requesting the aircraft to logon with.
 - 3) The CM Contact Response message provides the information indicating whether or not the requested contact was successful.
- d) *Forwarding Functional Description*
- 1) This function provides a method for a ground system to forward aircraft information received from the CM Logon function to another ground system. This function is initiated by a ground system, which supports ground-ground forwarding, having completed a successful logon that can then forward the aircraft CM Logon

information to other ground systems. It is a one-way forwarding of information with an indication of success, failure or non-support from the receiving ground system. If the ground system receiving this CM information supports ground-ground forwarding, it can then initiate a CM Update function to provide information to the aircraft for any air-initiated applications.

- 2) The CM Forward Request message contains the information as provided in the initial logon.

e) *Registration Functional Description*

- 1) This function provides a method for the air and ground CM applications to make available the application name, address, and version number for each application exchanged in the logon, update or forward functions to other applications or communications systems in the aircraft or on the ground.
- 2) There are no message exchanges for this function.

2.1.2 GENERAL REQUIREMENTS

2.1.2.1 CM ASE Version Number

2.1.2.1.1 The CM-air-ASE and CM-ground-ASE version numbers shall both be set to one.

2.1.2.2 Error Processing Requirements

2.1.2.2.1 In the event of information input by the CM-user being incompatible with that able to be processed by the system, the CM-user shall be notified.

2.1.2.2.2 In the event of a CM-user invoking a CM service primitive when the CM-ASE is not in a state specified in 2.1.5, the CM-user shall be notified.

2.1.3 THE ABSTRACT SERVICE

2.1.3.1 Service Description

2.1.3.1.1 An implementation of either the CM ground based service or the CM air based service shall exhibit external behaviour consistent with having implemented a CM-ground-ASE or CM-air-ASE respectively.

Note 1.— 2.1.3 defines the abstract service interface for the CM service. The CM-ASE abstract service is described from the viewpoint of the CM-air-user, the CM-ground-user and the CM-service-provider.

Note 2.— 2.1.3 defines the static behaviour (i.e., the format) of the CM abstract service. Its dynamic behaviour (i.e., how it is used) is described in 2.1.7.

Note 3.— Figure 2.1.3-1 shows the functional model of the CM Application. The functional modules identified in this model are the following:

- a) the CM-user,
- b) the CM Application Entity (CM-AE) service interface,
- c) the CM-AE,
- d) the CM Control Function (CM-CF),
- e) the CM Application Service Element (CM-ASE) service interface,
- f) the CM-ASE, and
- g) the Dialogue Service (DS) interface.

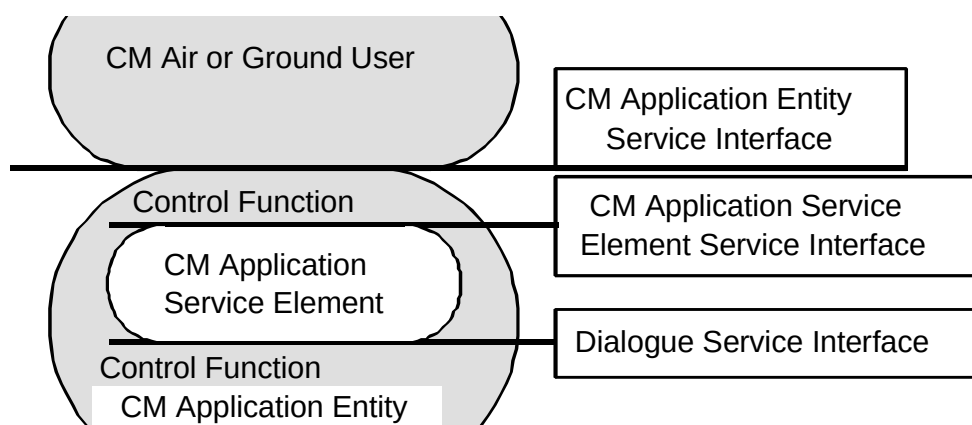


Figure 2.1.3-1. Functional Model of the CM Application

Note 4.— The CM-user represents the operational part of the CM system. This user does not perform the communication functions but relies on a communication service provided to it via the CM-AE through the CM-AE service interface. The individual actions at this interface are called CM-AE service primitives. Similarly, individual actions at other interfaces in the communication system are called service primitives at these interfaces.

Note 5.— The CM-AE consists of several elements including the CM-ASE and the CM-CF. The DS interface is made available by the CM-CF to the CM-ASE for communication with the peer CM-ASE.

Note 6.— The CM-ASE is the element in the communication system which executes the CM specific protocol. In other words, it takes care of the CM specific service primitive sequencing actions, message creation, timer management, error and exception handling.

Note 7.— The CM-ASE interfaces only with the CM-CF. This CM-CF is responsible for mapping service primitives received from one element (such as the CM-ASE and the CM-user) to other elements which interface with it. The part of the CM-CF which is relevant from the point of view of the CM application, i.e. the part between the CM-user and the CM-ASE, will map CM-AE service primitives to CM-ASE service primitives transparently.

Note 8.— The DS interface is the interface between the CM-ASE and the part of CM-CF underneath the CM-ASE, and provides the dialogue service as defined in 4.2.

2.1.3.2 The CM-ASE Abstract Service

Note.— There is no requirement to implement the service in a CM product; however, it is necessary to implement the ground based and air based system in such a way that it will be impossible to detect (from the peer system) whether or not an interface has been built.

2.1.3.2.1 The CM-ASE abstract service shall consist of a set of the following services as allowed by the subsetting rules in 2.1.8:

- a) CM-logon service as defined in 2.1.3.3,
- b) CM-update service as defined in 2.1.3.4,
- c) CM-contact service as defined in 2.1.3.5,
- d) CM-end service as defined in 2.1.3.6,
- e) CM-forward service as defined in 2.1.3.7,
- f) CM-user-abort service as defined in 2.1.3.8, and
- g) CM-provider-abort service as defined in 2.1.3.9.

Note 1.— For a given primitive, the presence of each parameter is described by one of the following values in the parameter tables in 2.1.3.

- a) **blank** not present;
- b) **C** conditional upon some predicate explained in the text;
- c) **C(=)** conditional upon the value of the parameter to the left being present, and equal to that value;
- d) **M** mandatory;
- e) **M(=)** mandatory, and equal to the value of the parameter to the left;
- f) **U** user option.

Note 2.— The following abbreviations are used:

- a) **Req** - request; data is input by CM-user initiating the service to its respective ASE,
- b) **Ind** - indication; data is indicated by the receiving ASE to its respective CM-user,
- c) **Rsp** - response; data is input by receiving CM-user to its respective ASE, and
- d) **Cnf** - confirmation; data is confirmed by the initiating ASE to its respective CM-user.

Note 3.— An unconfirmed service allows a message to be transmitted in one direction, without providing a corresponding response.

Note 4.— A confirmed service provides end-to-end confirmation that a message sent by one user was received by its peer user.

Note 5.— An abstract syntax is a syntactical description of a parameter which does not imply a specific implementation. Only when the CM-ASE maps a parameter onto an APDU field, or vice versa, is the abstract syntax of the parameter described by using the ASN.1 of 2.1.4 for this field.

2.1.3.3 CM-logon Service

Note.— The CM-logon service allows the CM-air-user to initiate data link service. The CM-air-user provides information on each data link application for which it desires a data link service. The CM-ground-user responds indicating whether or not the CM-logon was successful, and if successful, includes information on each data link application it can support. It is a confirmed service.

2.1.3.3.1 If the CM-air-ASE version number is less than or equal to the CM-ground-ASE, then the CM-logon service shall contain the primitives and parameters as presented in Table 2.1.3-1.

Table 2.1.3-1. CM-logon Service Parameters Air-ASE version Number ≤ Ground-ASE Version Number

Parameter Name	Req	Ind	Rsp	Cnf
Facility Designation	M			
Aircraft Address	M	M(=)		
CM ASE Version Number		C		
Logon Request	M	M(=)		
Logon Response			M	M(=)
Class of Communication Service	U			
Maintain Dialogue			U	C(=)

2.1.3.3.2 If the CM-air-ASE version number is greater than the CM-ground-ASE, then the CM-logon service shall contain the primitives and parameters as presented in Table 2.1.3-2.

Table 2.1.3-2. CM-logon Service Parameters Air-ASE version Number > Ground-ASE Version Number

Parameter Name	Req	Cnf
Facility Designation	M	
Aircraft Address	M	
CM ASE Version Number		M
Logon Request	M	
Class of Communication Service	U	

2.1.3.3.3 Facility Designation

Note.— This parameter contains the addressed ground system 's facility designation.

2.1.3.3.3.1 The *Facility Designation* parameter value shall conform to the abstract syntax four to eight-character facility designation.

2.1.3.3.4 Aircraft Address

Note.— This parameter contains 24-bit aircraft address of the aircraft initiating the CM-logon service.

2.1.3.3.4.1 The *Aircraft Address* parameter value shall conform to the abstract syntax 24-bit aircraft address.

2.1.3.3.5 CM ASE Version Number

Note.— This parameter contains the version number of the CM-ASE.

2.1.3.3.5.1 When provided by the CM-ASE, the *Version Number* parameter shall conform to an abstract integer value from 1 to 255.

2.1.3.3.5.2 Only if the CM-air-ASE version number is less than the CM-ground-ASE version number shall the CM-air-ASE version number be indicated to the CM-ground-user.

2.1.3.3.5.3 Only if the CM-air-ASE version number is greater than the CM-ground-ASE version number shall the CM-ground-ASE version number be confirmed to the CM-air-user.

Note 1.— If the CM-air-ASE version number is the same as the CM-ground-ASE version number, the *Version Number* parameter is not present in the indication given to the CM-ground-user, nor in the confirmation to the CM-air-user.

Note 2.— The CM-air-ASE and CM-ground-ASE version numbers are both set to 1.

2.1.3.3.6 Logon Request

Note.— The Logon Request parameter contains the following data:

- a) information for each data link application available on the aircraft, for which the aircraft requires data link service, and
- b) aircraft flight plan information (e.g. flight id, aircraft destination and departure airport and time) as required by the addressed ground system.

2.1.3.3.6.1 The *Logon Request* parameter value shall conform to the ASN.1 abstract syntax CMLogonRequest.

2.1.3.3.7 Logon Response

Note.— This parameter contains information for each requested data link application for which the ground is able to provide data link service.

2.1.3.3.7.1 The *Logon Response* parameter value shall conform to the ASN.1 abstract syntax CMLogonResponse.

2.1.3.3.8 Class of Communication Service

Note.— This parameter contains the value of the required class of communication service if specified by the CM-air-user.

2.1.3.3.8.1 When this parameter is specified by the CM-air-user, the *Class of Communication Service* parameter value shall have one of the following abstract values: “A”, “B”, “C”, “D”, “E”, “F”, “G”, or “H”.

Note.— If not specified by the CM-air-user, this indicates that there is no routing preference.

2.1.3.3.9 Maintain Dialogue

Note 1.— This parameter is used to indicate whether or not the requested CM dialogue is to remain open after a Logon Response.

Note 2.— Whenever a CM dialogue is kept open by the CM-ground-user, it must later be explicitly closed by the CM-ground-user.

Note 3.— This parameter is only provided by the CM-ground-user when the CM-ground-user wishes to keep the CM dialogue open.

2.1.3.3.9.1 If provided by the CM-ground-user this parameter shall have the abstract value “maintain”.

2.1.3.4 CM-update Service

Note.— The CM-update service allows the CM-ground-user to transmit updated ground information for its applications to update previously coordinated CM-logon information. It is an unconfirmed service.

2.1.3.4.1 The CM-update service shall contain the primitives and parameters as presented Table 2.1.3-3.

Table 2.1.3-3. CM-update Service Parameters

Parameter Name	Req	Ind
Aircraft Address	C	
Facility Designation	C	C(=)
Update Information	M	M(=)
Class of Communication Service	U	

2.1.3.4.2 Aircraft Address

Note.— This parameter contains the addressed aircraft's 24-bit aircraft address.

2.1.3.4.2.1 The *Aircraft Address* parameter value shall conform to the abstract syntax 24-bit aircraft address.

2.1.3.4.2.2 If a CM dialogue does not exist when a CM-ground user invokes the CM-update service request, the CM-ground-user shall provide the *Aircraft Address* parameter value.

Note.— The CM-update service does not use this parameter when a CM dialogue exists.

2.1.3.4.3 Facility Designation

Note.— This parameter contains the ground system's facility designation.

2.1.3.4.3.1 The *Facility Designation* parameter value shall conform to the abstract syntax four to eight-character facility designation.

2.1.3.4.3.2 If a CM dialogue does not exist when a CM-ground user invokes the CM-update service request, the CM-ground-user shall provide the *Facility Designation* parameter value.

Note.— The CM-update service does not use this parameter when a CM dialogue exists.

2.1.3.4.4 Update Information

Note.— This parameter contains information on each updated data link application.

2.1.3.4.4.1 The *Update Information* parameter value shall conform to the ASN.1 abstract syntax CMUpdate.

2.1.3.4.5 Class of Communication Service

Note 1.— This parameter contains the value of the required class of communication service if specified by the CM-ground-user.

Note 2.— The CM-update service does not use this parameter when a CM dialogue exists.

2.1.3.4.5.1 Where specified by the CM-ground-user, the *Class of Communication Service* parameter shall have one of the following abstract values: "A", "B", "C", "D", "E", "F", "G", or "H".

Note.— If not specified by the CM-ground-user, this indicates that there is no routing preference.

2.1.3.5 CM-contact Service

Note.— The CM-contact service allows the CM-ground-user, after successful completion of a CM logon, to request that an aircraft logon with another ground system. It is a confirmed service.

2.1.3.5.1 The CM-contact service shall contain the primitives and parameters as presented in Table 2.1.3-4.

Table 2.1.3-4. CM-contact Service Parameters

Parameter Name	Req	Ind	Rsp	Cnf
Aircraft Address	C			
Facility Designation	C	C(=)		
Contact Request	M	M(=)		
Contact Response			M	M(=)
Class of Communication Service	U			

2.1.3.5.2 Aircraft Address

Note.— This parameter contains the addressed aircraft's 24-bit aircraft address.

2.1.3.5.2.1 The *Aircraft Address* parameter value shall conform to the abstract syntax 24-bit aircraft address.

2.1.3.5.2.2 If a CM dialogue does not exist when a CM-ground user invokes the CM-contact service request, the CM-ground-user shall provide the *Aircraft Address* parameter value.

Note.— The CM-contact service does not use this parameter when a CM dialogue exists.

2.1.3.5.3 Facility Designation

Note.— This parameter contains the ground system's facility designation.

2.1.3.5.3.1 The *Facility Designation* parameter value shall conform to the abstract syntax four to eight-character facility designation.

2.1.3.5.3.2 If a CM dialogue does not exist when a CM-ground user invokes the CM-contact service request, the CM-ground-user shall provide the *Facility Designation* parameter value.

Note.— The CM-contact service does not use this parameter when a CM dialogue exists.

2.1.3.5.4 Contact Request

Note.— This parameter contains the facility designation for the ground system that the CM-ground-user requests the aircraft to contact.

2.1.3.5.4.1 The *Contact Request* parameter value shall conform to the ASN.1 abstract syntax CMContactRequest.

2.1.3.5.5 Contact Response

Note.— This parameter indicates success, or lack thereof, of the requested contact.

2.1.3.5.5.1 The *Contact Response* parameter value shall conform to the ASN.1 abstract syntax CMContactResponse.

2.1.3.5.6 Class of Communication Service

Note.— This parameter contains the value of the required class of communication service if specified by the CM-ground-user.

2.1.3.5.6.1 When this parameter is specified by the CM-ground-user, the *Class of Communication Service* parameter value shall have one of the following abstract values: “A”, “B”, “C”, “D”, “E”, “F”, “G”, or “H”.

Note.— If not specified by the CM-ground-user, this indicates that there is no routing preference.

2.1.3.6 CM-end Service

Note 1.— This service provides the capability for the CM-ground-user to terminate a CM dialogue. This service is only needed when the CM-ground-user maintains a CM dialogue during the logon process. It is an unconfirmed service.

Note 2.— Only the CM-ground-user will be capable of initiating the CM-end service.

2.1.3.6.1 The CM-end service shall contain the primitives as presented Table 2.1.3-5.

Table 2.1.3-5. CM-end Service Parameters

Parameter Name	Req	Ind
<i>none</i>		

2.1.3.7 CM-forward Service

Note.— The CM-forward service allows a CM-ground-user to forward data received in a CM-logon request to another CM-ground system. It is a confirmed service.

2.1.3.7.1 If the sending CM-ground-ASE version number is less than or equal to the receiving CM-ground-ASE version number, then the CM-forward service shall contain the primitives and parameters as presented in Table 2.1.3-6.

Table 2.1.3-6. CM-forward Service Parameters Sending Ground-ASE Version Number ≤ Receiving Ground-ASE Version Number

Parameter Name	Req	Ind	Cnf
Called Facility Designation	M		
Calling Facility Designation	M	M(=)	
CM ASE Version Number		C	
Forward Request	M	M(=)	
Class of Communication Service	U		
Result			M

2.1.3.7.2 If the sending CM-ground-ASE version number is greater than the receiving CM-ground-ASE version number, then the CM-forward service shall contain the primitives and parameters as presented in Table 2.1.3-7.

Table 2.1.3-7. CM-forward Service Parameters Sending Ground-ASE Version Number > Receiving Ground-ASE Version Number

Parameter Name	Req	Cnf
Called Facility Designation	M	
Calling Facility Designation	M	
CM ASE Version Number		M
Forward Request	M	
Class of Communication Service	U	
Result		M

2.1.3.7.3 Called Facility Designation

Note.— This parameter contains the receiving ground system's facility designation.

2.1.3.7.3.1 The *Called Facility Designation* parameter value shall conform to the abstract syntax four to eight-character facility designation.

2.1.3.7.4 Calling Facility Designation

Note.— This parameter contains the sending ground system 's facility designation.

2.1.3.7.4.1 The *Calling Facility Designation* parameter value shall conform to the abstract syntax four to eight-character facility designation.

2.1.3.7.5 CM ASE Version Number

Note.— This parameter contains the version number of the CM-ground-ASE.

2.1.3.7.5.1 When provided by the CM-ASE, the *Version Number* parameter shall conform to an abstract integer value from 1 to 255.

2.1.3.7.5.2 Only if the sending CM-ground-ASE version number is less than the receiving CM-ground-ASE version number shall the sending CM-ground-ASE version number be indicated to the receiving CM-ground-user.

2.1.3.7.5.3 Only if the sending CM-ground-ASE version number is greater than the receiving CM-ground-ASE version number shall the receiving CM-ground-ASE version number be confirmed to the sending CM-ground-user.

Note 1.— If the sending CM-ground-ASE version number is the same as the receiving CM-ground-ASE version number, the Version Number parameter is not present in the indication given to the receiving CM-ground-user, nor in the confirmation to the sending CM-ground-user.

Note 2.— The sending CM-ground-ASE and receiving CM-ground-ASE version numbers are both set to 1.

2.1.3.7.6 Forward Request

Note.— This parameter contains information as provided in the CM Logon Request.

2.1.3.7.6.1 The *Forward Request* parameter value shall conform to the ASN.1 abstract syntax CMForwardRequest.

2.1.3.7.7 Class of Communication Service

Note.— This parameter contains the value of the required class of communication service if specified by the initiating CM-ground-user.

2.1.3.7.7.1 When this parameter is specified by the CM-ground-user, the *Class of Communication Service* parameter value shall have one of the following abstract values: "A", "B", "C", "D", "E", "F", "G", or "H".

Note.— If not specified by the CM-ground-user, this indicates that there is no routing preference.

2.1.3.7.8 Result

Note.— This parameter indicates whether or not the information was forwarded as requested.

2.1.3.7.8.1 The *Result* parameter shall conform to the ASN.1 abstract syntax CMForwardResponse.

Note.— When the sending CM-ground-ASE version number is less than or equal to the receiving CM-ground-ASE version number the *Result* parameter takes the abstract value “success”. When the sending CM-ground-ASE version number is greater than the receiving CM-ground-ASE version number the *Result* parameter takes the abstract value “incompatible version”. When the receiving CM-ground-ASE does not support ground-ground forwarding the *Result* parameter takes the abstract value “service not supported”.

2.1.3.8 CM-user-abort Service

Note 1.— This service provides the capability for either the CM-air-user or a CM-ground-user to abort communication with its peer. This can be used for operational or technical reasons. It can be invoked at any time by an active user. Messages in transit may be lost during this operation. It is an unconfirmed service.

Note 2.— If the service is invoked prior to complete establishment of the dialogue, the CM-user-abort indication may not be provided. A CM-provider-abort indication may result instead.

2.1.3.8.1 The CM-user-abort service shall contain the primitives as presented Table 2.1.3-8.

Table 2.1.3-8. CM-user-abort Service Parameters

Parameter Name	Req	Ind
<i>none</i>		

2.1.3.9 CM-provider-abort Service

Note.— This service provides the capability for the CM-service provider to inform its users that it can no longer provide the CM service. Messages in transit may be lost during this operation.

2.1.3.9.1 The CM-provider-abort service shall contain the primitives and parameters as presented Table 2.1.3-9.

Table 2.1.3-9. CM-provider-abort Service Parameters

Parameter Name	Ind
Reason	M

2.1.3.9.2 Reason

Note.— This parameter identifies the reason for the abort.

2.1.3.9.2.1 The *Reason* parameter value shall conform to the ASN.1 abstract syntax CMAbortReason.

2.1.4 FORMAL DEFINITIONS OF MESSAGES

2.1.4.1 Encoding/decoding Rules

2.1.4.1.1 A CM-air-ASE shall be capable of encoding CMAircraftMessage APDUs and decoding CMGroundMessage APDUs.

2.1.4.1.2 A CM-ground-ASE shall be capable of encoding and decoding CMGroundMessage APDUs and decoding CMAircraftMessage APDUs.

2.1.4.2 CM ASN.1 Abstract Syntax

2.1.4.2.1 The abstract syntax of the CM protocol data units shall comply with the description contained in the ASN.1 module CMMesageSetVersion1 (conforming to ISO/IEC 8824-1), as defined in 2.1.4.

CMMesageSetVersion1 DEFINITIONS ::=

BEGIN

-- CM Message Structure

-- Aircraft-generated messages

CMAircraftMessage ::= CHOICE

```
{
  cmLogonRequest      [0]CMLogonRequest,
  cmContactResponse   [1]CMContactResponse,
  cmAbortReason       [2]CMAbortReason,
  ...
}
```

-- Ground-generated messages

CMGroundMessage ::= CHOICE

```
{
  cmLogonResponse     [0]CMLogonResponse,
  cmUpdate            [1]CMUpdate,
  cmContactRequest    [2]CMContactRequest,
  cmForwardRequest    [3]CMForwardRequest,
  cmAbortReason       [4]CMAbortReason,
  cmForwardResponse   [5]CMForwardResponse,
  ...
}
```

-- CM Message Components

AircraftFlightIdentification ::= IA5String (SIZE(2..8))

Airport ::= IA5String (SIZE(4))

APAddress ::= CHOICE

```
{
  longTsap    [0] LongTsap,
  shortTsap   [1] ShortTsap
}
```

AEQualifier ::= INTEGER (0..255)

-- ATN AE-Qualifier Numeric Values are described in 4

AEQualifierVersion ::= SEQUENCE

```
{
  aeQualifier    AEQualifier,
  apVersion      VersionNumber
}
```

AEQualifierVersionAddress ::= SEQUENCE

```
{
  aeQualifier    AEQualifier,
  apVersion      VersionNumber,
  apAddress      APAddress
}
```

CMAbortReason ::= ENUMERATED

```
{
  timer-expired                (0),
  undefined-error              (1),
  invalid-PDU                  (2),
  protocol-error               (3),
  dialogue-acceptance-not-permitted (4),
  dialogue-end-not-accepted     (5),
  communication-service-error   (6),
  communication-service-failure (7),
  invalid-QOS-parameter        (8),
  expected-PDU-missing         (9),
  ...
}
```

CMContactRequest ::= SEQUENCE

```
{
  facilityDesignation    FacilityDesignation,
  address                LongTsap
}
```

CMContactResponse ::= Response

CMForwardRequest ::= CMLogonRequest

CMForwardResponse ::= ENUMERATED

```
{
  success                (0),
  incompatible-version   (1),
  service-not-supported   (2)
}
```

CMLogonRequest ::= SEQUENCE

```
{
  aircraftFlightIdentification [0] AircraftFlightIdentification,
  cMLongTSAP                  [1] LongTsap,
  groundInitiatedApplications [2] SEQUENCE SIZE (1..256) OF AEQualifierVersionAddress
                                                                    OPTIONAL,
  airOnlyInitiatedApplications [3] SEQUENCE SIZE (1..256) OF AEQualifierVersion
                                                                    OPTIONAL,
  facilityDesignation         [4] FacilityDesignation
                                                                    OPTIONAL,
  airportDeparture             [5] Airport
                                                                    OPTIONAL,
  airportDestination           [6] Airport
                                                                    OPTIONAL,
  dateTimeDepartureETD        [7] DateTime
                                                                    OPTIONAL
}
```

CMLogonResponse ::= SEQUENCE

```
{
  airInitiatedApplications [0] SEQUENCE SIZE (1..256) OF AEQualifierVersionAddress
                                                                    OPTIONAL,
  groundOnlyInitiatedApplications [1] SEQUENCE SIZE (1..256) OF AEQualifierVersion
                                                                    OPTIONAL
}
```

CMUpdate ::= CMLogonResponse

Date ::= SEQUENCE

```
{
  year      Year,
  month     Month,
  day       Day
}
```

-- The Date field does not have to correspond to the flight if the field is not to be used;
 -- the field's value can be assigned a meaningless, but compliant, value locally. If operational
 -- use of the Date field is intended, there must be bilateral agreements in place to ensure its proper
 -- use. This is a local implementation issue.

DateTime ::= SEQUENCE

```
{
  date      Date,
  time      Time
}
```

Day ::= INTEGER (1..31)
--unit = Day, Range (1..31), resolution = 1

FacilityDesignation ::= IA5String (SIZE(4..8))

LongTsap ::= SEQUENCE
{
 rDP OCTET STRING (SIZE(5)),
 shortTsap ShortTsap
}

Month ::= INTEGER (1..12)
--unit = Month, Range (1..12), resolution = 1

Response ::= ENUMERATED
{
 contactSuccess (0),
 contactNotSuccessful (1)
}

ShortTsap ::= SEQUENCE
{
 aRS [0] OCTET STRING (SIZE(3)) OPTIONAL,
 -- the aRS contains the ICAO 24 bit aircraft address when the ShortTsap belongs to an aircraft;
 -- or a ground address when the Short Tsap belongs to a ground system
 locSysNselTsel [1] OCTET STRING (SIZE(10..11))
}

Time ::= SEQUENCE
{
 hours Timehours,
 minutes Timeminutes
}

Timehours ::= INTEGER (0..23)
-- units = hour, range (0..23), resolution = 1 hour

Timeminutes ::= INTEGER (0..59)
-- units = minute, range (0..59), resolution = 1 minute

VersionNumber ::= INTEGER (1..255)
-- VersionNumber 0 is reserved for the Dialogue Service

Year ::= INTEGER (1996..2095)
--unit = Year, Range (1996..2095), resolution = 1

END

2.1.5 PROTOCOL DEFINITION

2.1.5.1 Sequence Rules

2.1.5.1.1 With the exception of abort primitives, only the sequence of primitives described in figures 2.1.5-1 through 2.1.5-22 shall be permitted.

Note 1.— The following figures define the valid sequences of primitives that are possible to be invoked during the operation of the CM application. It shows the relationship in time between the service request and the resulting indication, and if applicable, the subsequent response and resulting confirmation.

Note 2.— Abort primitives may interrupt and terminate any of the normal message sequences outlined below.

Note 3.— More than one CM-logon attempt may be made for a given CM-contact request. The number of attempts may be determined by local procedures.

Note 4.— Primitives are processed in the order in which they are received.

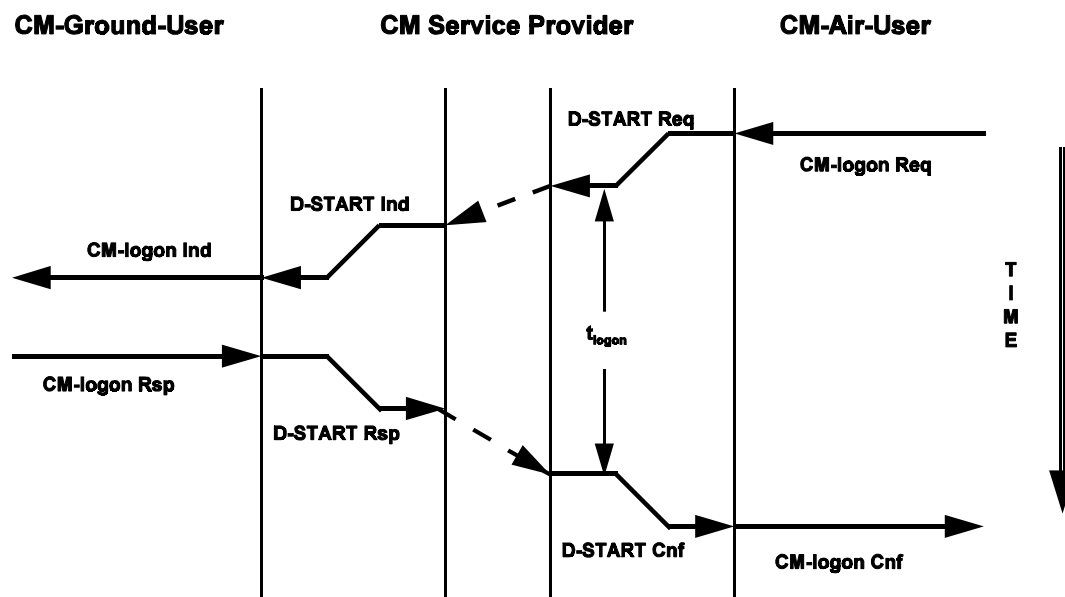


**Figure 2.1.5-1. Sequence Diagram for CM-logon Service
CM-Air-ASE Version $\leq$ CM-Ground-ASE Version**

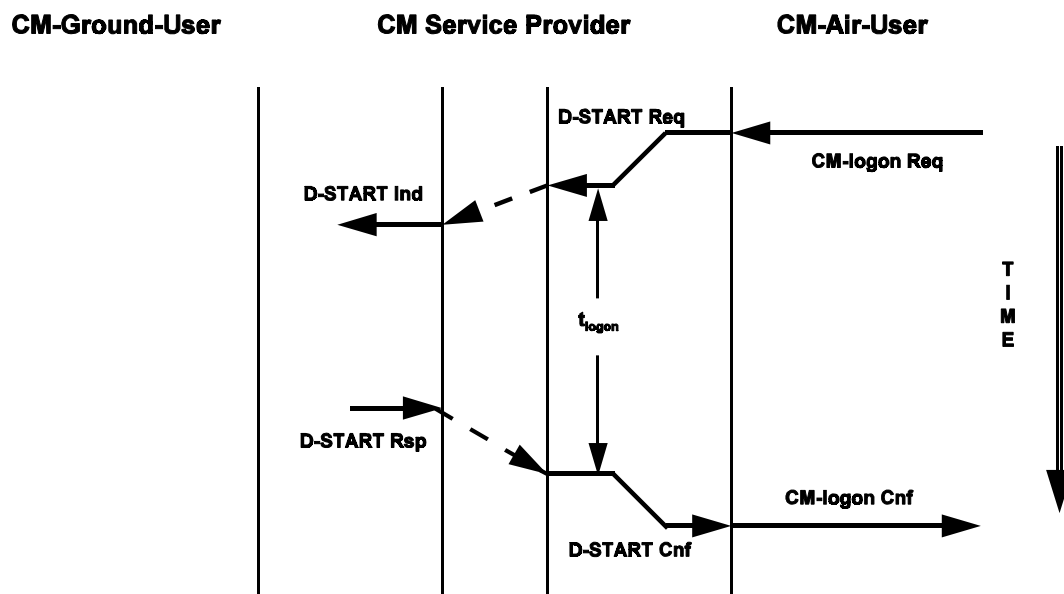


Figure 2.1.5-2. Sequence Diagram for CM-logon Service
CM-Air-ASE Version > CM-Ground-ASE Version

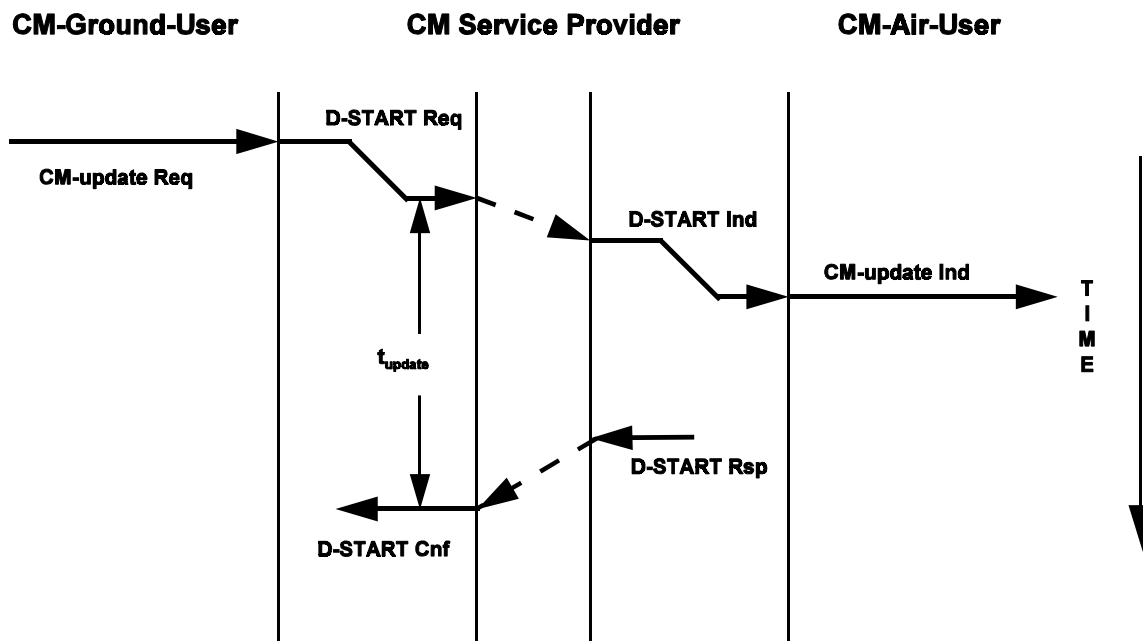


Figure 2.1.5-3. Sequence Diagram for CM-update Service
No Existing CM Dialogue

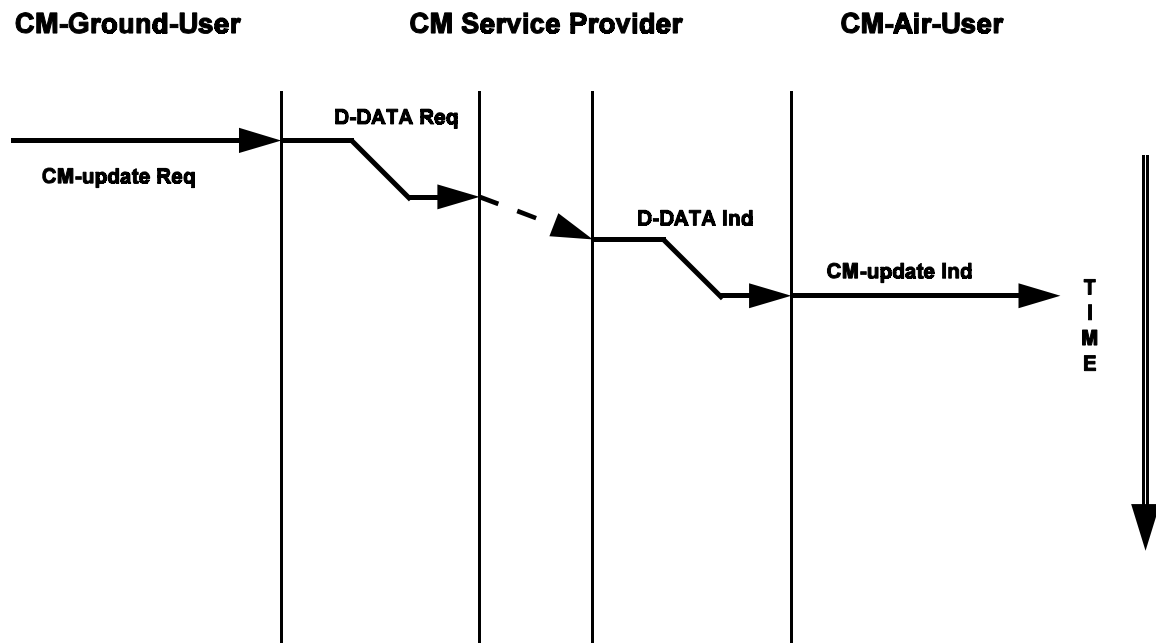


Figure 2.1.5-4. Sequence Diagram for CM-update Service
Existing CM Dialogue

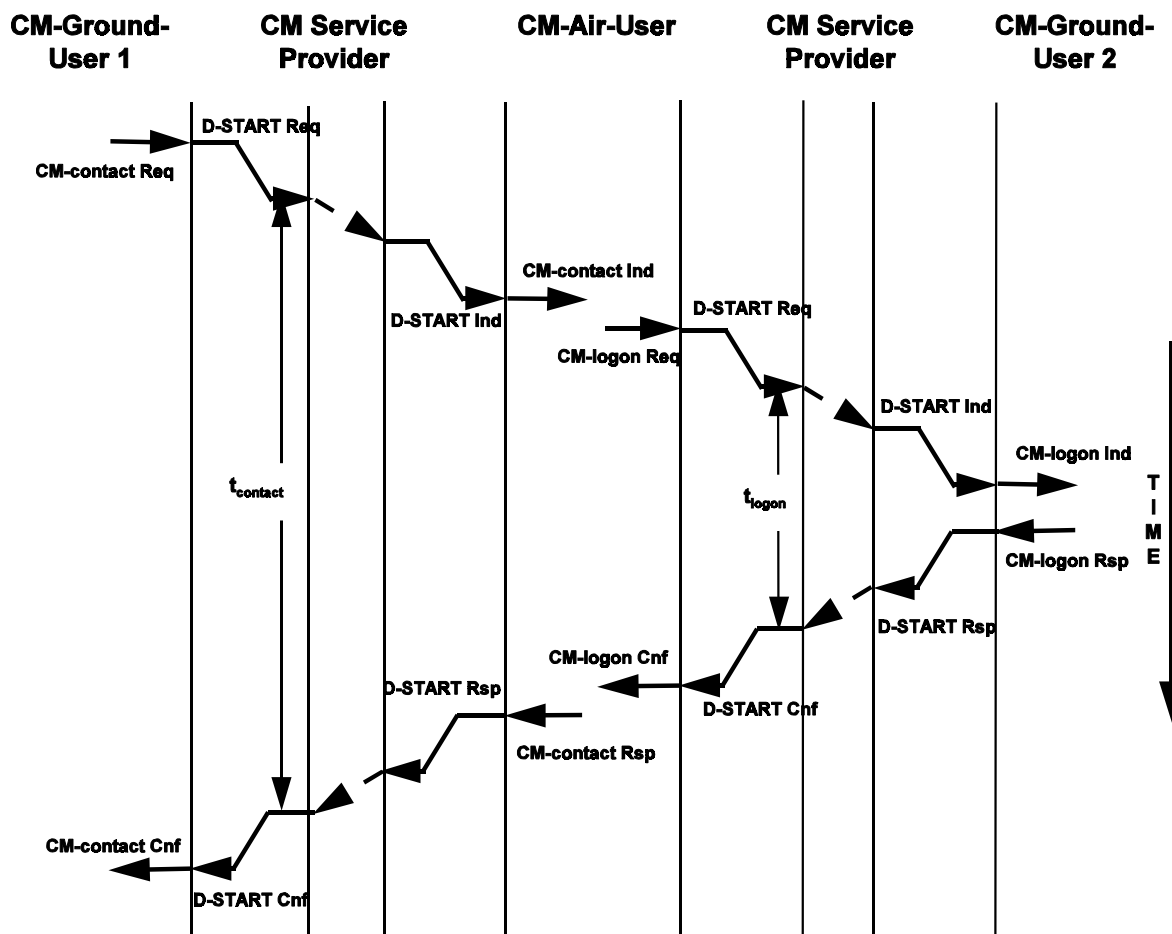


Figure 2.1.5-5. Sequence Diagram for CM-contact Service
No Existing CM Dialogue
With CM-logon Service as in Figure 2.1.5-1

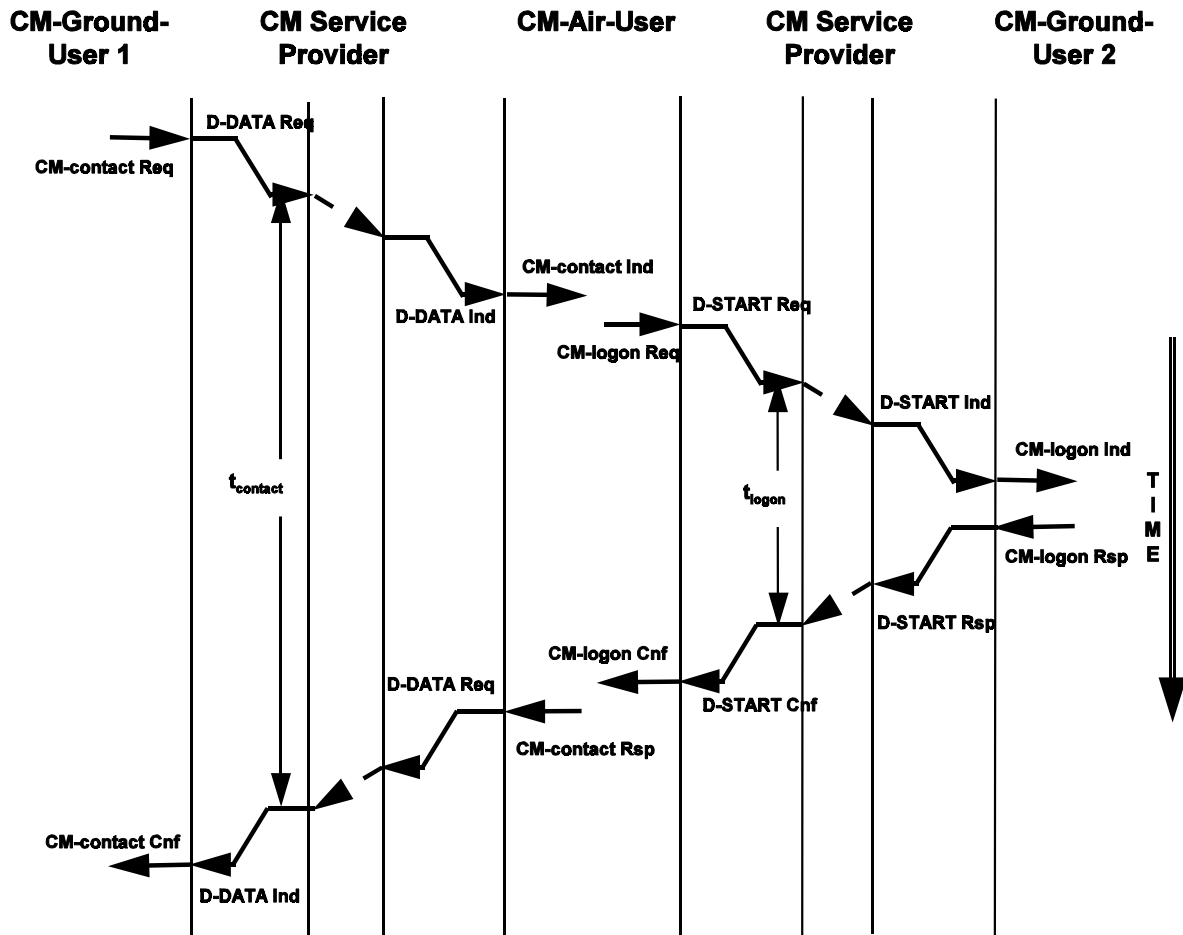


Figure 2.1.5-6. Sequence Diagram for CM-contact Service
With Existing CM Dialogue
With CM-logon Service as in Figure 2.1.5-1

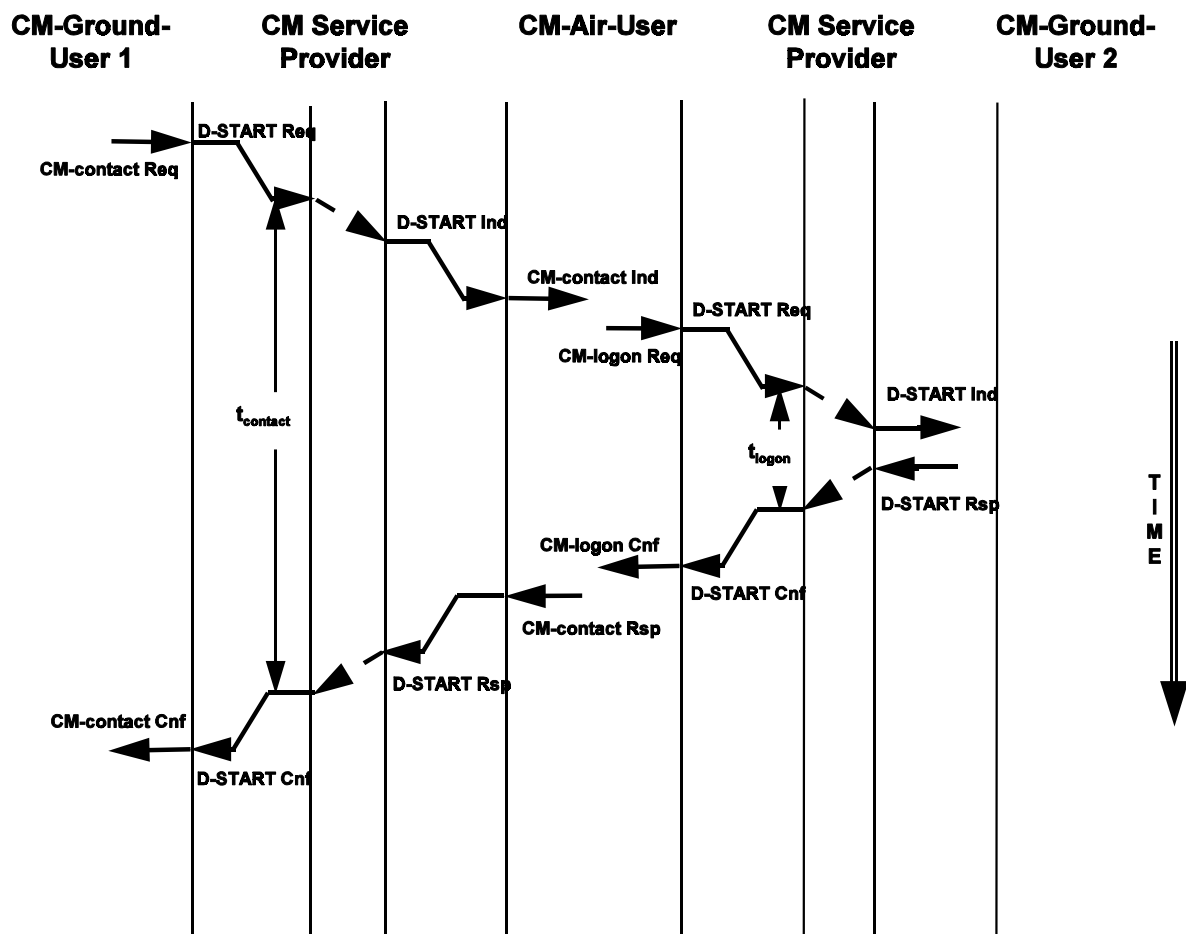


Figure 2.1.5-7. Sequence Diagram for CM-contact Service
No Existing CM Dialogue
With CM-logon Service as in Figure 2.1.5-2

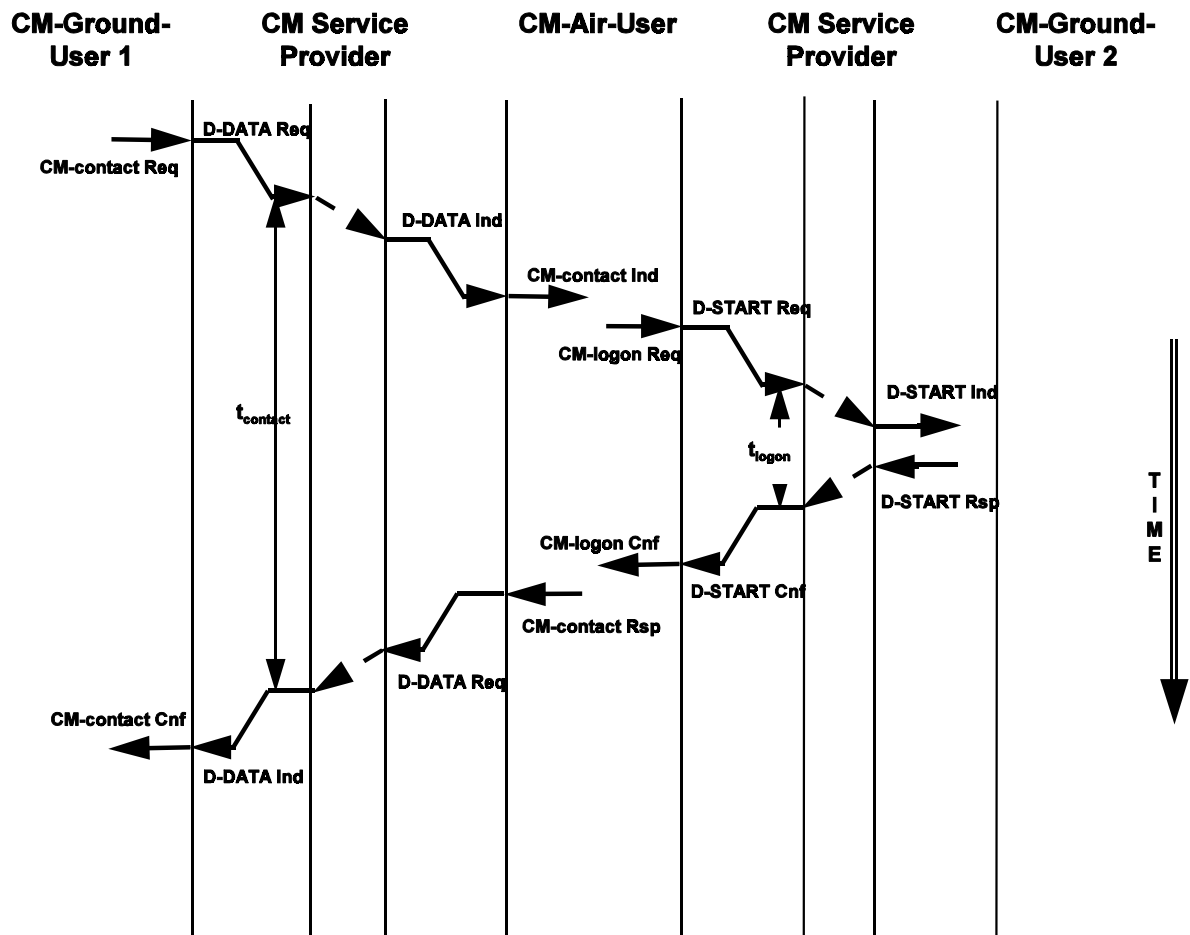


Figure 2.1.5-8. Sequence Diagram for CM-contact Service
Existing CM Dialogue
With CM-logon Service as in Figure 2.1.5-2

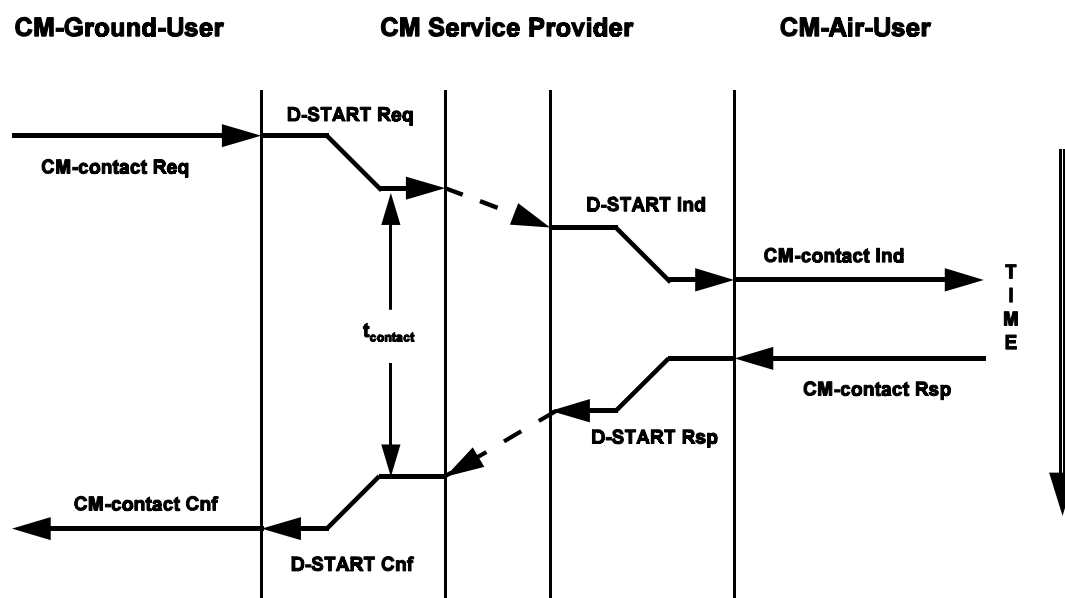


Figure 2.1.5-9. Sequence Diagram for CM-contact Service
No Existing CM Dialogue
Without CM-logon Service as Requested

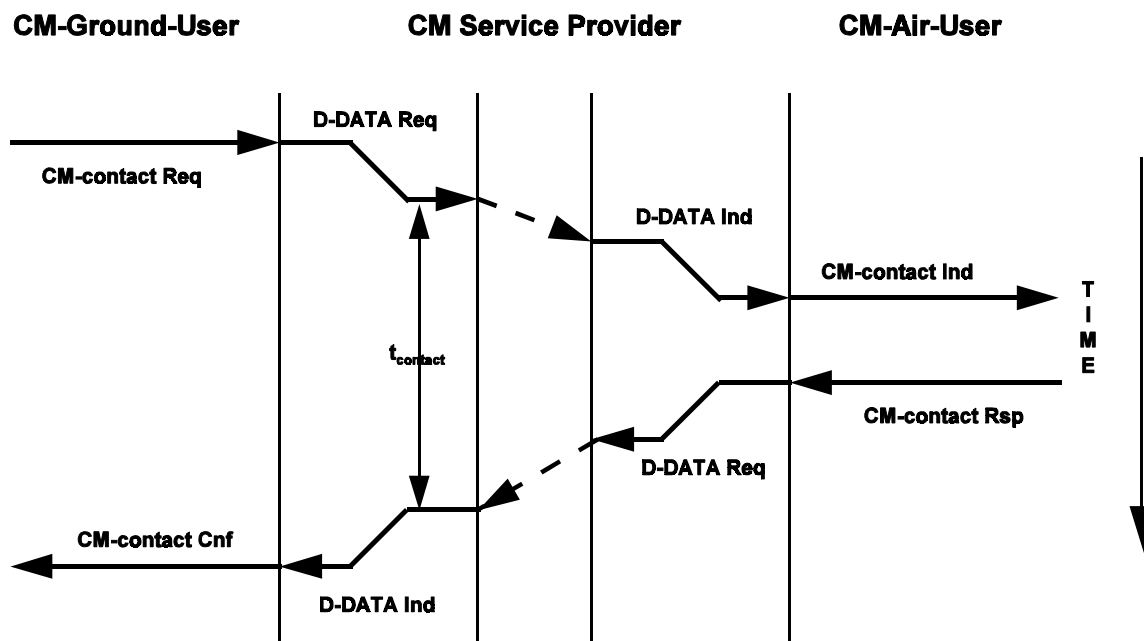


Figure 2.1.5-10. Sequence Diagram for CM-contact Service
With Existing CM Dialogue
Without CM-logon Service as Requested

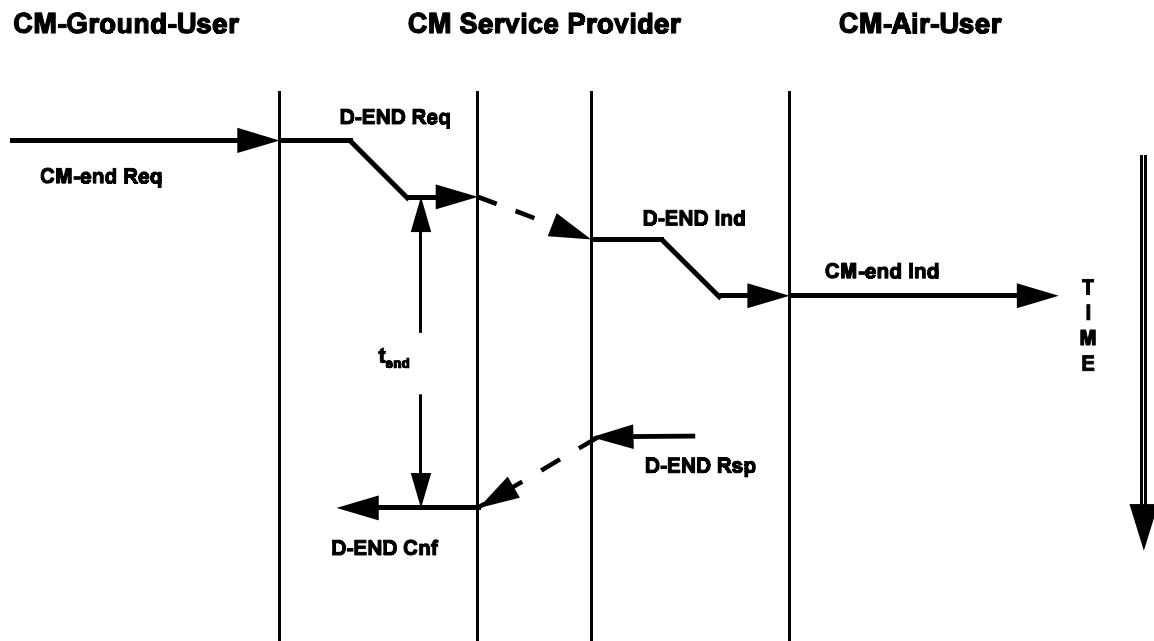


Figure 2.1.5-11. Sequence Diagram for CM-end Service

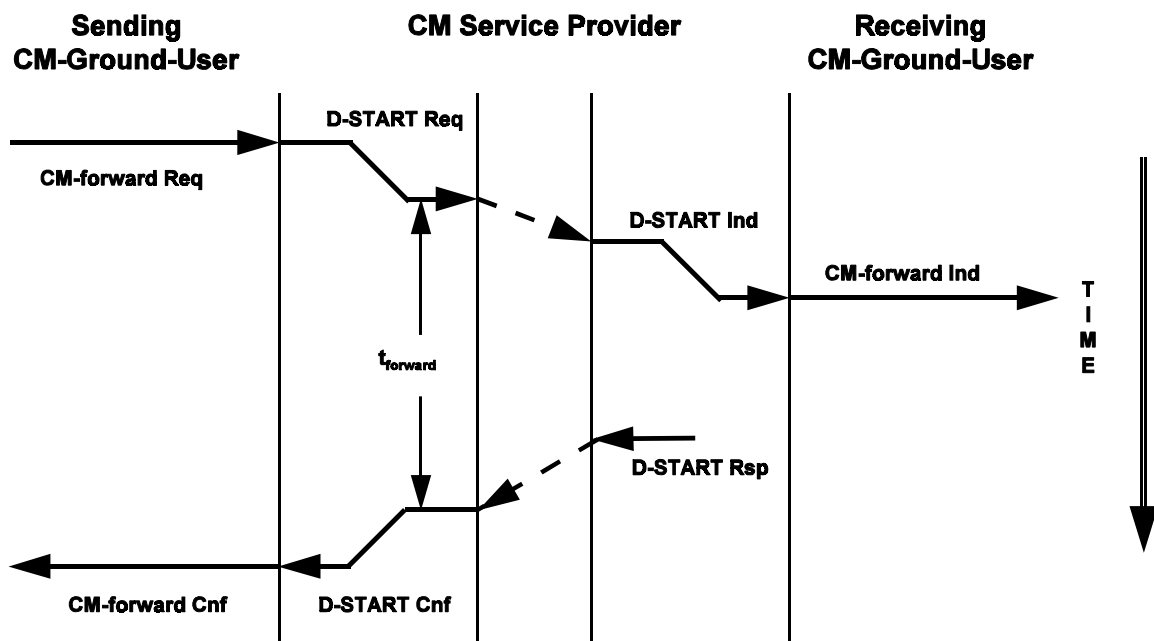


Figure 2.1.5-12. Sequence Diagram for CM-forward Service
 Sending CM-Ground-ASE Version ≤ Receiving CM-Ground-ASE Version,
 Ground-ground Forwarding Supported

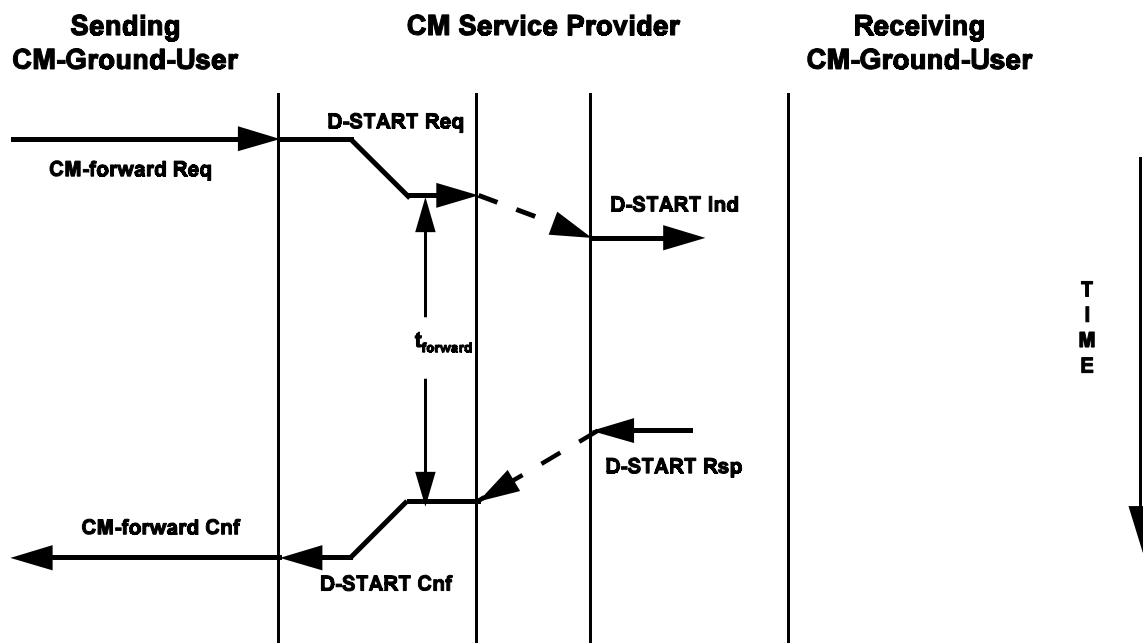


Figure 2.1.5-13. Sequence Diagram for CM-forward Service
 Sending CM-Ground-ASE Version > Receiving CM-Ground-ASE Version
 or Receiving CM-Ground-ASE does not Support Ground-Ground Forwarding

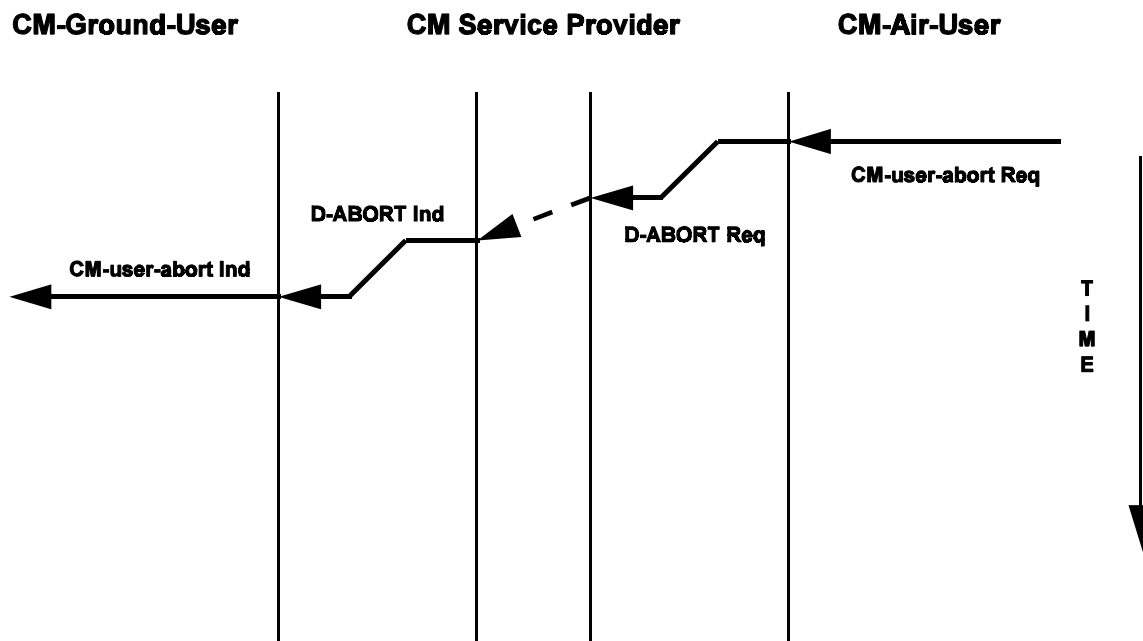


Figure 2.1.5-14. Sequence Diagram for CM-user-abort Service
 CM-Air-User Initiated

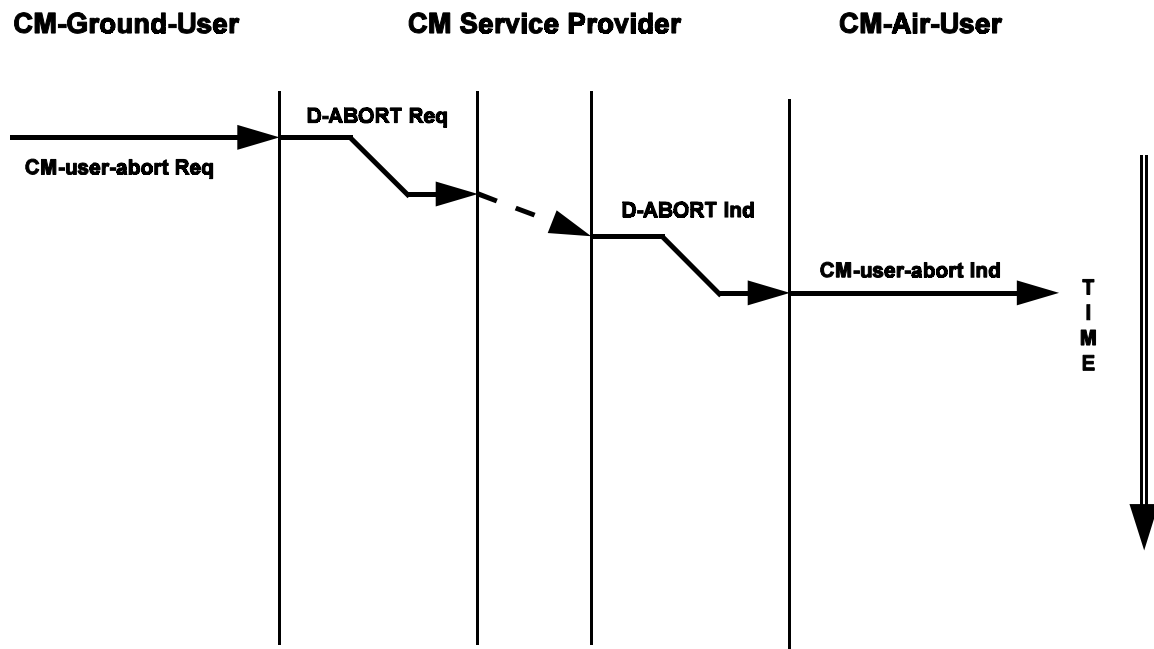


Figure 2.1.5-15. Sequence Diagram for CM-user-abort Service
CM-Ground-User Initiated

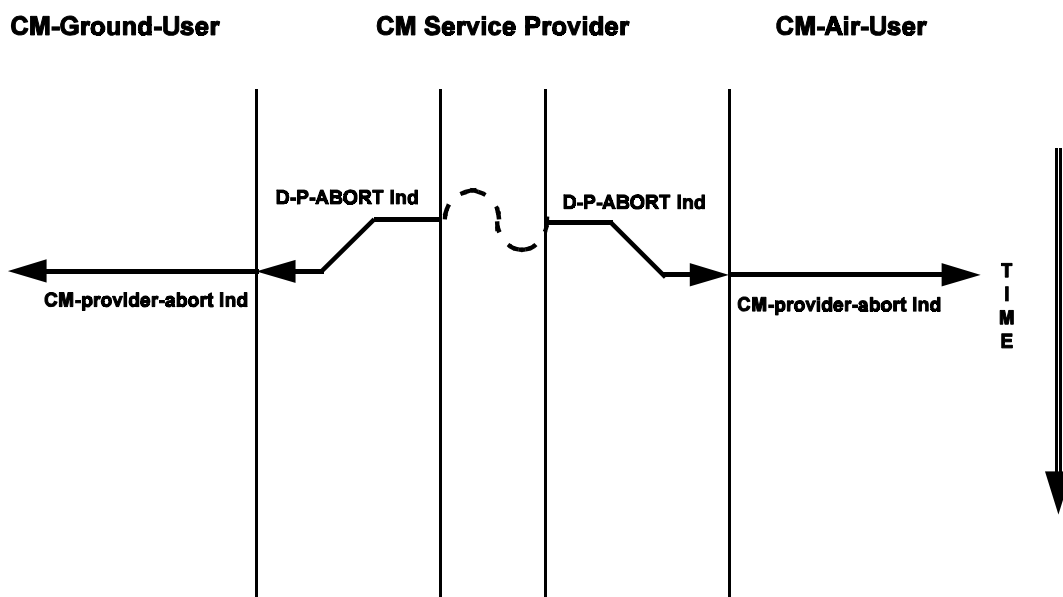


Figure 2.1.5-16. Sequence Diagram for CM-provider-abort Service:
Dialogue Service Abort

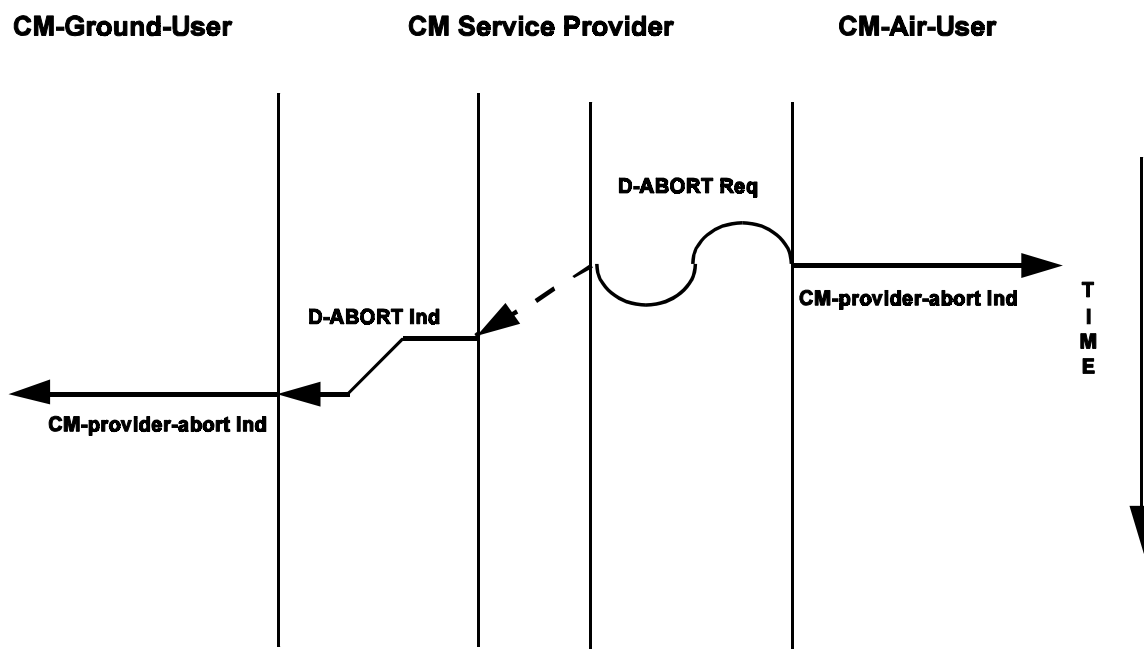


Figure 2.1.5-17. Sequence Diagram for CM-provider-abort Service:
CM-Air-ASE Abort

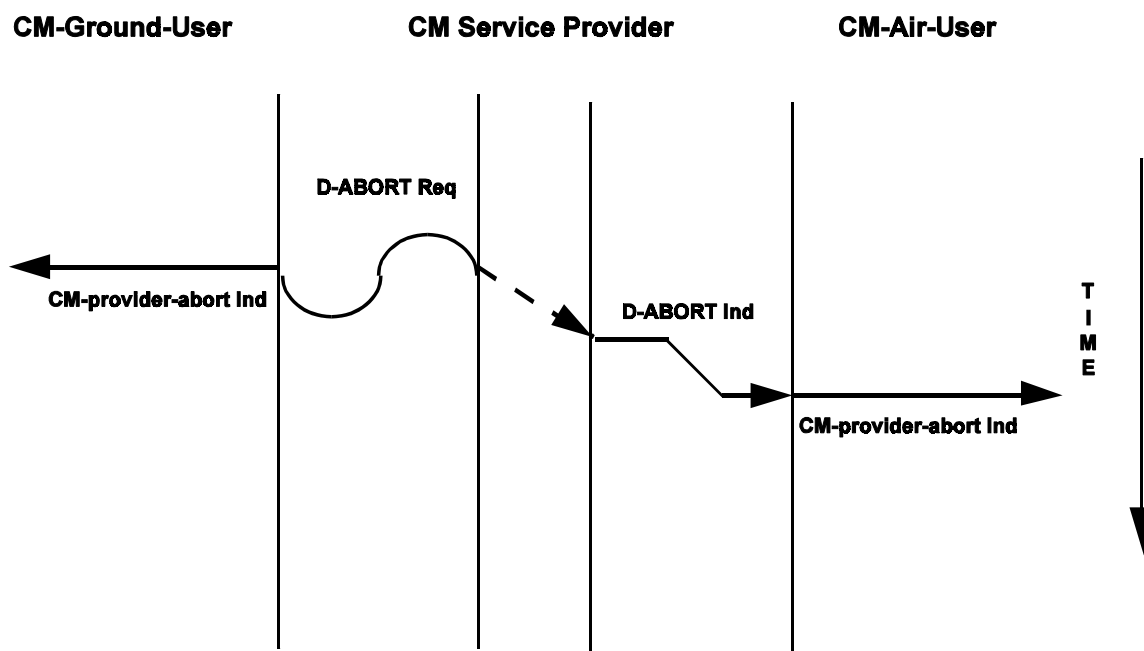


Figure 2.1.5-18. Sequence Diagram for CM-provider-abort Service:
CM-Ground-ASE Abort

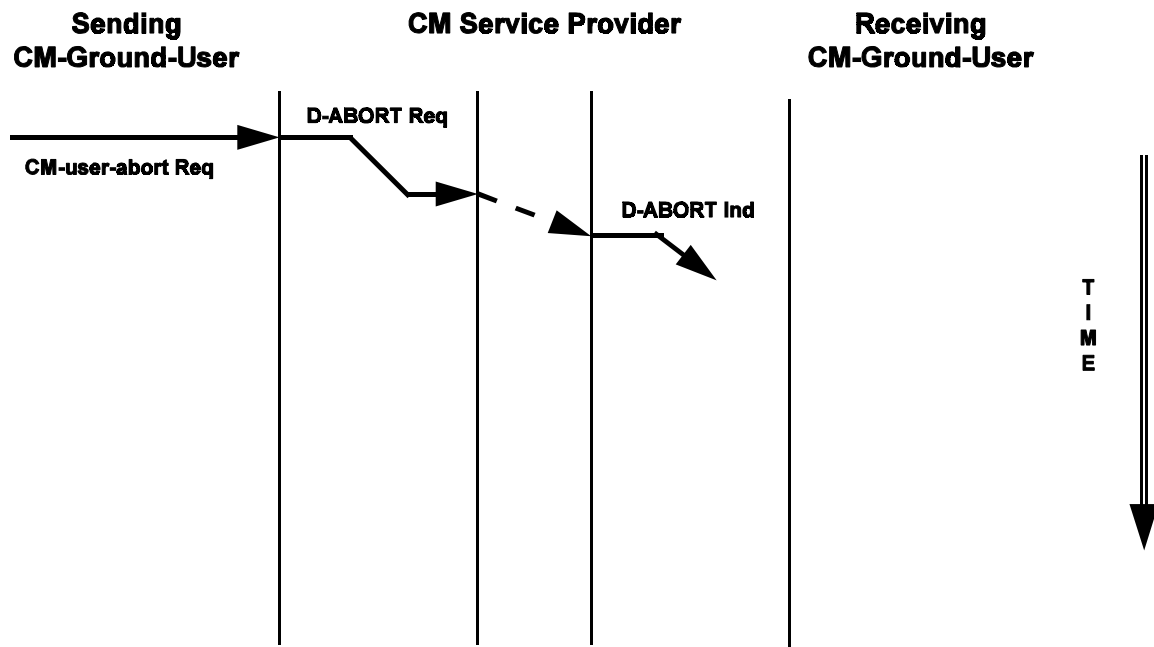


Figure 2.1.5-19. Sequence Diagram for CM-user-abort Service
Sending CM-Ground-User Initiated

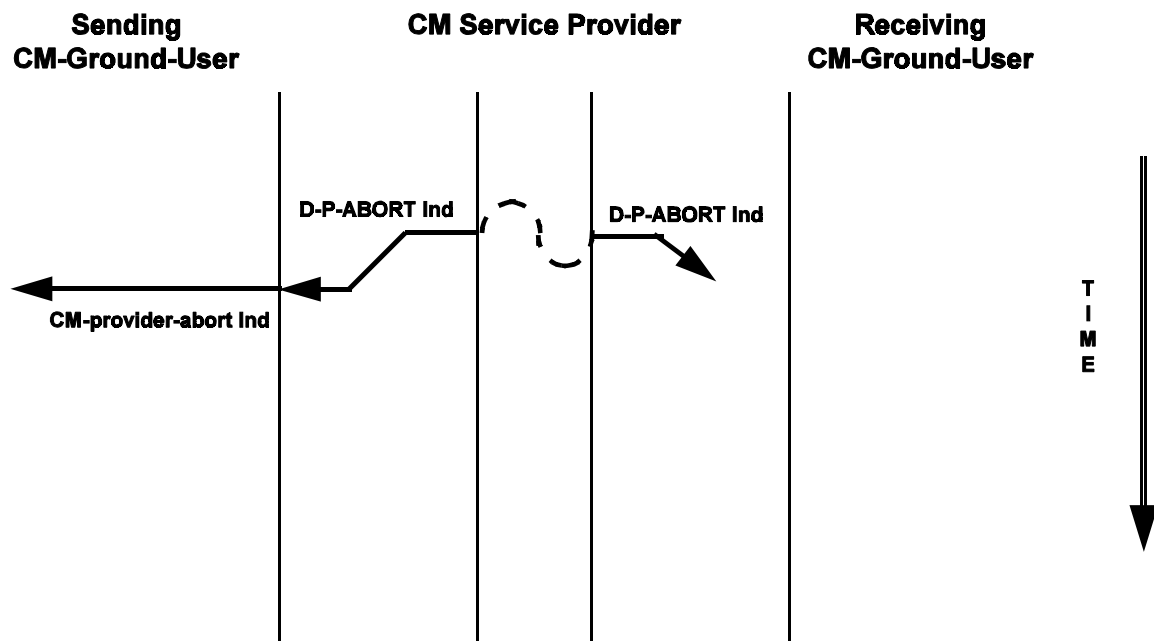


Figure 2.1.5-20. Sequence Diagram for CM-provider-abort Service:
Dialogue Service Abort

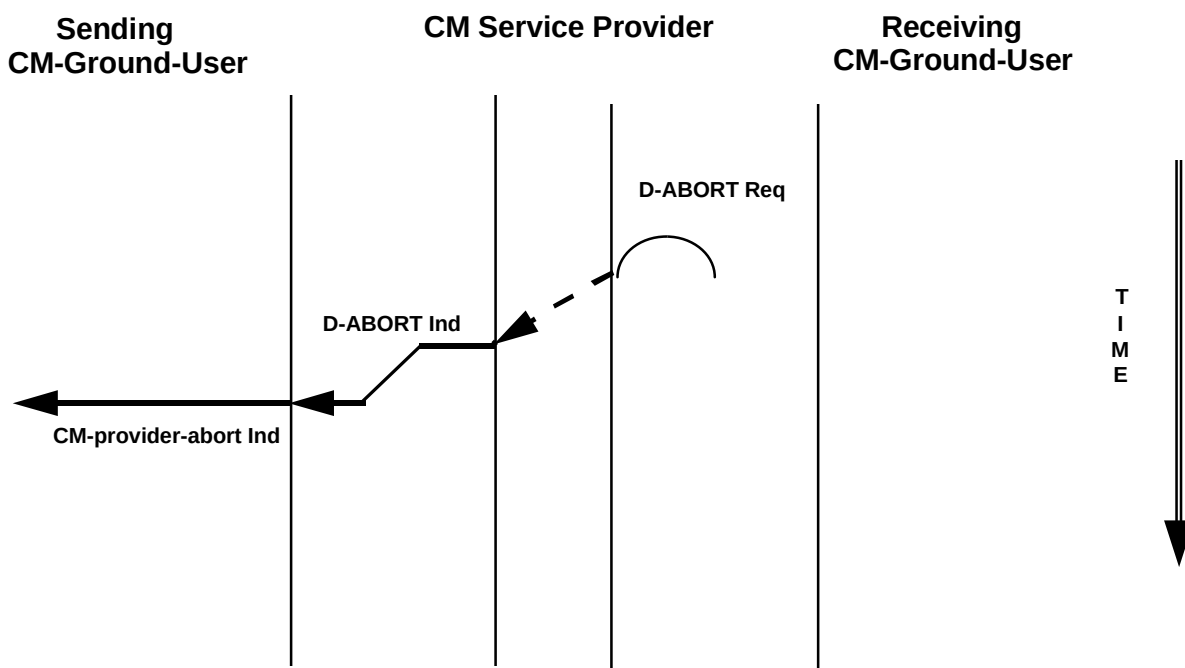


Figure 2.1.5-21. Sequence Diagram for CM-provider-abort Service:
Receiving CM-Ground-ASE Abort

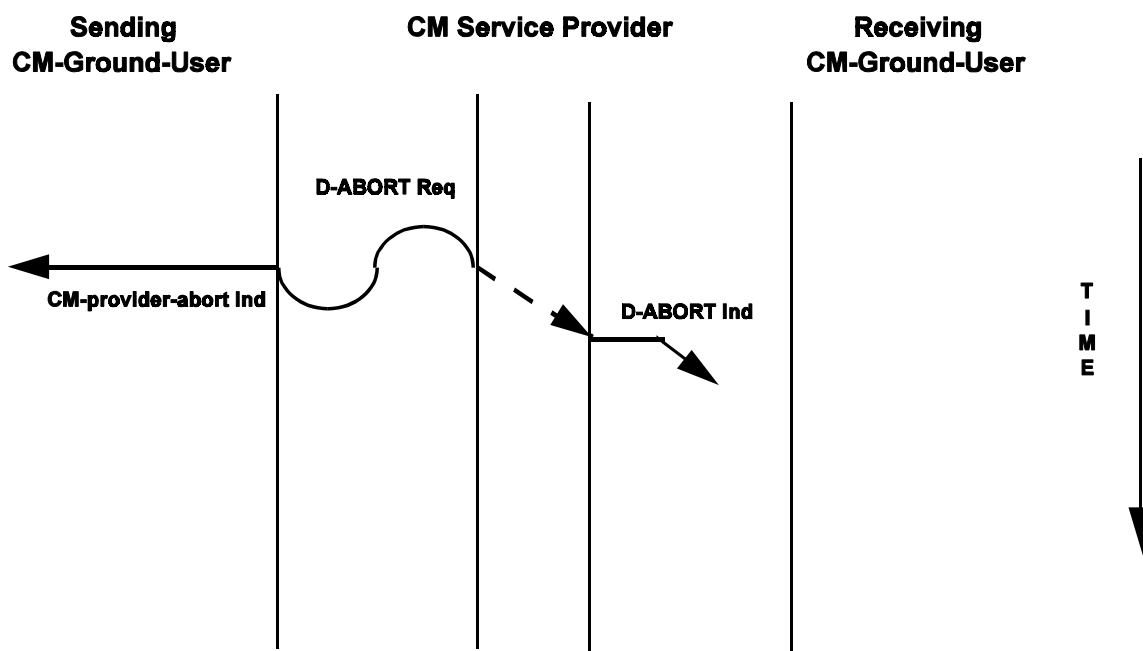


Figure 2.1.5-22. Sequence Diagram for CM-provider-abort Service:
Sending CM-Ground-ASE Abort

2.1.5.2 CM Service Provider Timers

2.1.5.2.1 A CM-ASE shall be capable of detecting when a timer expires.

Note 1.— Table 2.1.5-1 lists the time constraints related to the CM application. Each time constraint requires a timer to be set in the CM protocol machine.

Note 2.— If the timer expires before the final event has occurred, a CM-ASE takes the appropriate action as defined in 2.1.5.4.1.

2.1.5.2.2 **Recommendation.** — The timer values should be as indicated in Table 2.1.5-1.

Table 2.1.5-1. CM Service Provider Timers

CM Service	Timer	Timer Value	Timer Start Event	Timer Stop Event
CM-logout	t_{logout}	4 min	D-START request	D-START confirmation
CM-update	t_{update}	4 min	D-START request	D-START confirmation
CM-contact	t_{contact}	8 min	D-START request D-DATA request	D-START confirmation D-DATA indication
CM-forward	t_{forward}	4 min	D-START request	D-START confirmation
CM-end	t_{end}	4 min	D-END request	D-END confirmation

Note.— The receipt of a CM-user-abort request, D-ABORT indication, or D-P-ABORT indication are also timer stop events.

2.1.5.3 CM-ASE Protocol Description

2.1.5.3.1 Introduction

Note.— 2.1.5.3 presents requirements for CM-ASEs in specific states. 2.1.5.5 contains state tables for the CM-ASEs.

2.1.5.3.1.1 If no actions are described for a CM service primitive when a CM-ASE is in a specific state, then the invocation of that service primitive shall be prohibited while the CM-ASE is in that state.

2.1.5.3.1.2 Upon receipt of a PDU or dialogue service primitive, if no actions are described for their arrival when a CM-ASE is in a specific state, then they are considered not permitted and exception handling procedures as described in 2.1.5.4.4 shall apply.

2.1.5.3.1.3 If a PDU is received that cannot be decoded, then that PDU is considered an invalid PDU and exception handling procedures as described in 2.1.5.4.3 shall apply.

2.1.5.3.1.4 If a PDU is not received where one is expected, then that PDU is considered an invalid PDU and exception handling procedures as described in 2.1.5.4.3 shall apply.

2.1.5.3.2 CM-Air-ASE Protocol Description

Note 1.— The states defined for the CM-air-ASE are the following:

- a) *IDLE,*
- b) *LOGON,*
- c) *CONTACT,*
- d) *DIALOGUE, and*
- e) *CONTACT DIALOGUE.*

Note 2.— The CM-air-user is considered an active user from the time:

- a) *the CM-air-user invokes a CM-logon Req until it:*
 - 1) *receives a CM-logon Cnf, if a dialogue is not maintained,*
 - 2) *receives a CM-end Ind, if a dialogue is maintained,*
 - 3) *receives a CM-user abort,*
 - 4) *receives a CM-provider abort, or*
 - 5) *invokes a CM-user abort,*
- b) *the CM-air-user receives a CM-contact Ind until it:*
 - 1) *invokes a CM-contact Rsp, if a dialogue is not maintained,*
 - 2) *receives a CM-user abort,*
 - 3) *receives a CM-provider abort, or*
 - 4) *invokes a CM-user abort.*

2.1.5.3.2.1 On initiation, the CM-air-ASE shall be in the *IDLE* state.

2.1.5.3.2.2 D-START Indication

2.1.5.3.2.2.1 Upon receipt of a D-START indication, if the CM-air-ASE is in the *IDLE* state and the D-START *Priority* QOS parameter has the abstract value “flight regularity communications”, the D-START *RER* QOS parameter has the abstract value of “low”, the D-START *Routing Class* QOS parameter identifies the traffic category “Air Traffic Service Communications (ATSC)” and the *Calling Peer ID* parameter is a valid four to eight character facility designation then:

2.1.5.3.2.2.1.1 If the APDU contained in the D-START *User Data* parameter is a [CMUpdate] APDU the CM-air-ASE shall:

- a) invoke CM-update service indication with the following:
 - 1) the D-START *Calling Peer ID* parameter value as the CM-update *Facility Designation* parameter value, and
 - 2) the APDU contained in the D-START *User Data* parameter as the CM-update *Update Information* parameter value,
- b) invoke D-START response with the abstract value “rejected (permanent)” provided as D-START *Result* parameter value, and
- c) enter the *IDLE* state.

2.1.5.3.2.2.1.2 If the APDU contained in the D-START *User Data* parameter is a [CMContactRequest] APDU the CM-air-ASE shall:

- a) invoke CM-contact service indication with the following:
 - 1) The D-START *Calling Peer ID* parameter value as the CM-contact *Facility Designation* parameter value, and
 - 2) The APDU contained in the D-START *User Data* parameter as the CM-contact *Contact Request* parameter value, and
- b) enter the *CONTACT* state.

2.1.5.3.2.3 D-START Confirmation

2.1.5.3.2.3.1 Upon receipt of a D-START confirmation, if the CM-air-ASE is in the *LOGON* state and if the APDU contained in the D-START *User Data* parameter is a [CMLogonResponse] APDU or if the D-START *User Data* parameter is not present but the D-START *DS User Version Number* parameter is present, the CM-air-ASE shall:

- a) stop timer t_{logon} ,

- b) if the abstract value of the D-START *Result* parameter is “rejected (permanent)” and the abstract value of the D-START *Reject Source* parameter is “DS user” then:
 - 1) if the version number of the CM-air-ASE is greater than the D-START *DS User Version Number* parameter value, invoke CM-logon service confirmation with the D-START *DS User Version Number* parameter value as the CM-logon *CM-ASE Version Number* parameter value, or
 - 2) else invoke CM-logon service confirmation with the APDU contained in the D-START *User Data* parameter as the CM-logon *Logon Response* parameter, and
 - 3) enter the *IDLE* state.
- c) if the abstract value of the D-START *Result* parameter is “accepted” then:
 - 1) invoke CM-logon service confirmation with the following:
 - i) the APDU contained in the D-START *User Data* parameter as the CM-logon *Logon Response* parameter value,
 - ii) the D-START *Result* parameter as the CM-Logon *Maintain Dialogue* parameter value, and
 - 2) enter the *DIALOGUE* state.

2.1.5.3.2.4 D-DATA Indication

2.1.5.3.2.4.1 Upon receipt of a D-DATA indication, if the CM-air-ASE is in the *DIALOGUE* state then:

2.1.5.3.2.4.1.1 If the APDU contained in the D-DATA *User Data* parameter is a [CMUpdate] APDU the CM-air-ASE shall:

- a) invoke CM-update service indication with the APDU contained in the D-DATA *User Data* parameter as the CM-update service *Update Information* parameter value, and
- b) remain in the *DIALOGUE* state.

2.1.5.3.2.4.1.2 If the APDU contained in the D-DATA *User Data* parameter is a [CMContactRequest] APDU the CM-air-ASE shall:

- a) invoke CM-contact service indication with the APDU contained in the D-DATA *User Data* parameter as the CM-contact service *Contact Request* parameter value, and
- b) enter the *CONTACT DIALOGUE* state.

2.1.5.3.2.5 D-END Indication

2.1.5.3.2.5.1 Upon receipt of a D-END indication, if the CM-air-ASE is in the *DIALOGUE* state then the CM-air-ASE shall:

- a) invoke CM-end service indication,
- b) invoke D-END response with the D-END *Result* parameter set to the abstract value “accepted”, and
- c) enter the *IDLE* state.

2.1.5.3.2.6 CM-logon Service Request

2.1.5.3.2.6.1 Upon receipt of a CM-logon service request:

2.1.5.3.2.6.1.1 If the CM-air-ASE is in the *IDLE* state the CM-air-ASE shall:

- a) create a CMAircraftMessage APDU with a cmLogonRequest APDU-element based on the *Logon Request* parameter value,
- b) invoke D-START request with the following:
 - 1) the CM-logon *Facility Designation* parameter value as the D-START *Called Peer ID* parameter value,
 - 2) the CM-logon *Aircraft Address* parameter value as the D-START *Calling Peer ID* parameter value,
 - 3) the CM-air-ASE version number as the D-START *DS User Version Number* parameter value,
 - 4) the D-START *Quality of Service* parameters set as follows:
 - i) if provided, the CM-logon service *Class of Communication Service* parameter value as the D-START *Routing Class* parameter value,
 - ii) the abstract value of “flight regularity communications”, as the D-START *Priority* parameter value, and
 - iii) the abstract value of “low” as the D-START *RER* parameter value, and
 - 5) the CMAircraftMessage APDU as the D-START *User Data* parameter value,
- c) start timer t_{logon} , and
- d) enter the *LOGON* state.

2.1.5.3.2.7 CM-contact Service Response

2.1.5.3.2.7.1 Upon receipt of a CM-contact service response, if the CM-air-ASE is in the *CONTACT* state the CM-air-ASE shall:

- a) create a CMAircraftMessage APDU with cmContactResponse APDU-element based on the CM-contact *Contact Response* parameter value,
- b) invoke D-START response with the following:
 - 1) the abstract value “rejected (permanent)” as D-START *Result* parameter value, and
 - 2) the CMAircraftMessage APDU as the D-START *User Data* parameter value, and
- c) enter the *IDLE* state.

2.1.5.3.2.7.2 Upon receipt of a CM-contact service response, if the CM-air-ASE is in the *CONTACT DIALOGUE* state the CM-air-ASE shall:

- a) create a CMAircraftMessage APDU with a cmContactResponse APDU-element based on the CM-contact *Contact Response* parameter value,
- b) invoke D-DATA request with the CMAircraftMessage APDU as the D-DATA *User Data* parameter value, and
- c) enter the *DIALOGUE* state.

2.1.5.3.2.8 CM-user-abort Service Request

2.1.5.3.2.8.1 Upon receipt of a CM-user-abort service request, if the CM-air-ASE is not in the *IDLE* state the CM-air-ASE shall:

- a) stop timer t_{logon} , if set,
- b) invoke D-ABORT request with the D-ABORT *Originator* parameter set to the abstract value “user”, and
- c) enter the *IDLE* state.

2.1.5.3.2.9 D-ABORT Indication

2.1.5.3.2.9.1 Upon receipt of a D-ABORT indication, if the CM-air-ASE is not in the *IDLE* state the CM-air-ASE shall:

- a) stop timer t_{logon} , if set

- b) if the CM-air-user is an active user, then:
 - 1) if the D-ABORT *Originator* parameter contains the abstract value “user” invoke CM-user-abort service indication, or
 - 2) else invoke CM-provider-abort service indication with the APDU contained in the D-ABORT *User Data* parameter as the CM-provider-abort service *Reason* parameter value, and
- c) enter *IDLE* state.

2.1.5.3.2.10 D-P-ABORT Indication

2.1.5.3.2.10.1 Upon receipt of a D-P-ABORT indication, if the CM-air-ASE is not in the *IDLE* state the CM-air-ASE shall:

- a) stop timer t_{logon} , if set,
- b) if the CM-air-user is an active user, invoke CM-provider-abort service indication with the CM-provider-abort *Reason* parameter set to the abstract value “communication-service-failure”, and
- c) enter the *IDLE* state.

2.1.5.3.3 CM-Ground-ASE Protocol Description

Note 1.— The states defined for the CM-ground-ASE are the following:

- a) *IDLE*,
- b) *LOGON*,
- c) *UPDATE*,
- d) *CONTACT*,
- e) *DIALOGUE*,
- f) *CONTACT DIALOGUE*,
- g) *END*, and
- h) *FORWARD*.

Note 2.— The CM-ground-user is considered an active user from the time:

- a) *the CM-ground-user receives a CM-logon Ind until it:*
 - 1) *invokes a CM-logon Rsp, if a dialogue is not maintained,*

- 2) *invokes a CM-end Req, if a dialogue is maintained,*
- 3) *receives a CM-user abort,*
- 4) *receives a CM-provider abort, or*
- 5) *invokes a CM-user abort,*
- b) *the CM-ground-user invokes a CM-contact Req until it*
 - 1) *receives a CM-contact Cnf, if a dialogue is not maintained,*
 - 2) *receives a CM-user abort,*
 - 3) *receives a CM-provider abort, or*
 - 4) *invokes a CM-user abort*
- c) *the CM-ground-user invokes a CM-forward Req until it*
 - 1) *receives a CM-forward Cnf,*
 - 2) *receives a CM-provider abort, or*
 - 3) *invokes a CM-user abort.*

2.1.5.3.3.1 On initiation, the CM-ground-ASE shall be in the *IDLE* state.

2.1.5.3.3.2 D-START Indication

2.1.5.3.3.2.1 Upon receipt of a D-START indication, if the CM-ground-ASE is in the *IDLE* state and the APDU contained in the D-START *User Data* parameter is a [CMLogonRequest] APDU and the abstract value of the D-START *Calling Peer ID* parameter is a 24 bit Aircraft Address and the D-START *Priority QOS* parameter has the abstract value “flight regularity communications” and the D-START *RER QOS* parameter has the abstract value of “low” and the D-START *Routing Class QOS* parameter identifies the traffic category “Air Traffic Service Communications (ATSC)”, then:

2.1.5.3.3.2.1.1 If the D-START *DS User Version Number* parameter value is greater than the CM-ground-ASE version number the CM-ground-ASE shall:

- a) *invoke D-START response with the following:*
 - 1) *the CM-ground-ASE version number as the D-START DS User Version Number parameter value, and*
 - 2) *the abstract value of “rejected (permanent)” as the D-START Result parameter value, and*
- b) *enter the IDLE state.*

2.1.5.3.3.2.1.2 If the D-START *DS User Version Number* parameter value is less than the CM-ground-ASE version number the CM-ground-ASE shall:

- a) invoke CM-logon service indication with the following:
 - 1) the D-START *Calling Peer ID* parameter value as the CM-logon service *Aircraft Address* parameter value,
 - 2) the D-START *DS User Version Number* parameter value as the CM-logon service *CM ASE Version Number* parameter value, and
 - 3) the APDU in the D-START *User Data* parameter as the CM-logon service *Logon Request* parameter value, and
- b) enter the *LOGON* state.

2.1.5.3.3.2.1.3 If the D-START *DS User Version Number* parameter value is equal to CM-ground-ASE version number the CM-ground-ASE shall:

- a) invoke CM-logon service indication with:
 - 1) the D-START *Calling Peer ID* parameter value as the CM-logon service *Aircraft Address* parameter value, and
 - 2) the APDU in the D-START *User Data* parameter as the CM-logon service *Logon Request* parameter value, and
- b) enter the *LOGON* state

2.1.5.3.3.2.2 Upon receipt of a D-START indication, if the receiving CM-ground-ASE is in the *IDLE* state and the APDU contained in the D-START *User Data* parameter is a [CMForwardRequest] APDU and the abstract value of the D-START *Calling Peer ID* parameter is a 4 to 8 character Facility Designation and the D-START *Priority* QOS parameter has the abstract value “flight regularity communications” and the D-START *RER* QOS parameter has the abstract value of “low” and the D-START *Routing Class* QOS parameter identifies the traffic category “Air Traffic Service Communications (ATSC)”, then:

2.1.5.3.3.2.2.1 If the CM-ground-ASE does not support the CM-forward service, the CM-ground-ASE shall:

- a) create a CMGroundMessage APDU with a cmForwardResponse [service-not-supported] APDU message element,
- b) invoke D-START response with:
 - 1) the APDU as the D-START *User Data* parameter value, and
 - 2) the abstract value “rejected (permanent)” as the D-START *Result* parameter value, and
- c) remain in the *IDLE* state.

2.1.5.3.3.2.2.2 If the D-START *DS User Version Number* parameter value is greater than the CM-ground-ASE version number and the CM-ground-ASE supports the CM-forward service, the CM-ground-ASE shall:

- a) create a CMGroundMessage APDU with a cmForwardResponse [incompatible-version] APDU message element,
- b) invoke D-START response with the following:
 - 1) the CM-ground-ASE version number as the D-START *DS User Version Number* parameter value, and
 - 2) the APDU as the D-START *User Data* parameter value,
 - 3) the abstract value of “rejected (permanent)” as the D-START *Result* parameter value, and
- c) remain in the *IDLE* state.

2.1.5.3.3.2.2.3 If the D-START *DS User Version Number* parameter value is less than the CM-ground-ASE version number and the CM-ground-ASE supports the CM-forward service, the CM-ground-ASE shall:

- a) invoke CM-forward service indication with the following:
 - 1) the D-START *Calling Peer ID* parameter value as the CM-forward service *Calling Facility Designation* parameter value,
 - 2) the D-START *DS User Version Number* parameter value as the CM-forward service *CM ASE Version Number* parameter value, and
 - 3) the APDU in the D-START *User Data* parameter as the CM-forward service *Forward Request* parameter value,
- b) create a CMGroundMessage APDU with a cmForwardResponse [success] APDU message element,
- c) invoke D-START response with the following:
 - 1) the APDU as the D-START *User Data* parameter value, and
 - 2) the abstract value of “rejected (permanent)” as the D-START *Result* parameter value, and
- d) remain in the *IDLE* state.

Note.— This assumes that CM ASEs are backwards compatible.

2.1.5.3.3.2.2.4 If the D-START *DS User Version Number* parameter value is equal to the CM-ground-ASE version number and the CM-ground-ASE supports the CM-forward service, the CM-ground-ASE shall:

- a) invoke CM-forward service indication with the following:
 - 1) the D-START *Calling Peer ID* parameter value as the CM-forward service *Calling Facility Designation* parameter value, and
 - 2) the APDU in the D-START *User Data* parameter as the CM-forward service *Forward Request* parameter value,
- b) create a CMGroundMessage APDU with a cmForwardResponse [success] APDU message element,
- c) invoke D-START response with the following:
 - 1) the APDU as the D-START *User Data* parameter value, and
 - 2) the abstract value of “rejected (permanent)” as the D-START *Result* parameter value, and
- d) remain in the *IDLE* state.

2.1.5.3.3.3 D-START Confirmation

2.1.5.3.3.3.1 Upon receipt of a D-START confirmation:

2.1.5.3.3.3.1.1 If the CM-ground-ASE is in the *UPDATE* state and the D-START *User Data* parameter is not provided, the CM-ground-ASE shall:

- a) stop timer t_{update} , and
- b) if the abstract value of the D-START *Result* parameter is “rejected (permanent)” and the abstract value of the D-START *Reject Source* parameter is “DS user”, enter the *IDLE* state.

2.1.5.3.3.3.1.2 If the CM-ground-ASE is in the *CONTACT* state and the APDU contained in the D-START *User Data* parameter is a [CMContactResponse] APDU, the CM-ground-ASE shall:

- a) stop timer t_{contact} , and
- b) if the abstract value of the D-START *Result* parameter is “rejected (permanent)” and the abstract value of the D-START *Reject Source* parameter is “DS user” then:
 - 1) invoke CM-contact service confirmation with the APDU in the D-START *User Data* parameter as the CM-contact *Contact Response* parameter value, and
 - 2) enter the *IDLE* state.

2.1.5.3.3.1.3 If the CM-ground-ASE is in the *FORWARD* state and if the D-START *User Data* parameter is a [CMForwardResponse] APDU, and the abstract value of the D-START *Result* parameter is “rejected (permanent)” and the abstract value of the D-START *Reject Source* parameter is “DS user”, the CM-ground-ASE shall:

- a) stop timer t_{forward} ,
- b) if the abstract value of the D-START *User Data* parameter is “service-not-supported”, the CM-ground-ASE will invoke a CM-forward service confirmation with the CM-forward *Result* parameter set to the abstract value “service-not-supported”,
- c) if the abstract value of the D-START *User Data* parameter is “incompatible-version”, the CM-ground-ASE will invoke a CM-forward service confirmation with the *DS User Version Number* parameter value as the CM-forward *CM ASE Version Number* parameter and set the CM-forward *Result* parameter abstract value to “incompatible-version”,
- d) if the abstract value of the D-START *User Data* parameter is “success”, the CM-ground-ASE will invoke a CM-forward service confirmation with the CM-forward *Result* parameter set to the abstract value “success”, and
- e) enter the *IDLE* state.

2.1.5.3.3.4 D-DATA Indication

2.1.5.3.3.4.1 Upon receipt of a D-DATA indication if the CM-ground-ASE is in the *CONTACT DIALOGUE* state and the APDU contained in the D-DATA *User Data* parameter is a [CMContactResponse] APDU, the CM-ground-ASE shall:

- a) stop timer t_{contact} ,
- b) invoke CM-contact service confirmation with the APDU contained in the D-DATA *User Data* parameter as the CM-contact *Contact Response* parameter value, and
- c) enter the *DIALOGUE* state.

2.1.5.3.3.5 D-END Confirmation

2.1.5.3.3.5.1 Upon receipt of a D-END confirmation, if the CM-ground-ASE is in the *END* state and the abstract value of the D-END *Result* is “accepted”, the CM-ground-ASE shall:

- a) stop timer t_{end} , and
- b) enter the *IDLE* state.

2.1.5.3.3.6 CM-logon Service Response

2.1.5.3.3.6.1 Upon receipt of a CM-logon service response, if the CM-ground-ASE is in the *LOGON* state the CM-ground-ASE shall:

- a) create a CMGroundMessage APDU with a cmLogonResponse APDU-element based on the CM-logon *Logon Response* parameter value, and
- b) invoke D-START response with the following:
 - 1) the CMGroundMessage APDU as the D-START *User Data* parameter value,
 - 2) if the CM-logon *Maintain Dialogue* parameter is provided by the CM-ground-user:
 - i) set the abstract value “accepted” as the D-START *Result* parameter value, and
 - ii) enter the *DIALOGUE* state.
 - 3) if the CM-logon *Maintain Dialogue* parameter is not provided by the CM-ground-user:
 - i) set the abstract value “rejected (permanent)” as the D-START *Result* parameter value, and
 - ii) enter the *IDLE* state.

2.1.5.3.3.7 CM-update Service Request

2.1.5.3.3.7.1 Upon receipt of a CM-update service request, if the CM-ground-ASE is in the *IDLE* state, the CM-ground-ASE shall:

- a) create a CMGroundMessage APDU with a cmUpdate APDU-element based on the CM-update *Update Information* parameter value,
- b) invoke D-START request with the following:
 - 1) the CM-update *Aircraft Address* parameter value as the D-START *Called Peer ID* parameter value,
 - 2) the CM-update *Facility Designation* parameter value as the D-START *Calling Peer ID* parameter value,

- 3) set the D-START *Quality of Service* parameter as follows:
 - i) if provided, the CM-update service *Class of Communication Service* parameter value as the D-START *Routing Class* parameter value,
 - ii) the abstract value of “flight regularity communications”, as the D-START *Priority* parameter value, and
 - iii) the abstract value of “low” as the D-START *RER* parameter value,
- 4) the CMGroundMessage APDU as the D-START *User Data* parameter value,
- c) start timer t_{update} , and
- d) enter the *UPDATE* state.

2.1.5.3.3.7.2 Upon receipt of a CM-update service request, if the CM-ground-ASE is in the *DIALOGUE* state, the CM-ground-ASE shall:

- a) create a CMGroundMessage APDU with a cmUpdate APDU-element based on the CM-update *Update Information* parameter value,
- b) invoke D-DATA request with the CMGroundMessage APDU as the D-DATA *User Data* parameter value, and
- c) remain in the *DIALOGUE* state.

2.1.5.3.3.8 CM-contact Service Request

2.1.5.3.3.8.1 Upon receipt of a CM-contact service request, if the CM-ground-ASE is in the *IDLE* state the CM-ground-ASE shall:

- a) create a CMGroundMessage APDU with a cmContactRequest APDU-element based on the CM-contact *Contact Request* parameter value,
- b) invoke D-START request with the following:
 - 1) the CM-contact *Aircraft Address* parameter value as the D-START *Called Peer ID* parameter value,
 - 2) the CM-contact *Facility Designation* parameter value as the D-START *Calling Peer ID* parameter value,

- 3) set the D-START *Quality of Service* parameters as follows:
 - i) if provided, the CM-contact service *Class of Communication Service* parameter value as the D-START *Routing Class* parameter value,
 - ii) The abstract value of “flight regularity communications”, as the D-START *Priority* parameter value, and
 - iii) The abstract value of “low” as the D-START *RER* parameter value,
- 4) the CMGroundMessage APDU as the D-START *User Data* parameter value,
- c) start timer t_{contact} and
- d) enter the *CONTACT* state.

2.1.5.3.3.8.2 Upon receipt of a CM-contact service request, if the CM-ground-ASE is in the *DIALOGUE* state the CM-ground-ASE shall:

- a) create a CMGroundMessage APDU with a cmContactRequest APDU-element based on the CM-contact *Contact Request* parameter value,
- b) invoke D-DATA request with the CMGroundMessage APDU as the D-DATA *User Data* parameter value,
- c) start timer t_{contact} and
- d) enter the *CONTACT DIALOGUE* state.

2.1.5.3.3.9 CM-end Service Request

2.1.5.3.3.9.1 Upon receipt of a CM-end service request, if the CM-ground-ASE is in the *DIALOGUE* state the CM-ground-ASE shall:

- a) invoke D-END request,
- b) start timer t_{end} , and
- c) enter the *END* state.

2.1.5.3.3.10 CM-forward Service Request

2.1.5.3.3.10.1 Upon receipt of a CM-forward service request, if the CM-ground-ASE is in the *IDLE* state, the CM-ground-ASE shall:

- a) create a CMGroundMessage APDU with a cmForwardRequest APDU-element based on the CM-forward service *Forward Request* parameter value,

- b) invoke D-START request with the following:
 - 1) the CM-forward *Called Facility Designation* parameter value as the D-START *Called Peer ID* parameter value,
 - 2) the CM-forward *Calling Facility Designation* parameter value as the D-START *Calling Peer ID* parameter value,
 - 3) the sending CM-ground-ASE version number as the D-START *DS User Version Number* parameter value,
 - 4) set the D-START *Quality of Service* parameter as follows:
 - i) if provided, the CM-forward service *Class of Communication Service* parameter value as the D-START *Routing Class* parameter value,
 - ii) the abstract value of “flight regularity communications” as the D-START *Priority* parameter value, and
 - iii) the abstract value of “low” as the D-START *RER* parameter value,
 - 5) the CMGroundMessage APDU as the D-START *User Data* parameter value,
- c) start timer t_{forward} , and
- d) enter the *FORWARD* state.

2.1.5.3.3.11 CM-user-abort Service Request

2.1.5.3.3.11.1 Upon receipt of a CM-user-abort service request, if the CM-ground-ASE is not in the *IDLE* state the CM-ground-ASE shall:

- a) stop any timer, if set,
- b) invoke D-ABORT request with the D-ABORT *Originator* parameter set to the abstract value “user”, and
- c) enter the *IDLE* state.

2.1.5.3.3.12 D-ABORT Indication

2.1.5.3.3.12.1 Upon receipt of a D-ABORT indication, if the CM-ground-ASE is not in the *IDLE* state the CM-ground-ASE shall:

- a) stop any timer, if set,

- b) if the CM-ground-user is an active user, then:
 - 1) if the D-ABORT *Originator* parameter contains the abstract value “user” invoke CM-user-abort service indication,
 - 2) else invoke CM-provider-abort service indication with the APDU contained in the D-ABORT *User Data* parameter as the CM-provider-abort service *Reason* parameter value, and
- c) enter *IDLE* state.

2.1.5.3.3.13 D-P-ABORT Indication

2.1.5.3.3.13.1 Upon receipt of a D-P-ABORT indication, if the CM-ground-ASE is not in the *IDLE* state the CM-ground-ASE shall:

- a) stop any timer, if set,
- b) if the CM-ground-user is an active user, invoke CM-provider-abort service indication with the CM-provider-abort *Reason* parameter set to the abstract value “communication-service-failure”, and
- c) enter the *IDLE* state.

2.1.5.4 Exception Handling

2.1.5.4.1 Timer Expiration

2.1.5.4.1.1 If a CM-ASE detects that a timer has expired, that CM-ASE shall:

- a) stop all timers,
- b) interrupt any current activity,
- c) if the CM-ASE is a CM-air-ASE, create a CMAircraftMessage APDU with a cmAbortReason [timer-expired] APDU message element,
- d) if the CM-ASE is a CM-ground-ASE, create a CMGroundMessage APDU with a cmAbortReason [timer-expired] APDU message element,
- e) invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter value,

- f) if the CM-user is an active user, invoke CM-provider-abort service indication with the abstract value “timer-expired” as the CM-provider-abort *Reason* parameter value, and
- g) enter the *IDLE* state.

2.1.5.4.2 Unrecoverable System Error

2.1.5.4.2.1 **Recommendation.**— *If a CM-ASE has an unrecoverable system error, the CM-ASE should:*

- a) *stop all timers,*
- b) *if the CM-ASE is a CM-air-ASE, create a CMAircraftMessage APDU with a cmAbortReason [undefined-error] APDU message element,*
- c) *if the CM-ASE is a CM-ground-ASE, create a CMGroundMessage APDU with a cmAbortReason [undefined-error] APDU message element,*
- d) *invoke D-ABORT request with:*
 - 1) *the abstract value “provider” as the D-ABORT Originator parameter value, and*
 - 2) *the APDU as the D-ABORT User Data parameter value,*
- e) *if the CM-user is an active user, invoke CM-provider-abort service indication with the abstract value “undefined-error” as the CM-provider-abort Reason parameter value, and*
- f) *enter the IDLE state.*

2.1.5.4.3 Invalid PDU

2.1.5.4.3.1 If the *User Data* parameter of a D-START indication or D-DATA indication does not contain a valid PDU as defined in 2.1.5.3.1.3 and 2.1.5.3.1.4, the CM-ASE shall:

- a) stop all timers,
- b) if the CM-ASE is a CM-air-ASE, create a CMAircraftMessage APDU with a cmAbortReason [invalid-PDU] APDU message element,
- c) if the CM-ASE is a CM-ground-ASE, create a CMGroundMessage APDU with a cmAbortReason [invalid-PDU] APDU message element,
- d) invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and

- 2) the APDU as the D-ABORT *User Data* parameter value,
- e) if the CM-user is an active user, invoke CM-provider-abort service indication with the abstract value “invalid-PDU” as the CM-provider-abort *Reason* parameter value, and
- f) enter the *IDLE* state.

2.1.5.4.3.2 If the *User Data* parameter of a D-START confirmation does not contain a valid PDU as defined in 2.1.5.3.1.3 and 2.1.5.3.1.4, the CM-ASE shall:

- a) stop all timers,
- b) if the CM-ASE is a CM-air-ASE and if the D-START *Result* parameter is set to the abstract value “accepted”, then
 - 1) create a CMAircraftMessage APDU with a cmAbortReason [invalid-PDU] APDU message element,
 - 2) invoke D-ABORT request with:
 - i) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - ii) the APDU as the D-ABORT *User Data* parameter value,
- c) if the CM-user is an active user, invoke CM-provider-abort service indication with the abstract value “invalid-PDU” as the CM-provider-abort *Reason* parameter value, and
- d) enter the *IDLE* state.

2.1.5.4.4 Not Permitted PDUor Dialogue Service Primitive

2.1.5.4.4.1 If the *User Data* parameter of a D-START indication or D-DATA indication is a valid PDU, but is not a permitted PDU as defined within 2.1.5.3.1.2, the CM-ASE shall:

- a) stop all timers,
- b) if the CM-ASE is a CM-air-ASE, create a CMAircraftMessage APDU with a cmAbortReason [protocol-error] APDU message element,
- c) if the CM-ASE is a CM-ground-ASE, create a CMGroundMessage APDU with a cmAbortReason [protocol-error] APDU message element,
- d) invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and

- 2) the APDU as the D-ABORT *User Data* parameter value,
- e) if the CM-user is an active user, invoke CM-provider-abort service indication with the abstract value “protocol-error” as the CM-provider-abort *Reason* parameter value, and
- f) enter the *IDLE* state.

2.1.5.4.4.2 If the *User Data* parameter of a D-START confirmation is a valid PDU, but is not a permitted PDU as defined within 2.1.5.3.1.2, the CM-ASE shall:

- a) stop all timers,
- b) if the D-START *Result* parameter is set to the abstract value “accepted”:
 - 1) if the CM-ASE is a CM-air-ASE, create a CMAircraftMessage APDU with a cmAbortReason [protocol-error] APDU message element,
 - 2) if the CM-ASE is a CM-ground-ASE, create a CMGroundMessage APDU with a cmAbortReason [protocol-error] APDU message element,
 - 3) invoke D-ABORT request with:
 - i) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - ii) the APDU as the D-ABORT *User Data* parameter value,
- c) if the CM-user is an active user, invoke CM-provider-abort service indication with the abstract value “protocol-error” as the CM-provider-abort *Reason* parameter value, and
- d) enter the *IDLE* state.

2.1.5.4.4.3 Upon receipt of a Dialogue service primitive for which there are no instruction in 2.1.5.3 (i.e. the primitive was not expected or was expected under other conditions or with other parameter values), the CM-ASE shall:

- a) stop all timers,
- b) if the CM-ASE is a CM-air-ASE, create a CMAircraftMessage APDU with a cmAbortReason [protocol-error] APDU message element,
- c) if the CM-ASE is a CM-ground-ASE, create a CMGroundMessage APDU with a cmAbortReason [protocol-error] APDU message element,

- d) if a dialogue exists, invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter value,
- e) if the CM-user is an active user, invoke CM-provider-abort service indication with the abstract value “protocol-error” as the CM-provider-abort *Reason* parameter value, and
- f) enter the *IDLE* state.

2.1.5.4.5 D-START Confirmation *Result* or *Reject Source* Parameter Values Not as Expected

2.1.5.4.5.1 If the CM-ground-ASE receives a D-START confirmation with the D-START *Result* parameter having the abstract value of “accepted”, the CM-ground-ASE shall:

- a) stop all timers,
- b) create a CMGroundMessage APDU with a cmAbortReason [dialogue-acceptance-not-permitted] APDU message element,
- c) invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter value,
- d) if the CM-ground-user is an active user, invoke CM-provider-abort service indication with the abstract value “dialogue-acceptance-not-permitted” as the CM-provider-abort *Reason* parameter value, and
- e) enter the *IDLE* state.

2.1.5.4.5.2 If the CM-ASE receives a D-START confirmation with the D-START *Result* parameter having the abstract value of “rejected (transient)” or if the D-START *Reject Source* parameter has the abstract value of “DS provider”, the CM-ASE shall:

- a) stop all timers, and
- b) if the CM-user is an active user, invoke CM-provider-abort service indication with the abstract value “communication-service-error” APDU as the CM-provider-abort *Reason* parameter value.

2.1.5.4.6 D-END Confirmation Not as Expected

2.1.5.4.6.1 If the CM-ground-ASE receives a D-END confirmation with the D-END *Result* parameter that does not have the abstract value of “accepted”, the CM-ground-ASE shall:

- a) stop all timers,
- b) create a CMGroundMessage APDU with a cmAbortReason [dialogue-end-not-accepted] APDU message element,
- c) invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter value, and
- d) enter the *IDLE* state.

2.1.5.4.7 D-START Indication *Quality of Service* Parameter Not as Expected

2.1.5.4.7.1 If the abstract value of the D-START *Priority* QOS parameter is not “flight regularity communications” or the abstract value of the D-START *RER* QOS parameter is not “low” or the abstract value of the D-START *Routing Class* QOS parameter does not identify the traffic category “Air Traffic Service Communications (ATSC)”, the CM-ASE shall:

- a) stop all timers
- b) if the CM-ASE is a CM-air-ASE, create a CMAircraftMessage APDU with a CMAbortReason [invalid-QOS-parameter] APDU message element,
- c) if the CM-ASE is a CM-ground-ASE, create a CMGroundMessage APDU with a CMAbortReason [invalid-QOS-parameter] APDU message element,
- d) invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter value, and
- e) enter the *IDLE* state

2.1.5.4.8 Expected PDU Not Provided

2.1.5.4.8.1 If the *User Data* parameter of a D-START indication, D-START confirmation (with the *Result* parameter set to the abstract value “accepted”), or D-DATA indication is not provided where it is expected, the CM-ASE shall:

- a) stop all timers,
- b) if the CM-ASE is a CM-air-ASE, create a CMAircraftMessage APDU with a cmAbortReason [expected-PDU-missing] APDU message element,
- c) if the CM-ASE is a CM-ground-ASE, create a CMGroundMessage APDU with a cmAbortReason [expected-PDU-missing] APDU message element,
- d) invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter value,
- e) invoke a CM-provider-abort service indication with the abstract value “expected-PDU-missing”, and
- f) enter the *IDLE* state.

2.1.5.4.8.2 If the *User Data* parameter of a D-START confirmation (with the *Result* parameter set to the abstract value “rejected (transient)” or “rejected (permanent)”) is not provided where it is expected, the CM-ASE shall:

- a) stop all timers,
- b) invoke a CM-provider-abort service indication with the abstract value “expected-PDU-missing”, and
- c) enter the *IDLE* state.

2.1.5.5 CM ASE State Tables

2.1.5.5.1 Priority

2.1.5.5.1.1 If the state tables for the CM-air-ASE and the CM-ground-ASE shown below conflict with textual statements made elsewhere in this document, the textual statements shall take precedence.

Note 1.— In the following state tables, the statement “cannot occur” means that if the implementation conforms to the SARPs, it is impossible for this event to occur. If the event does occur, this implies that there is an error in the implementation. If such a situation is detected, it is suggested that the ASE aborts with the error “unrecoverable system error”.

Note 2.— In the following state tables, the statement “not permitted” means that the implementation must prevent this event from occurring through some local means. If the event does occur this implies that there is an error in the implementation. If such a situation is detected, it is suggested that the ASE performs a local rejection of the request rather than aborting the dialogue.

Table 2.1.5-2. CM-Ground-ASE State Table

STATE ⇒ EVENT ↓	IDLE	LOGON	UPDATE	CONTACT	DIALOGUE	CONTACT DIALOGUE	END	FORWARD
DIALOGUE Service Events								
D-START Indication <i>Version Number</i> is greater than the CM-ground-ASE <i>version number</i> , ground-ground forwarding supported	●D-START response ⇒IDLE	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur
D-START Indication <i>Version Number</i> is less than or equal to the CM-ground-ASE <i>version</i> number, <i>User Data</i> = CMLogonRequest, ground-ground forwarding supported	●CM-logon indication ⇒LOGON	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur
D-START Indication <i>Version Number</i> is less than or equal to the CM-ground-ASE <i>version number</i> , <i>User Data</i> = CMForwardRequest, ground-ground forwarding supported	●CM-forward indication ●D-START response ⇒IDLE	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur
D-START Indication, Ground-ground forwarding is not supported, <i>User Data</i> = CMForwardRequest	●D-START response ⇒IDLE	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur
D-START Confirmation <i>Result</i> “rejected (permanent)” and <i>Reject Source</i> “DS user”, D-START <i>User Data</i> parameter not provided	cannot occur	cannot occur	●Stop timer t_{update} ⇒IDLE	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur
D-START Confirmation <i>Result</i> “rejected (permanent)” and <i>Reject Source</i> “DS user”, D-START <i>User Data</i> =CMContactResponse	cannot occur	cannot occur	cannot occur	●Stop timer $t_{contact}$ ●CM-contact confirmation ⇒IDLE	cannot occur	cannot occur	cannot occur	cannot occur

STATE ⇒ EVENT ↓	IDLE	LOGON	UPDATE	CONTACT	DIALOGUE	CONTACT DIALOGUE	END	FORWARD
D-START Confirmation Result "rejected (permanent)" and Reject Source "DS user", D-START User Data=CMForwardResponse	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	● Stop timer $t_{forward}$ ● CM-forward confirmation ⇒ IDLE
D-DATA Indication	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	● CM contact confirmation ● Stop timer $t_{contact}$ ⇒ DIALOGUE	cannot occur	cannot occur
D-END Confirmation Result "accepted"	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	● stop timer t_{end} ⇒ IDLE	cannot occur
CM-User Events								
CM-update Request	● D-START request ● Start timer t_{update} ⇒ UPDATE	not permitted	not permitted	not permitted	● D-DATA request ⇒ DIALOGUE	not permitted	not permitted	not permitted
CM-contact Request	● D-START request ● Start timer $t_{contact}$ ⇒ CONTACT	not permitted	not permitted	not permitted	● D-DATA request ● Start timer $t_{contact}$ ⇒ CONTACT ⇒ DIALOGUE	not permitted	not permitted	not permitted
CM-forward Request	● D-START request ● Start timer $t_{forward}$ ⇒ FORWARD	not permitted	not permitted	not permitted	not permitted	not permitted	not permitted	not permitted
CM-logon Response Maintain Dialogue not supplied by CM-ground user	not permitted	● D-START response ⇒ IDLE	not permitted	not permitted	not permitted	not permitted	not permitted	not permitted
CM-logon Response Maintain Dialogue "accepted"	not permitted	● D-START response ⇒ DIALOGUE	not permitted	not permitted	not permitted	not permitted	not permitted	not permitted
CM-end Request	not permitted	not permitted	not permitted	not permitted	● D-END request ● Start timer t_{end} ⇒ END	not permitted	not permitted	not permitted
ABORT Events								

STATE $\Rightarrow$ EVENT $\Downarrow$	IDLE	LOGON	UPDATE	CONTACT	DIALOGUE	CONTACT DIALOGUE	END	FORWARD
CM-user-abort Request	not permitted	●D-ABORT request $\Rightarrow$ IDLE	●stop timer t_{update} ●D-ABORT request $\Rightarrow$ IDLE	●stop timer $t_{contact}$ ●D-ABORT request $\Rightarrow$ IDLE	●D-ABORT request $\Rightarrow$ IDLE	●D-ABORT request Stop timer $t_{contact}$ $\Rightarrow$ IDLE	not permitted	●stop timer $t_{forward}$ ●D-ABORT request $\Rightarrow$ IDLE
D-ABORT Indication Originator is "provider"	cannot occur	●CM- provider- abort indication $\Rightarrow$ IDLE	●stop timer t_{update} ●CM-provi- der-abort indication $\Rightarrow$ IDLE	●stop timer $t_{contact}$ ●CM-provider- abort indication $\Rightarrow$ IDLE	●CM-provider- abort indication $\Rightarrow$ IDLE	●CM-provider- abort indication Stop timer $t_{contact}$ $\Rightarrow$ IDLE	●stop timer t_{end} $\Rightarrow$ IDLE	●stop timer $t_{forward}$ ●CM- provider- abort indication $\Rightarrow$ IDLE
D-ABORT Indication Originator is "user"	cannot occur	●CM-user- abort indication $\Rightarrow$ IDLE	●stop timer t_{update} ●CM-user- abort indication $\Rightarrow$ IDLE	●stop timer $t_{contact}$ ●CM-user-abort indication $\Rightarrow$ IDLE	●CM-user-abort indication $\Rightarrow$ IDLE	●CM-user-abort indication Stop timer $t_{contact}$ $\Rightarrow$ IDLE	●stop timer t_{end} $\Rightarrow$ IDLE	cannot occur
D-P-ABORT indication	cannot occur	●CM- provider- abort indication $\Rightarrow$ IDLE	●stop timer t_{update} ●CM-provi- der-abort indication $\Rightarrow$ IDLE	●stop timer $t_{contact}$ ●CM-provider- abort indication $\Rightarrow$ IDLE	●CM-provider- abort indication $\Rightarrow$ IDLE	●CM-provider- abort indication Stop timer $t_{contact}$ $\Rightarrow$ IDLE	●stop timer t_{end} $\Rightarrow$ IDLE	●stop timer $t_{forward}$ ●CM- provider- abort indication $\Rightarrow$ IDLE
T_{update} Expires	cannot occur	cannot occur	●D-ABORT request ●CM-provi- der-abort indication $\Rightarrow$ IDLE	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur
$T_{contact}$ Expires	cannot occur	cannot occur	cannot occur	●D-ABORT request ●CM-provider- abort indication $\Rightarrow$ IDLE	cannot occur	●D-ABORT request ●CM-provider- abort indication $\Rightarrow$ IDLE	cannot occur	cannot occur
T_{end} Expires	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	●D-ABORT request $\Rightarrow$ IDLE	cannot occur

STATE ⇒ EVENT ↓	IDLE	LOGON	UPDATE	CONTACT	DIALOGUE	CONTACT DIALOGUE	END	FORWARD
T _{forward} Expires	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur	●D-ABORT request ●CM-provider-abort indication ⇒IDLE

Table 2.1.5-3. CM-Air-ASE State Table

STATE ⇒ EVENT ↓	IDLE	LOGON	LOGON	CONTACT	DIALOGUE	CONTACT DIALOGUE
DIALOGUE Service Events						
D-START Indication User Data CMUUpdate	●CM-update indication ●D-START response ⇒IDLE	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur
D-START Indication User Data CMContactRequest	●CM-contact indication ⇒CONTACT	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur
D-START Confirmation Result "rejected (permanent)" and Reject Source "DS user"	cannot occur	●Stop timer t _{logon} ●CM-logon confirmation ⇒IDLE	cannot occur	cannot occur	cannot occur	cannot occur
D-START Confirmation Result "accepted"	cannot occur	●Stop timer t _{logon} ●CM-logon confirmation ⇒DIALOGUE	cannot occur	cannot occur	cannot occur	cannot occur
D-DATA Indication User Data CMUUpdate	cannot occur	cannot occur	cannot occur	cannot occur	●CM-update indication ⇒DIALOGUE	cannot occur
D-DATA Indication User Data CMContactRequest	cannot occur	cannot occur	cannot occur	cannot occur	●CM-contact indication ⇒CONTACT DIALOGUE	cannot occur
D-END Indication	cannot occur	cannot occur	cannot occur	cannot occur	●CM-end indication ●D-END response ⇒IDLE	cannot occur

STATE → EVENT ↓	IDLE	LOGON	CONTACT	DIALOGUE	CONTACT DIALOGUE
CM-User Events					
CM-contact Response	not permitted	not permitted	●D-START response ⇒IDLE	not permitted	●D-DATA request ⇒DIALOGUE
CM-logon Request	●D-START request ●Start timer $t_{\logon}$ ⇒LOGON	not permitted	not permitted	not permitted	not permitted
ABORT Events					
CM-user-abort Request	not permitted	●stop timer $t_{\logon}$ ●D-ABORT request ⇒IDLE	●D-ABORT request ⇒IDLE	●D-ABORT request ⇒IDLE	●D-ABORT request ⇒IDLE
D- ABORT Indication Originator is "provider"	cannot occur	●stop timer $t_{\logon}$ ●CM-provider-abort indication ⇒IDLE	●CM-provider-abort indication ⇒IDLE	●CM-provider-abort indication ⇒IDLE	●CM-provider-abort indication ⇒IDLE
D-ABORT Indication Originator is "user"	cannot occur	●stop timer $t_{\logon}$ ●CM-user-abort indication ⇒IDLE	●CM-user-abort indication ⇒IDLE	●CM-user-abort indication ⇒IDLE	●CM-user-abort indication ⇒IDLE
D-P-ABORT indication	cannot occur	●stop timer $t_{\logon}$ ●CM-provider-abort indication ⇒IDLE	●CM-provider-abort indication ⇒IDLE	●CM-provider-abort indication ⇒IDLE	●CM-provider-abort indication ⇒IDLE
$T_{\logon}$ Expires	cannot occur	●D-ABORT request ●CM-provider-abort indication ⇒IDLE	cannot occur	cannot occur	cannot occur

2.1.6 COMMUNICATION REQUIREMENTS

2.1.6.1 Encoding Rules

2.1.6.1.1 The CM application shall use PER encoding as defined in ISO/IEC 8825-2, using the Basic Unaligned variant to encode/decode the ASN.1 message structure and content specified in 2.1.4.

2.1.6.2 Dialogue Service Requirements

2.1.6.2.1 Primitive Requirements

2.1.6.2.1.1 Where dialogue service primitives, that is D-START, D-DATA, D-END, D-ABORT, and D-P-ABORT are described as being invoked in 2.1.5, the CM-ground-ASE and the CM-air-ASE shall exhibit external behaviour consistent with the dialogue service, as described in 4.2, having been implemented and its primitives invoked.

2.1.6.2.2 ATN Quality-of-Service Requirements

2.1.6.2.2.1 The *Priority* Quality-of-Service parameter of the D-START for CM shall be the abstract value of “flight regularity communications”.

2.1.6.2.2.2 The *RER* Quality-of-Service parameter of the D-START for CM shall be set to the abstract value of “low”.

2.1.6.2.2.3 The CM-ASE shall map the Class of Communication Service abstract values to the Routing Class abstract value part of the D-START Quality-of-Service parameter as presented in Table 2.1.6-1.

Table 2.1.6-1. Mapping Between Class of Communication and Routing Class Abstract Values

Class of Communication Abstract Value	Routing Class Abstract Value
A	Traffic follows Class A ATSC route(s)
B	Traffic follows Class B ATSC route(s)
C	Traffic follows Class C ATSC route(s)
D	Traffic follows Class D ATSC route(s)
E	Traffic follows Class E ATSC route(s)
F	Traffic follows Class F ATSC route(s)
G	Traffic follows Class G ATSC route(s)
H	Traffic follows Class H ATSC route(s)

Note.— ATSC values are defined in 1.3.

2.1.7 CM USER REQUIREMENTS

Note.— Requirements imposed on CM-users concerning CM messages and interfacing with the CM-ASEs are presented in 2.1.7.

2.1.7.1 CM-Air-User Requirements

Note 1.— When a CM-air-user invokes the CM-logon service and requires a particular class of communication service, it sets the Class of Communication Service parameter to be the class of communication it requires.

Note 2.— When the CM-air-user invokes a CM-logon service and has no preference for the class of communication service to be used, the Class of Communication Service parameter does not need to be provided.

Note 3.— For each CM-air-ASE invocation, the CM-air-user establishes a correlation between a CM-air-ASE invocation and the facility designation.

Note 4.— Upon the initiation of a CM-logon service request, or upon receipt of a CM-update service indication or a CM-contact service indication, the ASE invocation correlation is based on the facility designation in the Facility Designation parameter of the respective CM service.

Note 5.— The correlation is maintained for the duration of the ASE invocation.

2.1.7.1.1 CM-logon Service Requirements

Note.— Only the CM-air-user is permitted to initiate the CM-logon service.

2.1.7.1.1.1 When invoking the CM-logon service request the CM-air-user shall provide the following as part of the CMLogonRequest:

- a) its CM long TSAP,
- b) the aircraft's Flight ID,
- c) information on each application for which it requires a data link service as follows:
 - 1) for air-only initiated services: application name and version number for all the versions that can be supported, and
 - 2) for applications that can be ground initiated: application name, version number, and address for all the versions that can be supported,
- d) the facility designation of the facility with which the CM-air-user wishes to exchange application information (if required), and
- e) flight information data as required by the ground system.

Note.— The facility designation is only used if the CM-air-user wants to explicitly identify a facility other than the one contained in the Facility Designation parameter of the CM-logon service for which it requires application information.

2.1.7.1.1.2 **Recommendation.** — The CM-air-user should only use the facilityDesignation field of the CMLogonRequest if requesting information for a facility other than the one specified in the Facility Designation parameter of the CM-logon service.

2.1.7.1.1.3 When invoking the CM-logon service request, if any RDP for a given application address is different than the CM RDP, the CM-air-user shall use the long TSAP for each application address provided.

Note.— The long TSAP = RDP + short TSAP. The short TSAP = ARS + LOC + SYS + NSEL + TSEL. The RDP = VER + ADM + RDF.

2.1.7.1.1.4 When invoking the CM-logon service request, the ARS component of the short TSAP shall contain the aircraft's 24 bit address.

Note.— If there is more than one routing domain on the aircraft, the LOC field is used to differentiate them.

2.1.7.1.1.5 Upon receipt of a CM-logon service confirmation, the CM-air-user shall create the actual TSAP for each ground application information contained in the Logon Response based on the IDP and long TSAP for each application as defined in 2.1.4.

Note.— The actual TSAP = IDP + long TSAP. The IDP = AFI + IDI.

2.1.7.1.1.6 Upon receipt of a CM-logon service confirmation, the CM-air-user shall make the information contained in the CMLogonResponse available to the other applications (i.e., ADS, CPDLC, and FIS), as well as to the dialogue service provider.

2.1.7.1.1.7 Upon the receipt of a Logon Response from a CM-logon service confirmation from a ground facility for which CM information has previously been received, the CM-air-user shall only replace the previous information for which new logon information has been received.

Note 1.— If the facilityDesignation field of the LogonRequest was provided, then the information contained in the Logon Response parameter corresponds to that facility designation. If the facilityDesignation field of the LogonRequest was not provided, then the information contained in the LogonResponse parameter corresponds to the Facility Designation parameter of the CM-logon service.

Note 2.— If the facilityDesignation field of the LogonRequest was provided and no application information is returned in the Logon Response parameter, this means that the CM-ground-user does not have access to the information for the requested facility or that the requested facility does not support any of the proposed applications.

2.1.7.1.2 CM-update Service Requirements

Note.— If a CM-update service indication is received, then the information contained in the Update Information parameter corresponds to the facility designation contained in the Facility Designation parameter, or, if a dialogue exists, the facility with which the dialogue is in place.

2.1.7.1.2.1 Upon the receipt of *Update Information* from a CM-update service indication from a ground facility designation for which CM information has previously been received, the CM-air-user shall only replace the previous information for which updated information has been received.

2.1.7.1.2.2 Upon receipt of a CM-update service indication, the CM-air-user shall create the actual TSAP for each ground application information contained in the *Update Information* based on the IDP and long TSAP for each application as defined in 2.1.4.

Note.— *The actual TSAP = IDP + long TSAP. The IDP = AFI + IDI.*

2.1.7.1.2.3 The CM-air-user shall make the updated information contained in the *Update Information* available to the other applications (i.e., ADS, CPDLC, and FIS), as well as to the dialogue service provider.

2.1.7.1.3 CM-contact Service Requirements

2.1.7.1.3.1 **Recommendation.** — *Upon receipt of a CM-contact indication, the CM-air-user should invoke the CM-logon request with the indicated ground system within 0.5 seconds.*

2.1.7.1.3.2 Upon receipt of a CM-logon confirmation when performing the CM-contact service, the CM-air-user shall invoke a CM-contact response.

2.1.7.1.3.3 **Recommendation.** — *Upon receipt of a CM-logon confirmation when performing the CM-contact service, the CM-air-user should invoke a CM-contact response within 0.5 seconds.*

2.1.7.1.3.4 Upon receipt of a CM-contact service indication, the CM-air-user shall attempt to initiate a CM-logon service request with the indicated ground system.

Note.— *If a CM-logon service request is initiated, the CM-air-user will comply with the CM-logon requirements as stated in 2.1.7.1.1. However, the facility designation will not be provided as part of the CMLogonRequest in this case.*

2.1.7.1.3.5 In addition to the above CM-logon service requirements, upon receipt of a CM-logon service response from the indicated facility designation, or if no CM-logon service request can be initiated, the CM-air-user shall invoke the CM-contact service response indicating the success or lack thereof of the CM-logon service request.

2.1.7.2 CM-Ground-User Requirements

2.1.7.2.1 General CM-Ground-User Requirements

2.1.7.2.1.1 A CM-ground-user shall invoke the CM-logon service, CM-update service, CM-contact service, and CM-end service only when communicating with a CM-air-user.

2.1.7.2.1.2 A CM-ground-user shall invoke the CM-forward service only when communicating with another CM-ground-user.

Note 1.— *When a CM-ground-user invokes the CM-update service, CM-contact service, or CM-forward service and requires a particular class of communication service, it will set the Class of Communication Service parameter to be the class of communication it requires.*

Note 2.— When the CM-ground-user invokes a CM-update service, CM-contact service, or CM-forward service and has no preference for the class of communication service to be used, the Class of Communication Service parameter does not need to be provided.

Note 3.— When a CM-ground-user specifies the Class of Communication Service parameter and the dialogue is in place, the class of communication parameter is ignored.

Note 4.— For each CM-ground-ASE invocation, the CM-ground-user establishes a correlation between a CM-ground-ASE invocation and the aircraft 24 bit address

Note 5.— Upon the initiation a CM-update service request or CM-contact service request, or upon receipt of a CM-logon service indication the ASE invocation correlation is based on the 24-bit aircraft address in the Aircraft Address parameter of the respective CM service.

Note 6.— The correlation is maintained for the duration of the ASE invocation.

2.1.7.2.2 CM-logon Service Requirements

2.1.7.2.2.1 Recommendation. — *Upon receipt of a CM-logon indication, the CM-ground-user should invoke the CM-logon response within 0.5 seconds.*

2.1.7.2.2.2 Upon receipt of a CM-logon service indication, the CM-ground-user shall make the aircraft application information contained in the *Logon Request* available to the other applications (i.e., ADS, CPDLC, and FIS), as well as to the dialogue service provider.

2.1.7.2.2.3 Upon receipt of a CM-logon service indication, the CM-ground-user shall create the actual TSAP for each aircraft application information contained in the *Logon Request* based on the IDP and long TSAP for each application as defined in 2.1.4.

Note.— The actual TSAP = IDP + long TSAP. The IDP = AFI + IDI.

2.1.7.2.2.4 Upon the receipt of a *Logon Request* from a CM-logon service indication from an aircraft for which CM information has previously been received and still being maintained, the CM-ground-user shall update the aircraft information accordingly.

2.1.7.2.2.5 Upon receipt of a CM-logon service indication, the CM-ground-user shall invoke a CM-logon service response with a CMLogonResponse containing:

- a) application names, addresses, and version numbers for the requested applications that can be air-initiated for all versions that the ground and aircraft systems can support, and
- b) application names and version numbers for the requested ground-only initiated applications that the ground system can support.

2.1.7.2.2.6 If the facility designation is present in the *Logon Request* parameter, then the application information contained in the *CM LogonResponse* shall correspond to that facility designation.

Note.— If a CM-ground-user does not have access to the information for the requested facility, no application information is returned.

2.1.7.2.2.7 If the facility designation is not present in the *Logon Request* parameter, then the application information contained in the *LogonResponse* shall correspond to the responding CM-ground-user's facility designation.

2.1.7.2.2.8 When invoking the CM-logon service response, if any RDP for a given application address is different than the CM RDP, the CM-ground-user shall use the long TSAP for each application address provided.

Note 1.— The long TSAP = RDP + short TSAP. The short TSAP = ARS (optional) + LOC + SYS + NSEL + TSEL. The RDP = VER + ADM + RDF.

Note 2.— If there is more than one routing domain on the ground, the ARS field is used to differentiate them. If there is not more than one routing domain on the ground, the ARS field need not be used.

Note 3.— The value of the ARS field is a 24-bit unsigned binary number that uniquely identifies the addressed system in a single routing domain and is assigned by the State or Organization identified in the ADM field.

2.1.7.2.2.9 When the CM-ground-user requires a CM dialogue to be maintained, the CM-ground-user shall set the CM-logon service response *Maintain Dialogue* parameter, if and only if the dialogue maintain service is supported.

2.1.7.2.3 CM-update Service Requirements

Note.— Only the CM-ground-user is permitted to initiate the CM-update-service.

2.1.7.2.3.1 When invoking the CM-update service request, the CM-ground-user shall provide a CMUpdate containing application names, addresses, and version numbers for each of the data link applications being updated.

Note.— The CM-update service only corresponds to a ground facility's local applications.

2.1.7.2.3.2 When invoking the CM-update service request, the CM-ground-user shall use the Long TSAP for each application address provided.

2.1.7.2.4 CM-contact Service Requirements

Note.— Only the CM-ground-user is permitted to initiate the CM-contact-service.

2.1.7.2.4.1 When invoking the CM-contact service request, the CM-ground-user shall provide a CMContactRequest containing the facility designation of the ground facility that the ground requests the aircraft to contact.

2.1.7.2.5 CM-end Service Requirements

Note 1.— Only the CM-ground-user is permitted to initiate the CM-end-service.

Note 2.— If the CM-ground-user establishes a CM dialogue with the CM-logon Maintain Dialogue parameter set, the CM-ground user is responsible for closing the CM dialogue with the CM-end service.

2.1.7.2.6 CM-forward Service Requirements

Note.— Only the CM-ground-user is permitted to initiate the CM-forward-service.

2.1.7.2.6.1 When requesting the CM-forward service, the CM-ground-user shall provide all of the information from either a CM-logon request message or a CM-forward request message, whichever is the more recent.

2.1.7.2.6.2 Upon receipt of a CM-forward service indication, the CM-ground-user shall make the aircraft application information contained in the *Forward Request* available to the other applications (i.e., ADS, CPDLC, and FIS), as well as to the dialogue service provider.

2.1.7.2.6.3 Upon receipt of a CM-forward service indication, the CM-ground-user shall create the actual TSAP for each aircraft application information contained in the *Forward Request* based on the IDP and long TSAP for each application as defined in 2.1.4.

Note.— The actual TSAP = IDP + long TSAP. The IDP = AFI + IDI.

2.1.7.2.6.4 Upon the receipt of a *Forward Request* from a CM-forward service indication concerning an aircraft identifier for which CM information has previously been received and is still being maintained, the CM-ground-user shall update the aircraft information accordingly.

2.1.7.2.6.5 **Recommendation.** — *Upon the receipt of a Forward Request from a CM-forward service indication, the receiving CM-ground-user should invoke a CM-update service request with the indicated aircraft, if the update service is supported.*

2.1.7.3 Parameter Value Unit, Range and Resolution

2.1.7.3.1 A CM user shall interpret parameter value unit, range and resolution as defined in 2.1.4.

2.1.8 SUBSETTING RULES

2.1.8.1 General

Note.— 2.1.8.1.1 specifies conformance requirements which all implementations of the CM protocol obey.

2.1.8.1.1 An implementation of either the CM ground based service or the CM air based service claiming conformance shall support the CM protocol features as shown in the table below.

Note 1.— The ‘status’ column indicates the level of support required for conformance to the CM-ASE protocol. The values are as follows:

- a) **‘M’** mandatory support is required,
- b) **‘O’** optional support is permitted for conformance to the CM protocol,
- c) **‘N/A’** the item is not applicable, and
- d) **‘C.n’** the item is conditional where n is the number which identifies the condition which is applicable.

Table 2.1.8-1. CM Protocol Versions Implemented

	Status	Associated Predicate
Version 1	M	none

Table 2.1.8-2. CM Operational Functional Units

	Status	Associated Predicate
The CM system acts as an airborne system	C.1	CM/air
The CM system acts as a ground system	C.1	CM/ground
The ground CM system can update application information	if (CM/ground) O, else N/A	G-UP-FU
The ground CM system can request the aircraft to initiate a logon with a specified CM ground system	if (CM/ground) O, else N/A	G-CO-FU

	Status	Associated Predicate
The ground CM user can process forwarded aircraft information	if (CM/ground) O, else N/A	G-FO-FU
The ground CM system can forward aircraft information to another CM ground system	if (CM/ground) O, else N/A	G-FO-IN
The air CM user can process an update request	if (CM/air) O, else N/A	A-UP-FU
The air CM user can process an contact request	if (CM/air) O, else N/A	A-CO-FU

C.1: a conforming implementation will support one and only one of these two options.

Table 2.1.8-3. CM Ground Configurations

	List of Predicates	Functionality Description
I	CM/ground	Logon exchanges are supported with an aircraft. Received forward requests are rejected with reason 'service not supported'.
II	CM/ground + G-FO-IN	Logon exchanges are supported with an aircraft. Forward exchanges initiated by the local ground system user are supported. Received forward requests are rejected with reason 'service not supported'.
III	CM/ground + G-FO-FU	Logon exchanges are supported with an aircraft. Received forward requests from peer CM ground systems are processed by the local CM user.
IV	CM/ground + G-FO-IN + G-FO-FU	Logon exchanges are supported with an aircraft. Forward exchanges initiated by the local ground system user are supported. Received forward requests from peer CM ground systems are processed by the local CM user.
V	CM/ground + G-UP-FU	Same as subset I plus the capability of a CM ground system to update application information on an aircraft.
VI	CM/ground + G-UP-FU + G-FO-IN	Same as subset II plus the capability of a CM ground system to update application information on an aircraft.

	List of Predicates	Functionality Description
VII	CM/ground + G-UP-FU + G-FO-FU	Same as subset III plus the capability of a CM ground system to update application information on an aircraft.
VIII	CM/ground + G-UP-FU + G-FO-IN + G-FO-FU	Same as subset IV plus the capability of a CM ground system to update application information on an aircraft.
IX	CM/ground + G-CO-FU	Same as subset I plus the capability of a CM ground system to request an aircraft to perform a logon with a specified CM ground system.
X	CM/ground + G-CO-FU + G-FO-IN	Same as subset II plus the capability of a CM ground system to request an aircraft to perform a logon with a specified CM ground system.
XI	CM/ground + G-CO-FU + G-FO-FU	Same as subset III plus the capability of a CM ground system to request an aircraft to perform a logon with a specified CM ground system.
XII	CM/ground + G-CO-FU + G-FO-IN + G-FO-FU	Same as subset IV plus the capability of a CM ground system to request an aircraft to perform a logon with a specified CM ground system.
XIII	CM/ground + G-CO-FU + G-UP-FU	Same as subset I plus the capability of a CM ground system to request an aircraft to perform a logon with a specified CM ground system and the capability of a CM ground system to update application information on an aircraft.
XIV	CM/ground + G-CO-FU + G-UP-FU + G-FO-IN	Same as subset II plus the capability of a CM ground system to request an aircraft to perform a logon with a specified CM ground system and the capability of a CM ground system to update application information on an aircraft.
XV	CM/ground + G-CO-FU + G-UP-FU + G-FO-FU	Same as subset III plus the capability of a CM ground system to request an aircraft to perform a logon with a specified CM ground system and the capability of a CM ground system to update application information on an aircraft.

	List of Predicates	Functionality Description
XVI	CM/ground + G-CO-FU + G-UP-FU + G-FO-IN + G-FO-FU	Same as subset IV plus the capability of a CM ground system to request an aircraft to perform a logon with a specified CM ground system and the capability of a CM ground system to update application information on an aircraft.

Note 2.— A CM ground system may or may not support the maintain dialogue feature.

Table 2.1.8-4. CM Air Configurations

	List of Predicates	Functionality Description
I	CM/air	Logon exchanges can be initiated with ground systems. Received contact requests are rejected with reason 'contact not successful'. Received update requests are processed by the CM-air-ASE but ignored by the CM-air-user.
II	CM/air + A-UP-FU	Logon exchanges can be initiated with ground systems. Received contact requests are rejected with reason 'contact not successful'. Received update requests are processed by the CM-air-ASE and the CM-air-user.
III	CM/air + A-CO-FU	Logon exchanges can be initiated with ground systems. Received update requests are processed by the CM-air-ASE but ignored by the CM-air-user. Received contact requests are processed and acted upon by the CM-air-ASE and the CM-air-user.
IV	CM/air + A-UP-FU + A-CO-FU	Logon exchanges can be initiated with ground systems. Received update requests are processed by the CM-air-ASE and the CM-air-user. Received contact requests are processed and acted upon by the CM-air-ASE and the CM-air-user.

Note 3.— A CM air system must support the maintain dialogue feature.

2.2 AUTOMATIC DEPENDENT SURVEILLANCE APPLICATIONS

Note.— Structure of 2.2: 2.2.1 defines the air-ground communication aspects of ADS. 2.2.2 defines the ground-ground (i.e. ADS report forwarding) aspects of ADS.

2.2.1 AUTOMATIC DEPENDENT SURVEILLANCE APPLICATION

2.2.1.1 Introduction

2.2.1.1.1 The ADS air ground application will allow users to obtain positional and other information from suitably equipped aircraft in a timely manner in accordance with their requirements. The ADS application is designed to give automatic reports about aircraft to a user. The ADS reports give positional as well as other information likely to be of use to the air traffic management function, including air traffic control. The aircraft provides the information to the user under the following circumstances:

- a) under a contract (known as a demand contract) agreed with the ground system, the aircraft provides the information immediately and once only;
- b) under a contract (known as a periodic contract) agreed with the ground system, the aircraft provides information on a regular basis;
- c) under a contract (known as an event contract) agreed with the ground system, the aircraft provides information when certain events are detected by the avionics;
- d) under emergency conditions the aircraft provides information on a regular basis with no prior agreement with the ground system (known as an emergency contract).

Note 1.— Structure of 2.2: This chapter defines the air-ground communication aspects of ADS only.

- a) 2.2.1.1: *INTRODUCTION* contains 2.2.1 's purpose, structure, and a summary of the functions of ADS.
- b) 2.2.1.2: *GENERAL REQUIREMENTS* contains backwards compatibility and error processing requirements.
- c) 2.2.1.3: *THE ABSTRACT SERVICE* contains the description of the abstract service provided by the application service elements (ASE) defined for ADS.
- d) 2.2.1.4: *FORMAL DEFINITION OF MESSAGES* contains the formal definition of messages exchanged by ADS-ASEs using Abstract Syntax Notation Number One (ASN.1).
- e) 2.2.1.5: *PROTOCOL DEFINITION* describes the exchanges of messages allowed by the ADS protocol, as well as time constraints and the exception handling procedures associated with these exchanges. 2.2.1 describes also the ADS protocol in terms of state tables.
- f) 2.2.1.6: *COMMUNICATION REQUIREMENTS* contains the requirements that the ADS ASE application imposes on the underlying communication system.

- g) 2.2.1.7: *ADS USER REQUIREMENTS* outlines the requirements that a user of an ADS ASE must meet.
- h) 2.2.1.8: *SUBSETTING RULES* provides rules for subsetting the ADS SARPs.

Note 2.— General Functionality

- a) *The avionics are capable of supporting contracts with at least four ATC ground systems simultaneously; they are also capable of supporting one demand, one event and one periodic contract with each ground system simultaneously.*
- b) *In addition if the pilot or avionics elects, the avionics will suspend any existing periodic contract, and establishes an emergency contract with each ground system with which it has an ADS contract.*
- c) *It will be necessary for an implementation to provide information which is both accurate and timely in the ADS reports, however, quantification of the age and accuracy of the information is beyond the scope of 2.2.1.*

Note 3.— Establishment and Operation of a Demand Contract

- a) *Functional Description*
 - 1) *This function allows the ground system to establish a demand contract with an aircraft, and then for the conditions of that contract to be realised. Realisation of the contract involves the sending of a single report from an aircraft to the ground system.*
 - 2) *Any number of demand contracts may be sequentially established with an aircraft. Basic information is sent with the report. Optionally, at the request of the ground system, other information may also be sent.*
 - 3) *The ground system sends a demand contract request to the avionics. This contains an indication of which optional information blocks are required. The avionics then determines whether or not there are errors in the request, and if there are no errors, whether or not it is able to comply with the request. If the avionics can comply with the demand contract request it sends the report as soon as possible. If there are errors in the contract request, or if the avionics cannot comply with the request, it sends a negative acknowledgement to the ground system indicating the reason for its inability to accept the contract. If the avionics can partially comply with the request, it sends a non-compliance notification indicating those parts of the contract with which it cannot comply, and then it sends an ADS-report.*

b) Message Descriptions

- 1) The demand contract stipulates which of the optional information fields are to be included in the ADS report.*
- 2) Each ADS-report always contains the following basic information:*
 - i) the 3-D position of the aircraft;*
 - ii) the time;*
 - iii) an indication of the accuracy of the positional information (figure of merit).*
- 3) Optionally, an ADS-report contains an indication of:*
 - i) the aircraft address;*
 - ii) the projected profile, indicating the position and predicted time of the next way point, and the position of the following way point;*
 - iii) the ground vector, indicating the track, ground speed and vertical rate;*
 - iv) the air vector, indicating the heading, air speed and vertical rate;*
 - v) weather information, indicating wind speed, wind direction, temperature and turbulence;*
 - vi) the short term intent, indicating the predicted location of the aircraft at some time in the future (as indicated in the demand contract) and, for any intermediate points where level, track or speed change is predicted to occur, the projected distance, track, level and time are given;*
 - vii) extended project profile, indicating the predicted position, level and time for the next several way points (as indicated in the demand contract).*
- 4) An ADS report can contain a positive acknowledgement indicating acceptance of the contract.*
- 5) A negative acknowledgement contains an indication of the reason why the contract has not been accepted.*
- 6) A non-compliance notification contains an indication of which optional information fields cannot be sent.*

*Note 4.— Establishment and Operation of an Event Contract**a) Functional Description*

- 1) This function allows the ground system to establish an event contract with the aircraft, and then for the conditions of that contract to be realised. Realisation of the contract involves the sending of reports from the aircraft to the ground system when certain agreed events occur.*
- 2) Only one event contract may exist between the ground system and avionics at any one time, but this may contain multiple event types. A set of basic information is sent with every report, and depending on the event that triggered the sending of the report, other information blocks may also be included. The contract that is agreed states the event types that are to trigger reports and also any values needed to clarify those event types.*
- 3) It is possible to request one or more of the following event types:*
 - i) Vertical rate change. This can be triggered in two ways. If the vertical rate threshold is positive, then the event is triggered when the aircraft's rate of climb is greater than the vertical rate threshold. If the vertical rate threshold is negative, then the event is triggered when the aircraft's rate of descent is less than the vertical rate threshold.*
 - ii) Way-point change. This is triggered by a change to the next way-point. This change is normally due to routine way point sequencing, but could be triggered by a way point which is not part of the ATC clearance but is entered by the pilot for operational reasons.*
 - iii) Lateral deviation change. This is triggered when the absolute value of the lateral distance between the aircraft's actual position and the aircraft's expected position on the active flight plan becomes greater than the lateral deviation threshold.*
 - iv) Level range deviation. This is triggered when the aircraft's level becomes greater than the level ceiling or less than the level floor.*
 - v) Airspeed change. This is triggered when the aircraft's airspeed differs negatively or positively from its value at the time of the previous ADS report containing an air vector, by an amount which is equal to the airspeed change threshold which is specified in the event contract request. If there has been no previous such report, one is sent immediately.*
 - vi) Ground speed change. This is triggered when the aircraft's ground speed differs negatively or positively from its value at the time of the previous ADS report containing a ground vector, by an amount which is equal to the ground speed threshold which is specified in*

the event contract request. If there has been no previous such report, one is sent immediately.

- vii) Heading change. This is triggered when the aircraft's heading differs negatively or positively from its value at the time of the previous ADS report containing an air vector, by an amount which is equal to the heading change threshold which is specified in the event contract request. If there has been no previous such report, one is sent immediately.*
 - viii) Extended projected profile change. This is triggered by a change to any of the set of future way points that define the active route of flight. The number of way points covered in the contract is either defined by a time interval (i.e. any way point planned to be achieved in the next N minutes), or by number of way points (i.e. any way point in the next N).*
 - ix) FOM (Figure of Merit) change. This is triggered by a change in the navigational accuracy, navigational system redundancy or airborne collision avoidance system (ACAS) availability.*
 - x) Track angle change. This is triggered when the aircraft's track angle differs negatively or positively from its value at the time of the previous ADS report containing a ground vector, by an amount which is equal to the track angle change threshold which is specified in the event contract request. If there has been no previous such report, one is sent immediately.*
 - xi) Level change. This is triggered when the aircraft's level differs negatively or positively from its value at the time of the previous ADS report, by an amount which is equal to the level change threshold which is specified in the event contract request. If there has been no previous such report, one is sent immediately.*
- 4) Acceptance of an event contract request implicitly cancels an existing event contract, if one exists.*
 - 5) The ground system sends an event contract request to the avionics. This contains the types of event to be reported on and the necessary parameters for that event (e.g. if the event is a level range deviation, then the upper and lower thresholds must be sent). The avionics then determines whether or not there are errors in the request, and if not, whether or not it is able to comply with the request. If the avionics can comply with the event contract request it sends a positive acknowledgement and any required baseline report. If the contracted event occurs, an ADS report is sent.*

- 6) *If there are errors in the event contract request, or if the avionics cannot comply with the request, it sends a negative acknowledgement to the ground system indicating the reason for its inability to accept the contract within 0.5 seconds.*
- 7) *If the avionics can partially comply with the request, it sends a non-compliance notification indicating those parts of the contract with which it cannot comply. If a contracted event occurs with which it can comply, an ADS-report is sent.*
- 8) *For lateral deviation, level range and vertical rate change, if the event occurs, a report is sent every 60 seconds while the limit(s) specified in the contract are exceeded. For all other events, a single report is sent every time the event occurs.*

b) Message Descriptions

- 1) *The event contract request contains an indication of the events to be reported on, together with clarifying information as follows:*
 - i) *lateral deviation change - containing the lateral deviation threshold;*
 - ii) *vertical rate change - containing the vertical rate threshold;*
 - iii) *leaving a given level range - containing the upper and lower level thresholds;*
 - iv) *way-point change - containing no further clarifying information;*
 - v) *air speed change - containing the airspeed change threshold;*
 - vi) *ground speed change - containing ground speed change threshold;*
 - vii) *heading change - containing heading change threshold;*
 - viii) *extended projected profile change - containing either a projected time or a number of way points;*
 - ix) *figure of merit change - containing no further clarifying information;*
 - x) *track angle change - containing the track angle change threshold;*
 - xi) *level change - containing level change range.*

- 2) *The ADS report has the same structure as in the operation of a demand contract, containing position, time and FOM. However the choice of additional optional information blocks is made as follows:*
 - i) *if the triggering event is a vertical rate change, a lateral deviation change, a level deviation change, a ground speed change, a track angle change or a level change, then the ADS report will contain the ground vector;*
 - ii) *if the triggering event is a way point change, then the ADS report will contain the projected profile;*
 - iii) *if the triggering event is an air speed change or heading change, then the ADS report will contain the air vector;*
 - iv) *if the triggering event is an extended projected profile change, then the ADS report will contain the extended projected profile;*
 - v) *if the triggering event is a FOM change, then the ADS report will contain no additional information other than the basic information contained in every ADS report).*
- 3) *An ADS report can contain a positive acknowledgement indicating acceptance of the contract.*
- 4) *A positive acknowledgement indicates acceptance of the contract and contains no further information.*
- 5) *A negative acknowledgement contains an indication of the reason why the contract has not been accepted.*
- 6) *A non-compliance notification contains an indication of the events which the avionics cannot detect.*

Note 5.— Establishment and Operation of a Periodic Contract

a) Functional Description

- 1) *This function allows the ground system to establish a periodic contract with the aircraft, and then for the conditions of that contract to be realised. Realisation of the contract involves the sending of reports from the aircraft to the ground system at regular intervals (the reporting rate).*
- 2) *Only one periodic contract may exist between a ground system and the avionics at any one time. A set of basic information is sent with every report. Optionally, at the request of the ground system, other information blocks may also be sent; they may only be sent at a time interval which is a multiple of the reporting rate. The contract that is agreed includes the*

reporting rate, the optional blocks of information to be sent and the rate at which they are to be sent.

- 3) *The ground system sends a periodic contract request to the avionics. This contains the basic reporting rate and an indication of which optional information blocks are required and how often they are to be sent relative to the basic rate (i.e. every time, every second report, every third report etc.). The avionics then determines whether or not there are errors in the request, and if not, whether or not it is able to comply with the request. If the avionics can comply with the periodic contract request it sends its first report, and then sends other reports at the intervals requested. If it cannot send the first report within 0.5 seconds, it sends a positive acknowledgement first to indicate its acceptance of the contract.*
- 4) *Acceptance of a periodic contract request implicitly cancels any existing periodic contract.*
- 5) *If there are errors in the periodic contract request, or if the avionics cannot accept the contract, it sends a negative acknowledgement to the ground system indicating the reason for its inability to accept the contract within 0.5 seconds.*
- 6) *If the avionics can partially comply with the request, it sends a non-compliance notification indicating those parts of the contract with which it cannot comply. It then sends ADS-reports at a rate with which it can comply, and containing information requested with which it can comply. Non-compliance can be caused by either inability to meet the requested reporting rate, and/or inability to supply the requested information.*

b) *Message Descriptions*

- 1) *The periodic contract request may optionally contain any of the following information:*
 - i) *reporting interval;*
 - ii) *aircraft address modulus;*
 - iii) *projected profile modulus;*
 - iv) *ground vector modulus;*
 - v) *air vector modulus;*
 - vi) *weather modulus;*
 - vii) *short term intent modulus and projection time;*
 - viii) *extended projected profile modulus.*

- ix) *Moduli indicate the multiple of the reporting rate that the information block is sent at (e.g. weather modulus of 5 means that the weather information block is sent with every 5th report).*
- 2) *The ADS report has the same structure as in the operation of a demand contract.*
- 3) *An ADS report can contain a positive acknowledgement indicating acceptance of the contract.*
- 4) *A positive acknowledgement indicates acceptance of the contract and contains no further information.*
- 5) *A negative acknowledgement contains an indication of the reason why the contract has not been accepted.*
- 6) *A non-compliance notification contains an indication of which optional information fields cannot be sent, and/or indicates that the requested periodic report cannot be met.*

Note 6.— Cancellation of Contracts

a) Functional Description

- 1) *This function allows the ground system explicitly to cancel a contract that is in operation. The ground system sends a cancel contract message to the avionics. The avionics cancels the contract and acknowledges the cancellation.*
- 2) *Implicit cancellation occurs when a periodic contract is in place, and then the ground system establishes a new periodic contract - the first one is implicitly cancelled on the establishment of the second; similarly with event contracts. Demand contracts are implicitly cancelled when the report is sent. There are no additional information flows associated with implicit cancellation.*
- 3) *The ground system may also cancel all contracts in a single cancel all contracts message. The avionics cancels all contracts and acknowledges the cancellation.*

b) Message Descriptions

- 1) *The cancel contract message contains an indication of the contract to be cancelled.*
- 2) *The cancel all contracts message contains no additional information.*
- 3) *A positive acknowledgement contains no additional information.*

*Note 7.— Establishment and Operation of Emergency Contracts**a) Functional Description*

- 1) This function allows the avionics to initiate an emergency contracts (either on instruction from the pilot or on its own initiative), between the avionics and all ground systems with which it has existing contracts. Realisation of the contract involves the sending of ADS emergency reports from the avionics to the ground system at regular intervals.*
- 2) Any existing periodic contract is suspended pending the cancellation of the emergency contract. Initially, the emergency reporting rate is the lesser of 60 seconds or half any existing periodic contract rate (if one exists).*
- 3) The avionics sends ADS-emergency-reports to the ground system at the emergency reporting rate.*
- 4) The avionics sends ADS-emergency-reports to all ground systems with which it has event or periodic contracts.*

b) Message Descriptions

- 1) Each ADS-emergency-report always contains the following basic information:*
 - i) the 3-D position of the aircraft;*
 - ii) the time;*
 - iii) an indication of the accuracy of the positional information (figure of merit).*
- 2) With every fifth ADS-emergency-report, the following information is also included:*
 - i) the aircraft address;*
 - ii) the ground vector, indicating the track, ground speed and vertical rate;*

*Note 8.— Modifying an Emergency Contract**a) Functional Description*

- 1) This function allows the reporting rate of an emergency contract to be modified.*
- 2) The ground system sends an emergency contract modification message to the avionics. The avionics modifies the reporting rate of the emergency*

contract, and then sends the emergency reports at the new interval. This only effects the emergency contract between the ground system making the request and the aircraft.

- 3) *If the avionics is unable to change the reporting rate, the avionics will send a negative acknowledgement within 0.5 seconds.*

b) Message Descriptions

- 1) *The emergency contract modification message contains only a new reporting rate.*
- 2) *A negative acknowledgement will contain an indication that the reporting rate cannot be changed.*

Note 9.— Cancellation of Emergency Contracts

a) Functional Description

- 1) *This function allows the aircraft to cancel an emergency contract.*
- 2) *The avionics sends a cancel emergency contract message to the ground system and cancels the emergency contract. If there is an periodic contract in place when the emergency is cancelled, then it is reinstated. Emergency contract cancellation cancels all emergency contracts.*

b) Message Descriptions

- 1) *The cancel emergency contract message contains no information.*

2.2.1.2 General Requirements

2.2.1.2.1 ADS ASE Version Number

- 2.2.1.2.1.1 The ADS-air-ASE and the ADS-ground-ASE version numbers shall both be set to one.

2.2.1.2.2 Error Processing Requirements

- 2.2.1.2.2.1 In the event of information input by the ADS-user being incompatible with that able to be processed by the system, the user shall be notified.

- 2.2.1.2.2.2 In the event of an ADS-user invoking an ADS service primitive, when the ADS-ASE is not in a state specified in 2.2.1.5, the user shall be notified.

2.2.1.3 The Abstract Service

2.2.1.3.1 Service Description

2.2.1.3.1.1 An implementation of either the ADS ground based service or the ADS air based service shall exhibit behaviour consistent with having implemented an ADS-ground-ASE, or ADS-air-ASE respectively.

Note 1.— 2.2.1.3 defines the abstract service interface for the ADS service. The ADS-ASE abstract service is described in 2.2.1.3 from the viewpoint of the ADS-ASE-air-user, the ADS-ASE-ground-user and the ADS service-provider.

Note 2.— 2.2.1.3 defines the static behaviour (i.e. the format) of the ADS abstract service. Its dynamic behaviour (i.e. how it is used) is described in 2.2.1.7.

Note 3.— Figure 2.2.1.3-1 shows the functional model of the ADS Application. The functional modules identified in this model are the following :

- a) the ADS-user,
- b) the ADS Application Entity (ADS-AE) service interface,
- c) the ADS-AE,
- d) the ADS Control Function (ADS-CF),
- e) the ADS Application Service Element (ADS-ASE) service interface,
- f) the ADS-ASE, and
- g) the Dialogue Service (DS) interface.

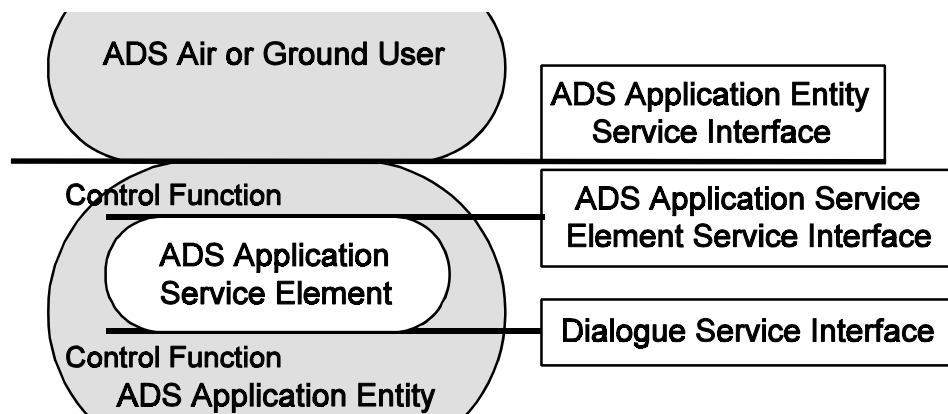


Figure 2.2.1.3-1 : Functional Model of the ADS Application

Note 4.— The ADS-user represents the operational part of the ADS system. This user does not perform the communication functions but relies on a communication service provided to it via the ADS-AE through the ADS-AE service interface. The individual actions at this interface are called ADS-AE service primitives. Similarly, individual actions at other interfaces in the communication system are called service primitives at these interfaces.

Note 5.— The ADS-AE consists of several elements, including the ADS-ASE and the ADS-CF. The DS interface is made available by the ADS-CF to the ADS-ASE for communication with the peer ADS-ASE.

Note 6.— The ADS-ASE is the element in the communication system which executes the ADS specific protocol. In other words, it takes care of the ADS specific service primitive sequencing actions, message creation, timer management, error and exception handling.

Note 7.— The ADS-ASE interfaces only with the ADS-CF. This ADS-CF is responsible for mapping service primitives received from one element (such as the ADS-ASE and the ADS-user) to other elements which interface with it. The part of the ADS-CF which is relevant from the point of view of these SARPs, i.e. the part between the ADS-user and the ADS-ASE, will map ADS-AE service primitives to ADS-ASE service primitives transparently.

Note 8.— The DS interface is the interface between the ADS-ASE and part of ADS-CF underneath, the ADS-ASE and provides the dialogue service.

2.2.1.3.2 The ADS-ASE Abstract Service

Note.— There is no requirement to implement the service in an ADS product; however, it is necessary to implement the ground based and air based system in such a way that it will be impossible to detect (from the peer system) whether or not an interface has been built.

2.2.1.3.2.1 The ADS-ASE abstract service shall consist of a set of the following services as allowed by the subsetting rules defined in 2.2.1.8:

- a) *ADS-demand-contract* service as defined in 2.2.1.3.4;
- b) *ADS-event-contract* service as defined in 2.2.1.3.5;
- c) *ADS-periodic-contract* service as defined in 2.2.1.3.6;
- d) *ADS-report* service as defined in 2.2.1.3.7;
- e) *ADS-cancel* service as defined in 2.2.1.3.8;
- f) *ADS-cancel-all-contracts* service as defined in 2.2.1.3.9;
- g) *ADS-emergency-report* service as defined in 2.2.1.3.10;
- h) *ADS-modify-emergency-contract* service as defined in 2.2.1.3.11;
- i) *ADS-cancel-emergency* service as defined in 2.2.1.3.12;

- j) *ADS-user-abort* service as defined in 2.2.1.3.13;
- k) *ADS-provider-abort* service as defined in 2.2.1.3.14.

Note 1.— ADS-demand-contract, ADS-event-contract, ADS-periodic-contract, ADS-cancel, ADS-cancel-all-contracts and ADS-modify-emergency-contract are only initiated by the ADS-ground-user.

Note 2.— ADS-report, ADS-emergency-report and ADS-cancel-emergency are only initiated by the ADS-air-user.

Note 3.— ADS-user-abort is initiated by an ADS-air-user or an ADS-ground-user.

Note 4.— ADS-provider-abort is only initiated by the ADS-service-provider.

Note 5.— An abstract syntax is a syntactical description of a parameter which does not imply a specific implementation. Only when the ADS-ASE maps a parameter onto an APDU field, or vice-versa, is the abstract syntax of the parameter described by using the ASN.1 of 2.2.1.4 for this field.

2.2.1.3.3 Conventions

Note 1.— For a given primitive, the presence of each parameter is described by one of the following values in the parameter tables 2.2.1.3:

- a) *blank* not present;
- b) *C* conditional upon some predicate explained in the text;
- c) *C(=)* conditional upon the value of the parameter to the immediate left being present, and equal to that value;
- d) *M* mandatory;
- e) *M(=)* mandatory, and equal to the value of the parameter to the immediate left;
- f) *U* user option.

Note 2.— The following abbreviations are used in this document:

- a) *Req* request; data is input by an ADS user initiating the service to its respective ASE;
- b) *Ind* indication; data is indicated by the receiving ASE to its respective ADS user;
- c) *Rsp* response; data is input by receiving ADS user to its respective ASE;
- d) *Cnf* confirmation; data is confirmed by the initiating ASE to its respective ADS user.

Note 3.— An unconfirmed service allows just one message to be transmitted, in one direction.

Note 4.— A confirmed service provides end-to-end confirmation that a message sent by one user was received by its peer user.

2.2.1.3.4 ADS-demand-contract Service

Note.— The ADS-demand-contract service allows the ADS-ground-user to request a demand contract with the aircraft. It is a confirmed service, initiated by the ADS-ground-user.

2.2.1.3.4.1 The ADS-demand-contract service shall contain primitives and parameters as presented in Table 2.2.1.3-1, when an acknowledgement is sent independent of an ADS-report.

Table 2.2.1.3-1: ADS-demand-contract service parameters

Parameter Name	Req	Ind	Rsp	Cnf
Aircraft address	M			
Class of communication service	U			
Contract details	M	M(=)		
Reply			M	M(=)
ICAO facility designation	C	C(=)		

2.2.1.3.4.2 The ADS-demand-contract service primitives shall contain the parameters as presented in Table 2.2.1 3-2, when a positive acknowledgement is embedded in an ADS-report.

Table 2.2.1.3-2: ADS-demand-contract service parameters

Parameter Name	Req	Ind
Aircraft address	M	
Class of communication service	U	
Contract details	M	M(=)
ICAO facility designation	C	C(=)

2.2.1.3.4.3 Aircraft Address

Note.— This parameter contains the 24 bit ICAO address of the aircraft with which the contract is being made.

2.2.1.3.4.3.1 The *aircraft address* parameter value shall conform to an abstract value corresponding to a 24-bit ICAO aircraft address.

2.2.1.3.4.4 Class of Communication Service

Note.— This parameter contains the value of the required class of communication service, if specified by the ADS-ground-user.

2.2.1.3.4.4.1 Where specified by the ADS-ground-user, the *class of communication service* parameter shall have one of the following abstract values: “A”, “B”, “C”, “D”, “E”, “F”, “G” or “H”.

Note 1.— If contracts are currently in place, the class of communication service parameter is not used by the ADS-service provider.

Note 2.— Where not specified by the ADS-ground-user, when there are no contracts already in force, this indicates that there will be no routing preference.

2.2.1.3.4.5 Contract Details

Note.— This parameter contains the details of the contract as requested by the ADS-ground-user.

2.2.1.3.4.5.1 The *contract details* parameter value shall conform to the ASN.1 abstract syntax *DemandContract*.

2.2.1.3.4.6 Reply

Note 1.— This parameter indicates the extent to which the contract request can be complied with. If it has the value NegativeAcknowledgement, it indicates that the contract has been rejected and gives reasons. If it has the value NoncomplianceNotification, it indicates that only some parts of the contract can be complied with, and indicates which ones have been rejected.

Note 2.— Unlike Event and Periodic contracts, the “Reply” parameter does not have the option of containing a positive acknowledgement.

2.2.1.3.4.6.1 The *reply* parameter value shall conform to one of the following abstract syntaxes:

- a) The abstract value “Negative Acknowledgement”, and a value conforming to the ASN.1 abstract syntax *Reason*; or
- b) The abstract value “Noncompliance Notification”, and a value conforming to the ASN.1 abstract syntax *NoncomplianceNotification* with the choice *demand-ncn*.

2.2.1.3.4.7 ICAO facility designation

Note.— This parameter contains the 4 to 8 character ICAO facility designation of the ICAO facility which is initiating the contract. If contracts are currently in place, this parameter is not used by the ADS-service-provider.

2.2.1.3.4.7.1 The *ICAO facility designation* parameter value shall conform to an abstract value corresponding to a 4 to 8 character ICAO facility designation.

2.2.1.3.4.7.2 The ICAO *facility designation* parameter value shall be provided when the ADS-ground-user has no other contracts in place with the aircraft.

2.2.1.3.5 ADS-event-contract Service

Note.— The ADS-event-contract service allows the ADS-ground-user to request an event contract with the aircraft. It is a confirmed service, initiated by the ADS-ground-user.

2.2.1.3.5.1 The ADS-event-contract service shall contain primitives and parameters as presented in Table 2.2.1.3-3, when an acknowledgement is sent independent of an ADS-report.

Table 2.2.1.3-3: ADS-event-contract service parameters

Parameter Name	Req	Ind	Rsp	Cnf
Aircraft address	M			
Class of communication service	U			
Contract details	M	M(=)		
Reply			M	M(=)
ICAO facility designation	C	C(=)		

2.2.1.3.5.2 The ADS-event-contract service shall contain primitives and parameters as presented in Table 2.2.1.3-4, when a positive acknowledgement is embedded in an ADS-report.

Table 2.2.1.3-4: ADS-event-contract service parameters

Parameter Name	Req	Ind
Aircraft address	M	
Class of communication service	U	
Contract details	M	M(=)
ICAO facility designation	C	C(=)

2.2.1.3.5.3 Aircraft Address

Note.— This parameter contains the 24 bit ICAO address of the aircraft with which the contract is being made.

2.2.1.3.5.3.1 The *aircraft address* parameter value shall conform to an abstract value corresponding to a 24-bit ICAO aircraft address.

2.2.1.3.5.4 Class of Communication Service

Note.— This parameter contains the value of the required class of communication service, if specified by the ADS-ground-user.

2.2.1.3.5.4.1 Where specified by the ADS-ground-user, the *class of communication service* parameter shall have one of the following abstract values: “A”, “B”, “C”, “D”, “E”, “F”, “G” or “H”.

Note 1.— If contracts are currently in place, the class of communication service parameter is not used by the ADS-service provider.

Note 2.— Where not specified by the ADS-ground-user, when there are no contracts already in force, this indicates that there will be no routing preference.

2.2.1.3.5.5 Contract Details

Note.— This parameter contains the details of the contract as requested by the ADS-ground-user.

2.2.1.3.5.5.1 The *contract details* parameter value shall conform to the ASN.1 abstract syntax *EventContract*.

2.2.1.3.5.6 Reply

Note.— This parameter indicates the extent to which the contract request can be complied with. If it has the value NegativeAcknowledgement, it indicates that the contract has been rejected and gives reasons. If it has the value NoncomplianceNotification, it indicates that only some parts of the contract can be complied with, and indicating which ones have been rejected. If it has the value PositiveAcknowledgement, it indicates full compliance with the contract has been accepted.

2.2.1.3.5.6.1 The *reply* parameter value shall conform to one of the following abstract syntaxes:

- a) The abstract value “Negative Acknowledgement”, and a value conforming to the ASN.1 abstract syntax *Reason*;
- b) The abstract value “Noncompliance Notification”, and a value conforming to the ASN.1 abstract syntax *NoncomplianceNotification* with the choice *event-ncn*;
- c) The abstract value “Positive Acknowledgement”, and a value conforming to the ASN.1 abstract syntax *NULL*.

2.2.1.3.5.7 ICAO facility designation

Note.— This parameter contains the 4 to 8 character ICAO facility designation of the ICAO facility which is initiating the contract. If contracts are currently in place, this parameter is not used by the ADS-service-provider.

2.2.1.3.5.7.1 The *ICAO facility designation* parameter value shall conform to an abstract value corresponding to a 4 to 8 character ICAO facility designation.

2.2.1.3.5.7.2 The ICAO *facility designation* parameter value shall be provided when the ADS-ground-user has no other contracts in place with the aircraft.

2.2.1.3.6 ADS-periodic-contract Service

Note.— The ADS-periodic-contract service allows the ADS-ground-user to request a periodic contract with the aircraft. It is a confirmed service, initiated by the ADS-ground-user.

2.2.1.3.6.1 The ADS-periodic-contract service shall contain primitives and parameters as presented in Table 2.2.1.3-5, when an acknowledgement is sent independent of an ADS-report.

Table 2.2.1.3-5: ADS-periodic-contract service parameters

Parameter Name	Req	Ind	Rsp	Cnf
Aircraft address	M			
Class of communication service	U			
Contract details	M	M(=)		
Reply			M	M(=)
ICAO facility designation	C	C(=)		

2.2.1.3.6.2 The ADS-periodic-contract service shall contain primitives and parameters as presented in Table 2.2.1.3-6, when a positive acknowledgement is embedded in an ADS-report.

Table 2.2.1.3-6: ADS-periodic-contract service parameters

Parameter Name	Req	Ind
Aircraft address	M	
Class of communication service	U	
Contract details	M	M(=)
ICAO facility designation	C	C(=)

2.2.1.3.6.3 Aircraft Address

Note.— This parameter contains the 24 bit ICAO address of the aircraft with which the contract is being made.

2.2.1.3.6.3.1 The *aircraft address* parameter value shall conform to an abstract value corresponding to a 24-bit ICAO aircraft address.

2.2.1.3.6.4 Class of Communication Service

Note.— This parameter contains the value of the required class of communication service, if specified by the ADS-ground-user.

2.2.1.3.6.4.1 Where specified by the ADS-ground-user, the *class of communication service* parameter shall have one of the following abstract values: “A”, “B”, “C”, “D”, “E”, “F”, “G” or “H”.

Note 1.— If contracts are currently in place, the class of communication service parameter is not used by the ADS-service provider.

Note 2.— Where not specified by the ADS-ground-user, when there are no contracts already in force, this indicates that there will be no routing preference.

2.2.1.3.6.5 Contract Details

Note.— This parameter contains the details of the contract as requested by the ADS-ground-user.

2.2.1.3.6.5.1 The *contract details* parameter value shall conform to the ASN.1 abstract syntax *PeriodicContract*.

2.2.1.3.6.6 Reply

Note.— This parameter indicates the extent to which the contract request can be complied with. If it has the value Negative Acknowledgement, it indicates that the contract has been rejected and gives reasons. If it has the value NoncomplianceNotification, it indicates that only some parts of the contract can be complied with, and indicating what part of the contract cannot be conformed to. If it has the value PositiveAcknowledgement, it indicates that the contract has been accepted but cannot be satisfied immediately, either because there is an emergency contract in place, or because the information is not available within the 0.5 second turnaround time.

2.2.1.3.6.6.1 The *reply* parameter value shall conform to one of the following abstract syntaxes:

- a) The abstract value “Negative Acknowledgement”, and a value conforming to the ASN.1 abstract syntax *Reason*;
- b) The abstract value “Noncompliance Notification”, and a value conforming to the ASN.1 abstract syntax *NoncomplianceNotification* with the choice *periodic-ncn*;
- c) The abstract value “Positive Acknowledgement”, and a value conforming to the ASN.1 abstract syntax *NULL*.

2.2.1.3.6.7 ICAO facility designation

Note.— This parameter contains the 4 to 8 character ICAO facility designation of the ICAO facility which is initiating the contract. If contracts are currently in place, this parameter is not used by the ADS-service-provider.

2.2.1.3.6.7.1 The *ICAO facility designation* parameter value shall conform to an abstract value corresponding to a 4 to 8 character ICAO facility designation.

2.2.1.3.6.7.2 The *ICAO facility designation* parameter value shall be provided when the ADS-ground-user has no other contracts in place with the aircraft.

2.2.1.3.7 ADS-report Service

Note.— The ADS-report service allows the ADS-air-user to send an ADS report to the ADS-ground-user. This is an unconfirmed service, initiated by the ADS-air-user.

2.2.1.3.7.1 The ADS-report service shall contain primitives and parameters as contained in Table 2.2.1.3-7.

Table 2.2.1.3-7: ADS-report service parameters

Parameter Name	Req	Ind
Contract type	M	M(=)
Event Type	C	C(=)
Positive Acknowledgement	U	C(=)
Report details	M	M(=)

2.2.1.3.7.2 Contract Type

Note.— This parameter identifies the type of contract that this report is in response to.

2.2.1.3.7.2.1 The *contract type* parameter value shall contain one of the abstract values “demand contract”, “event contract”, or “periodic contract”.

2.2.1.3.7.3 Event Type

Note.— This parameter indicates the type of event that triggered the report.

2.2.1.3.7.3.1 The *event type* parameter shall be present if, and only if, the *contract type* parameter has the abstract value *event-contract*.

2.2.1.3.7.3.2 The *event type* parameter shall contain a value conforming to the abstract syntax *EventTypeReported*.

2.2.1.3.7.4 Positive Acknowledgement

Note.— This parameter is used to indicate that the report carries with it a positive acknowledgement of a new event contract, or a new periodic contract, or a new demand contract.

2.2.1.3.7.4.1 The *positive acknowledgement* parameter abstract syntax shall be *NULL*.

2.2.1.3.7.5 Report Details

Note.— This parameter contains the details of the ADS report.

2.2.1.3.7.5.1 The *report details* parameter value shall conform to the ASN.1 abstract syntax *ADSReport*.

2.2.1.3.8 ADS-cancel Service

Note.— The ADS-cancel service allows the ADS-ground-user to cancel an existing contract. It is a confirmed service, initiated by the ADS-ground-user.

2.2.1.3.8.1 The ADS-cancel service shall contain primitives and parameters as contained in Table 2.2.1.3-8.

Table 2.2.1.3-8: ADS-cancel service parameters

Parameter Name	Req	Ind	Cnf
Contract type	M	M(=)	M(=)

2.2.1.3.8.2 Contract Type

Note.— This parameter identifies the type of contract that is to be cancelled.

2.2.1.3.8.2.1 The *contract type* parameter value shall conform to the ASN.1 abstract syntax *CancelContract*.

2.2.1.3.9 ADS-cancel-all-contracts Service

Note.— The ADS-cancel-all-contracts service allows the ADS-ground-user to cancel all contracts with a particular aircraft. It is a confirmed service, initiated by the ADS-ground-user.

2.2.1.3.9.1 The ADS-cancel-all-contracts service shall contain primitives as contained in Table 2.2.1.3-9.

Table 2.2.1.3-9: ADS-cancel-all-contracts service parameters

Parameter Name	Req	Ind	Cnf
none			

2.2.1.3.10 ADS-emergency-report Service

Note.— The ADS-emergency-report service allows the ADS-air-user to send an emergency ADS report to the ADS-ground-user. This is an unconfirmed service, initiated by the ADS-air-user.

2.2.1.3.10.1 The ADS-emergency-report service shall contain primitives and parameters as contained in Table 2.2.1.3-10.

Table 2.2.1.3-10: ADS-emergency-report service parameters

Parameter Name	Req	Ind
Positive acknowledgement of modification	U	C(=)
Emergency report details	M	M(=)

2.2.1.3.10.2 Positive Acknowledgement of Modification

Note.— The presence of this parameter indicates that the ADS-air-user has received the ADS-modify-emergency-contract indication and has accepted it.

2.2.1.3.10.2.1 The *positive acknowledgement of modification* parameter abstract syntax shall be *NULL*.

2.2.1.3.10.3 Emergency report details

Note.— The parameter contains the details of the emergency report.

2.2.1.3.10.3.1 The *emergency report details* parameter value shall conform to the ASN.1 abstract syntax *ADSEmergencyReport*.

2.2.1.3.11 ADS-modify-emergency-contract Service

Note.— The ADS-modify-emergency-contract service allows the ADS-ground-user to request changes to an emergency contract reporting rate. It is a confirmed service, initiated by the ADS-ground-user.

2.2.1.3.11.1 The ADS-modify-emergency-contract service shall contain primitives and parameters as contained in Table 2.2.1.3-11, when the request to modify the emergency rate is refused.

Table 2.2.1.3-11: ADS-modify-emergency-contract parameters

Parameter Name	Req	Ind	Rsp	Cnf
Reporting interval	M	M(=)		

2.2.1.3.11.2 The ADS-modify-emergency-contract service shall contain primitives and parameters as presented in Table 2.2.1.3-12, when the request to modify the emergency rate is accepted, and a positive acknowledgement is embedded in an ADS-emergency-report.

Table 2.2.1.3-12: ADS-modify-emergency-contract parameters

Parameter Name	Req	Ind
Reporting interval	M	M(=)

2.2.1.3.11.3 Reporting Interval

Note.— This parameter indicates the new interval for sending the ADS emergency reports.

2.2.1.3.11.3.1 The *reporting interval* parameter value shall conform to the ASN.1 abstract syntax *ReportingInterval*.

2.2.1.3.12 ADS-cancel-emergency Service

Note.— The ADS-cancel-emergency service allows the ADS-air-user to inform the ADS-ground-user that the emergency contract has been cancelled. When the emergency is concluded, the ADS-air-user must invoke this service with every ground system with which it has an emergency contract. This is an unconfirmed service, initiated by the ADS-air-user.

2.2.1.3.12.1 The ADS-cancel-emergency service shall contain primitives as contained in Table 2.2.1.3-13.

Table 2.2.1.3-13: ADS-cancel-emergency service parameters

Parameter Name	Req	Ind
none		

2.2.1.3.13 ADS-user-abort Service

Note 1.— The ADS-user-abort service allows the ADS-air-user to abort all ADS contracts with a particular ground system or ADS-ground-user to abort all ADS contracts with a particular aircraft. It is an unconfirmed service, initiated by an ADS-ground-user or the ADS-air-user. Messages in transit may be lost during this operation. It can be invoked at any time that the ADS-user is aware that any ADS service is in operation.

Note 2.— If the service is invoked prior to complete establishment of the dialogue, the ADS-user-abort indication may not be provided. An ADS-provider-abort indication may result instead.

2.2.1.3.13.1 The ADS-user-abort service shall contain primitives as contained in Table 2.2.1.3-14.

Table 2.2.1.3-14: ADS-user-abort service parameters

Parameter Name	Req	Ind
none		

2.2.1.3.14 ADS-provider-abort Service

Note.— The ADS-provider-abort service allows the ADS-service-provider to inform the ADS-ground-user and the ADS-air-user that it can no longer provide the ADS service for a particular ADS-ground-user - ADS-air-user pairing. It is initiated by the ADS-service-provider. Messages in transit may be lost during this operation.

2.2.1.3.14.1 The ADS-provider-abort service shall contain primitives and parameters as contained in Table 2.2.1.3-15.

Table 2.2.1.3-15: ADS-provider-abort service parameters

Parameter Name	Ind
Reason	M

2.2.1.3.14.2 Reason

Note.— This parameter identifies the reason for the abort.

2.2.1.3.14.2.1 The *reason* parameter shall conform to the ASN.1 abstract syntax *AbortReason*.

2.2.1.4 Formal Definitions of Messages**2.2.1.4.1 Encoding/Decoding Rules**

2.2.1.4.1.1 An ADS-air-ASE shall be capable of encoding [ADSAircraftPDUs] APDUs and decoding [ADSGroundPDUs] APDUs.

2.2.1.4.1.2 An ADS-ground-ASE shall be capable of encoding [ADSGroundPDUs] APDUs and decoding [ADSAircraftPDUs] APDUs.

2.2.1.4.2 ADS ASN.1 Abstract Syntax

2.2.1.4.2.1 The abstract syntax of the air-ground ADS protocol data units shall comply with the description contained in the ASN.1 module ADSMessageSetVersion1 (conforming to ISO/IEC 8824), as defined in 2.2.1.4.

Note .— Where units indicate directional information, the value is given relative to true North. If magnetic information is required this will be a matter for local ground implementation.

ADSMessagesetVersion1 DEFINITIONS::=

BEGIN

EXPORTS

AbortReason, ADSEmergencyReport, ADSReport, AircraftAddress, EventTypeReported;

-- Aircraft-generated and Ground-generated Message Choice

ADSAircraftPDUs ::= CHOICE

{	aDS-cancel-emergency-PDU	[0]	NULL,
	aDS-demand-report-PDU	[1]	ADSDemandReport,
	aDS-emergency-report-PDU	[2]	ADSEmergency,
	aDS-event-report-PDU	[3]	ADSEventReport,
	aDS-negative-acknowledgement-PDU	[4]	NegativeAcknowledgement,
	aDS-noncompliance-notification-PDU	[5]	NoncomplianceNotification,
	aDS-periodic-report-PDU	[6]	ADSPeriodicReport,
	aDS-positive-acknowledgement-PDU	[7]	PositiveAcknowledgement,
	aDS-provider-abort-PDU	[8]	AbortReason,
	...		
}			

ADSGroundPDUs ::= CHOICE

{	aDS-cancel-all-contracts-PDU	[0]	NULL,
	aDS-cancel-contract-PDU	[1]	CancelContract,
	aDS-cancel-emergency-acknowledgement-PDU	[2]	NULL,
	aDS-demand-contract-PDU	[3]	DemandContract,
	aDS-event-contract-PDU	[4]	EventContract,
	aDS-modify-emergency-contract-PDU	[5]	ModifyEmergency,
	aDS-periodic-contract-PDU	[6]	PeriodicContract,
	aDS-provider-abort-PDU	[7]	AbortReason,
	...		
}			

-- Ground-generated and Aircraft-generated message components - Protocol Data Units

AbortReason ::= ENUMERATED

{	communications-service-failure	(0),
	unrecoverable-system-error	(1),
	invalid-PDU	(2),
	sequence-error	(3),
	timer-expiry	(4),
}		

cannot-establish-contact	(5),		
undefined-error	(6),		
dialogue-end-not-accepted	(7),		
unexpected-PDU	(8),		
decoding-error	(9),		
invalid-qos-parameter	(10),		
...			

}

ADSDemandReport ::= SEQUENCE

{			
report	[0]	ADSReport,	
positive-acknowledgement	[1]	NULL	OPTIONAL,
...			
}			

ADSEmergency ::= SEQUENCE

{			
emergency-report	[0]	ADSEmergencyReport,	
positive-acknowledgement	[1]	NULL	OPTIONAL,
...			
}			

ADSEventReport ::= SEQUENCE

{			
event-type	[0]	EventTypeReported,	
report	[1]	ADSReport,	
positive-acknowledgement	[2]	NULL	OPTIONAL,
...			
}			

ADSPeriodicReport ::= SEQUENCE

{			
report	[0]	ADSReport,	
positive-acknowledgement	[1]	NULL	OPTIONAL,
...			
}			

CancelContract ::= ENUMERATED

{			
event-contract	(0),		
periodic-contract	(1),		
...			
}			

DemandContract ::= SEQUENCE

{			
aircraft-address	[0]	NULL	OPTIONAL,
projected-profile	[1]	NULL	OPTIONAL,

ground-vector	[2]	NULL	OPTIONAL,
air-vector	[3]	NULL	OPTIONAL,
weather	[4]	NULL	OPTIONAL,
short-term-intent	[5]	ProjectionTime	OPTIONAL,
extended-projected-profile	[6]	ExtendedProjectedProfileRequest	OPTIONAL,
...			
}			
EventContract ::= SEQUENCE			
{			
lateral-deviation-change	[0]	LateralChange	OPTIONAL,
vertical-rate-change	[1]	VerticalRateChange	OPTIONAL,
level-range	[2]	LevelRange	OPTIONAL,
way-point-change	[3]	NULL	OPTIONAL,
air-speed-change	[4]	AirSpeedChange	OPTIONAL,
ground-speed-change	[5]	GroundSpeedChange	OPTIONAL,
heading-change	[6]	DegreesDirection	OPTIONAL,
extended-projected-profile-change	[7]	ExtendedProjectedProfileRequest	OPTIONAL,
fom-change	[8]	NULL	OPTIONAL,
track-angle-change	[9]	DegreesDirection	OPTIONAL,
level-change	[10]	LevelChange	OPTIONAL,
...			
}			
ModifyEmergency ::= ReportingInterval			
NegativeAcknowledgement ::= SEQUENCE			
{			
request-type		RequestType,	
reason		Reason	
}			
NoncomplianceNotification ::= CHOICE			
{			
demand-ncn	[0]	SET OF ReportType,	
event-ncn	[1]	SET OF EventTypeContracted,	
periodic-ncn	[2]	SET OF ReportTypeAndPeriod,	
...			
}			
PeriodicContract ::= SEQUENCE			
{			
reporting-interval	[0]	ReportingInterval	DEFAULT minutes-scale: 5,
aircraft-address-modulus	[1]	Modulus	OPTIONAL,
projected-profile-modulus	[2]	Modulus	OPTIONAL,

ground-vector-modulus	[3]	Modulus	OPTIONAL,
air-vector-modulus	[4]	Modulus	OPTIONAL,
weather-modulus	[5]	Modulus	OPTIONAL,
short-term-intent-modulus	[6]	ShortTermIntentModulus	OPTIONAL,
extended-projected-profile-modulus	[7]	ExtendedProjectedProfileModulus	OPTIONAL,
...			

}

PositiveAcknowledgement ::= RequestType

 -- Reports and their components

ADSEmergencyReport ::= SEQUENCE

{			
position	[0]	Position,	
time-stamp	[1]	DateTimeGroup,	
fom	[2]	FigureOfMerit,	
aircraftAddress	[3]	AircraftAddress	OPTIONAL,
ground-vector	[4]	GroundVector	OPTIONAL
}			

ADSReport ::= SEQUENCE

{			
position	[0]	Position,	
time-stamp	[1]	DateTimeGroup,	
fom	[2]	FigureOfMerit,	
aircraft-address	[3]	AircraftAddress	OPTIONAL,
projected-profile	[4]	ProjectedProfile	OPTIONAL,
ground-vector	[5]	GroundVector	OPTIONAL,
air-vector	[6]	AirVector	OPTIONAL,
weather	[7]	Weather	OPTIONAL,
short-term-intent	[8]	ShortTermIntent	OPTIONAL,
extended-projected-profile	[9]	ExtendedProjectedProfile	OPTIONAL,
...			
}			

AircraftAddress ::= BIT STRING (SIZE (24))

-- 24 bit ICAO airframe identifier

AirVector ::= SEQUENCE

{			
heading	[0]	DegreesDirection	OPTIONAL,
air-speed	[1]	AirSpeed	OPTIONAL,
vertical-rate	[2]	VerticalRateChange	OPTIONAL
}			

AirSpeed ::= CHOICE

```
{
    mach                [0]    Mach,
    ias                 [1]    Ias,
    mach-and-ias        [2]    SEQUENCE
                           {
                               mach    Mach,
                               ias     Ias
                           }
}
```

-- When AirSpeed is returned in an ADS report, the choice of which of the above units of
 -- air speed are used depends on how the aircraft is equipped and whether the aircraft is
 -- flying on Mach or IAS at the time. The choice is made by the avionics.

ExtendedProjectedProfile ::= SEQUENCE SIZE (1..128) OF SEQUENCE

```
{
    way-point            Position,
    time                 Eta
}
```

FigureOfMerit ::= SEQUENCE

```
{
    position-accuracy    PositionAccuracy,
    multiple-navigational-units-operating BOOLEAN,
    acas-operational     BOOLEAN
}
```

PositionAccuracy ::= ENUMERATED -- nm = nautical miles

```
{
    complete-loss    (0),
    under30nm        (1),
    under15nm        (2),
    under8nm         (3),
    under4nm         (4),
    under1nm         (5),
    under-25nm       (6), -- under 0.25 nm
    under-05nm       (7) -- under 0.05 nm
}
```

GroundVector ::= SEQUENCE

```
{
    track                [0]    DegreesDirection    OPTIONAL,
    ground-speed         [1]    INTEGER (-50..2200)    OPTIONAL,
                                -- units =knots
                                -- range =-50 to +2200 knots
    vertical-rate        [2]    VerticalRateChange    OPTIONAL
}
```

ProjectedProfile ::= SEQUENCE

```
{
    next-way-point          Position,
    next-time               Eta,
    following-way-point     Position
}
```

ShortTermIntent ::= SEQUENCE

```
{
    position                Position,
    projected-time          ProjectionTime,
    intermediate-intent     IntermediateIntent
}
```

IntermediateIntent ::= SEQUENCE SIZE (0..7) OF SEQUENCE

```
{
    distance                INTEGER (1..8000),
                           -- units = Nautical miles
                           -- range = 1 to 8000 Nautical miles
    track                  DegreesDirection,
    level                  Level,
    projected-time          ProjectionTime
}
```

-- IntermediateIntent indicates a set of way points between the current position and the
 -- time indicated in the ShortTerm intent element.
 -- distance is expressed as relative to position at the time of the ADS-report.
 -- track is expressed as absolute track.
 -- level is expressed as absolute level.
 -- projected-time is expressed as relative to the timestamp on the ADS-report.

Weather ::= SEQUENCE

```
{
    wind-speed             [0]  INTEGER (0..300) OPTIONAL,
                           -- units = knots
                           -- range = 0 to 300 knots
    wind-direction         [1]  INTEGER (1..360) OPTIONAL,
                           -- units = degrees true North
                           -- range= 1 to 360 degrees
    temperature            [2]  INTEGER (-400..400) OPTIONAL,
                           -- units = 0.25 degrees Celsius
                           -- range = -100 to 100 degrees C
    turbulence              [3]  INTEGER (0..15) OPTIONAL
                           -- this is a place marker for a turbulence
                           -- index which is to be defined
}
```

-- Components of Contracts

AirSpeedChange ::= CHOICE

```
{
    mach-number-change      [0]    INTEGER (1..255),
                                -- units = 0.005 Mach
                                -- range = 0.005 to 1.275 Mach
    ias-change              [1]    INTEGER (1..700)
                                -- units = knots
                                -- range = 1 to 700 knots
}
```

LevelChange ::= INTEGER (1..500)

-- units = 10 feet
 -- range = 10 to 5 000 feet

LevelRange ::= SEQUENCE

```
{
    ceiling                Level,
    floor                  Level
}
```

DegreesDirection ::= INTEGER (1..3600)

-- units = 0.1 degrees true North,
 -- range = 0.1 to 360 degrees

ExtendedProjectedProfileModulus ::= SEQUENCE

```
{
    modulus                Modulus,
    extended-projected-profile-request ExtendedProjectedProfileRequest
}
```

ExtendedProjectedProfileRequest ::= CHOICE

```
{
    time-interval          [0]    INTEGER (1..80),
                                -- relative to current time stamp
                                -- units = 15 minutes
                                -- range = 15 minutes to 20 hours
    number-of-way-points   [1]    INTEGER (1..128)
}
```

GroundSpeedChange ::= INTEGER (0..300)

-- units =Knots
 -- range = 0 to 300 knots

Ias ::= INTEGER(0..1100)

-- units =knots
 -- range =0 to 1100 knots

LateralChange ::= INTEGER (0..2000)

-- units = 0.1 Nautical miles

```
-- range= 0 to 200 Nautical miles

Mach ::= INTEGER (500..4000)
    -- units = Mach 0.001
    -- range =0.5 Mach to 4 Mach

ShortTermIntentModulus ::= SEQUENCE
{
    intent-modulus                Modulus,
    intent-projection-time        ProjectionTime
}

ProjectionTime ::= INTEGER (1..240)
    -- units = minutes relative to current time stamp
    -- range = 1 minute to 4 hours

ReportingInterval ::= CHOICE
{
    seconds-scale                 [0]    INTEGER (1..59),
                                   -- units = seconds
                                   -- range = 1 second to 59 seconds
    minutes-scale                 [1]    INTEGER (1..120)
                                   -- units = minutes
                                   -- range = 1 minute to 2 hours
}

VerticalRateChange ::= INTEGER (-3000..3000)
    -- units = 10 feet per minute
    -- range =-30 000 to +30 000 feet per minute

-----
-- Miscellaneous components
-----

Reason ::= CHOICE
{
    aDS-service-unavailable       [0]    NULL,
    undefined                     [1]    NULL,
    -- the undefined value should not be used
    maximum-capacity-exceeded     [2]    GroundSystemsUsingService,
    undefined-reason               [3]    NULL,
    ...
}

GroundSystemsUsingService ::= SEQUENCE OF IA5String (SIZE(4..8))
-- contains a sequence of ICAO facility designations

RequestType ::= ENUMERATED
{
```

```
    event-contract                (0),
    periodic-contract              (1),
    demand-contract                (2),
    cancel-event-contract          (3),
    cancel-periodic-contract       (4),
    modify-emergency-contract      (5),
    cancel-all-contracts          (6),
    ...
}
```

EventTypeContracted ::= ENUMERATED

```
{
    lateral-deviation-change       (0),
    vertical-rate-change           (1),
    level-threshold                 (2),
    way-point-change               (3),
    air-speed-change               (4),
    ground-speed-change            (5),
    heading-change                 (6),
    extended-projected-profile-change (7),
    fom-change                     (8),
    track-angle-change             (9),
    level-change                   (10),
    ...
}
```

EventTypeReported ::= ENUMERATED

```
{
    lateral-deviation-change       (0),
    vertical-rate-change           (1),
    level-threshold                 (2),
    way-point-change               (3),
    air-speed-change               (4),
    ground-speed-change            (5),
    heading-change                 (6),
    extended-projected-profile-change (7),
    fom-change                     (8),
    track-angle-change             (9),
    level-change                   (10),
    baseline                       (11),
    ability-to-detect-events-impaired (12),
    ...
}
```

Modulus ::= INTEGER (1..255)

ReportType ::= ENUMERATED

```
{
    aircraft-address          (0),
    projected-profile         (1),
    ground-vector             (2),
    air-vector                (3),
    weather                   (4),
    short-term-intent         (5),
    extended-projected-profile (6),
    ...
}
```

ReportTypeAndPeriod ::= ENUMERATED

```
{
    aircraft-address          (0),
    projected-profile         (1),
    ground-vector             (2),
    air-vector                (3),
    weather                   (4),
    short-term-intent         (5),
    extended-projected-profile (6),
    reporting-rate            (7),
    ...
}
```

 -- Common components

Eta ::= Time

Position ::= SEQUENCE

```
{
    latitude      Latitude,
    longitude     Longitude,
    level         Level
}
```

Latitude ::= SEQUENCE

```
{
    sign          Sign,
    degrees       INTEGER (0..90),
                  -- units = degrees
                  -- range = 0 degrees to 90 degrees
    minutes       INTEGER (0..59),
                  -- units = minutes
                  -- range = 0 minutes to 59 minutes
    tenth-seconds INTEGER (0..599)
                  -- units = 0.1 seconds
                  -- range = 0 seconds to 59.9 seconds
}
```

Longitude ::= SEQUENCE

```
{
    sign          Sign,
    degrees       INTEGER (0..180),
                  -- units = degrees
                  -- range = 0 degrees to 180 degrees
    minutes       INTEGER (0..59),
                  -- units = minutes
                  -- range = 0 minutes to 59 minutes
    tenth-seconds INTEGER (0..599)
                  -- units = 0.1 seconds
                  -- range = 0 seconds to 59.9 seconds
}
```

Sign ::= ENUMERATED

```
{
    plus          (0),
    minus         (1)
}
```

Level ::= INTEGER(-75..10000)
 -- units = 10 feet
 — range = -750 to 100 000 feet

DateTimeGroup ::= SEQUENCE

```
{
    date      Date,
    time      Time
}
```

Date ::= SEQUENCE

```
{
    year      Year,
    month     Month,
    day       Day
}
```

Year ::= INTEGER (1996..2095)

 -- unit = year
 -- range = 1996 to 2095

Month ::= INTEGER (1..12)

 -- unit = month
 -- range = January to December

Day ::= INTEGER (1..31)

 -- unit = day
 -- range = 1 to 31


```
Time ::= SEQUENCE
{
    timeHours          [0]  TimeHours,
    timeMinutes        [1]  TimeMinutes,
    timeSeconds        [2]  TimeSeconds    OPTIONAL
}

TimeHours ::= INTEGER (0..23)
    -- units = hours
    -- range = midnight to 23.00 (11 PM)

TimeMinutes ::= INTEGER (0..59)
    -- units = minutes
    -- range = 0 minutes to 59 minutes

TimeSeconds ::= INTEGER (0..59)
    -- units = seconds
    -- range = 0 seconds to 59 seconds

END    -- of ADSMessageSetVersion1
```

2.2.1.5 Protocol Definition

2.2.1.5.1 Sequence Rules

2.2.1.5.1.1 Only the sequence of primitives illustrated in figures 2.2.1.5-1 to 2.2.1.5-35 shall be permitted.

Note 1.— The following figures define the valid sequences of primitives that are possible to be invoked during the operation of the ADS application. They show the relationship in time between the service request and the resulting indication, and if applicable, the subsequent response and the resulting confirmation.

Note 2.— Abort primitive may interrupt and terminate any of the normal message sequences outlined below.

Note 3.— Primitives are processed in the order in which they are received .

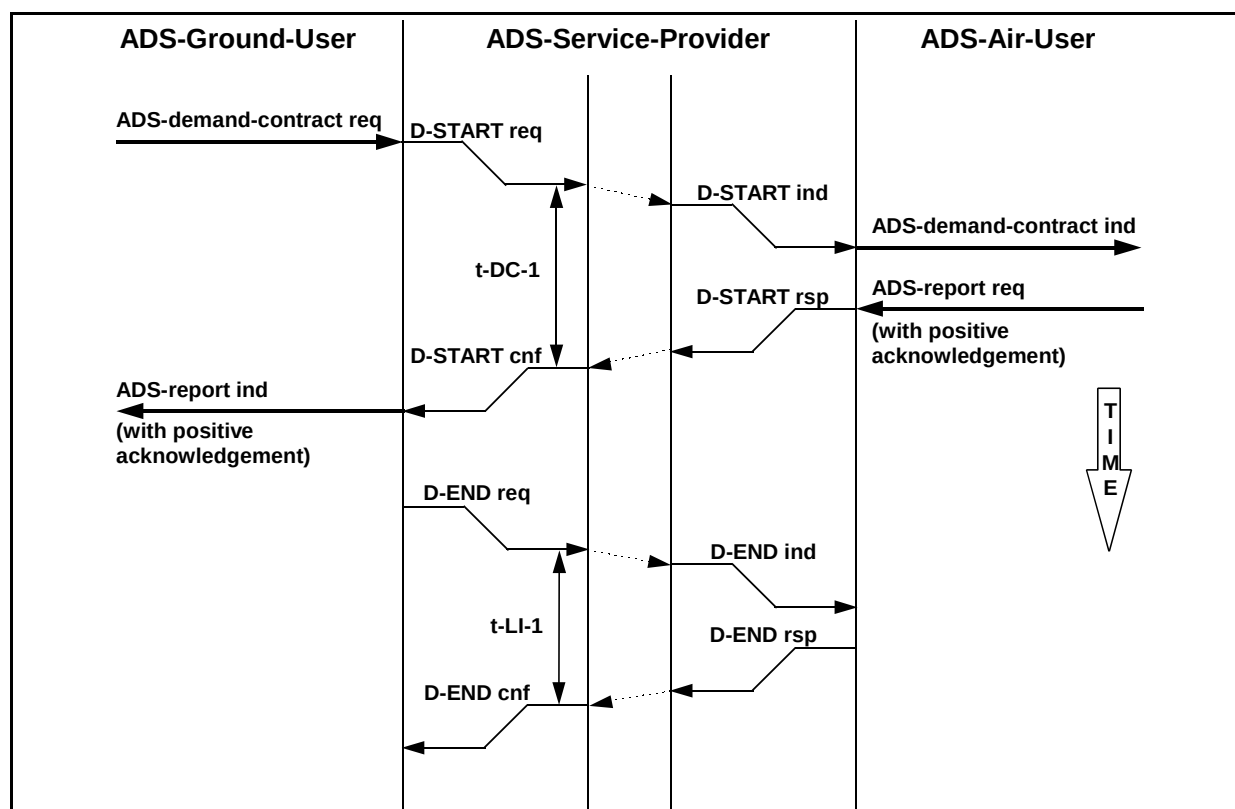


Figure 2.2.1.5-1: Use of demand contract with no dialogue existing

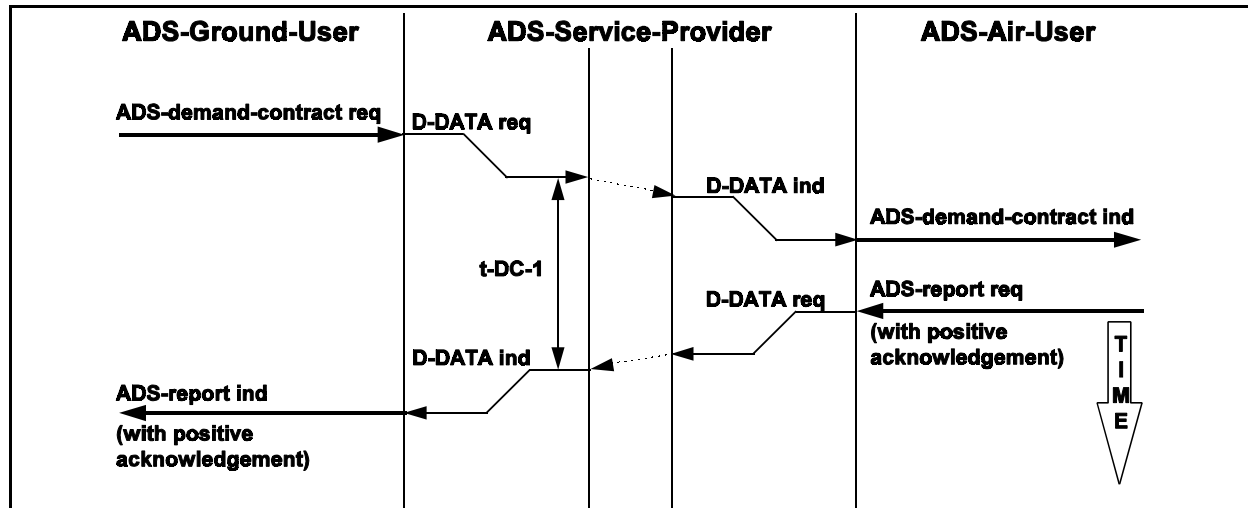


Figure 2.2.1.5-2: Use of demand contract with dialogue existing

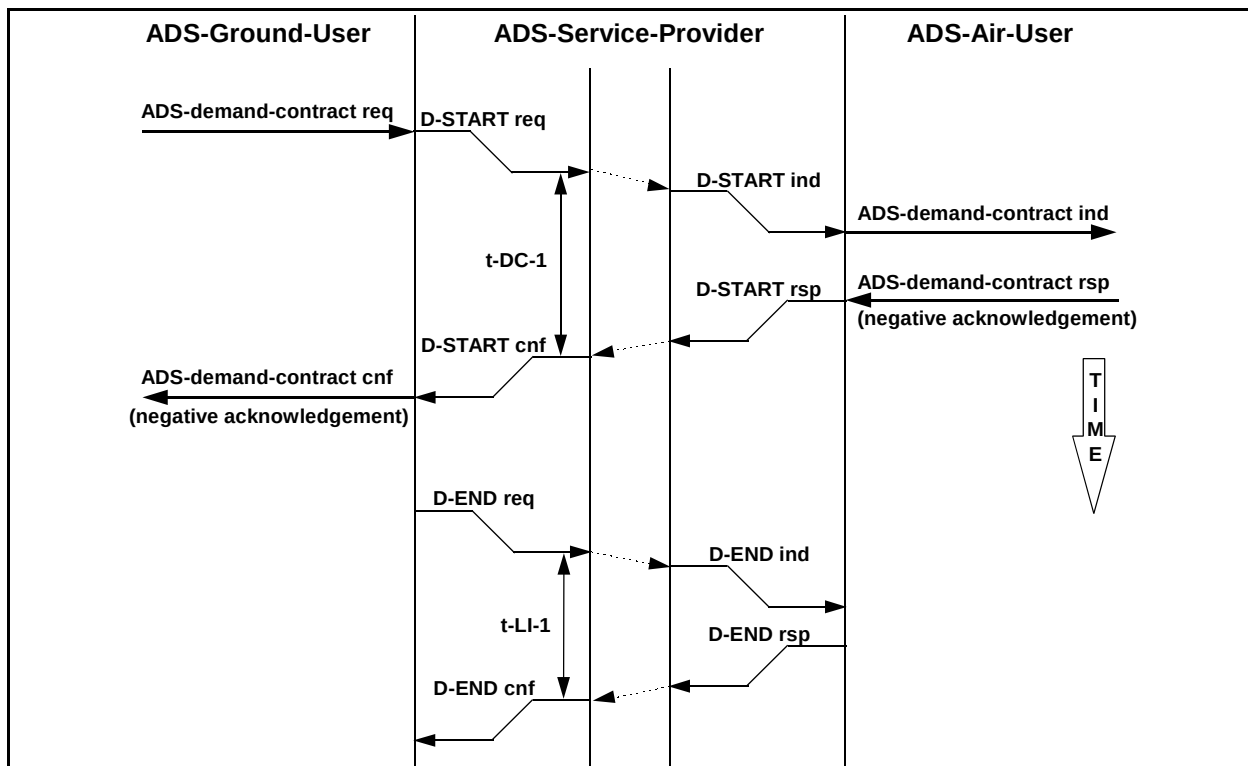


Figure 2.2.1.5-3: Use of demand contract with no dialogue existing

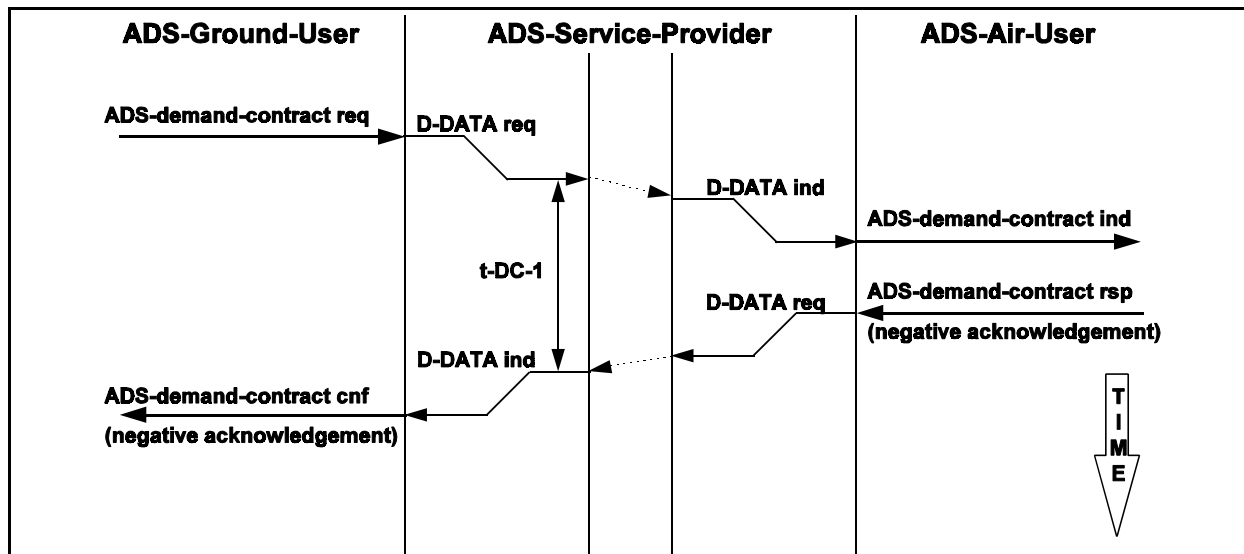


Figure 2.2.1.5-4: Use of demand contract with dialogue existing - with negative acknowledgement

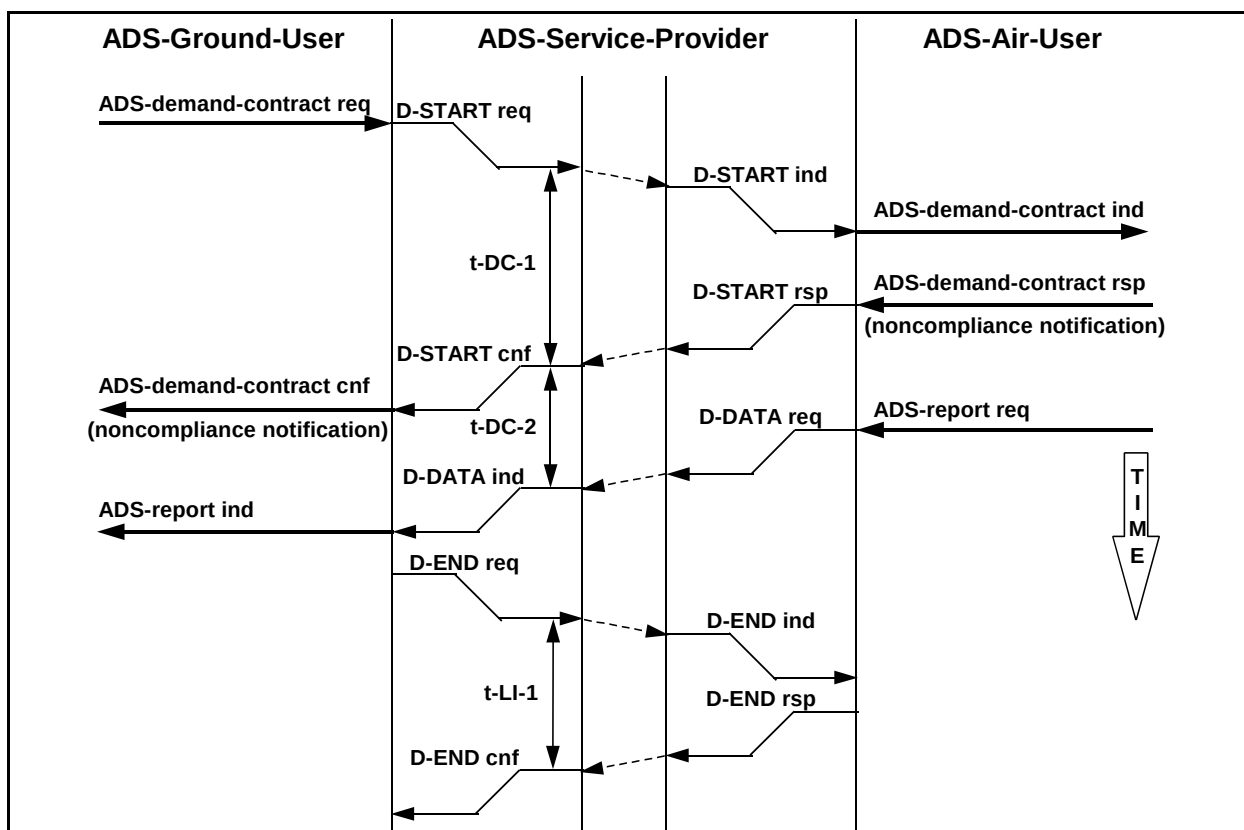
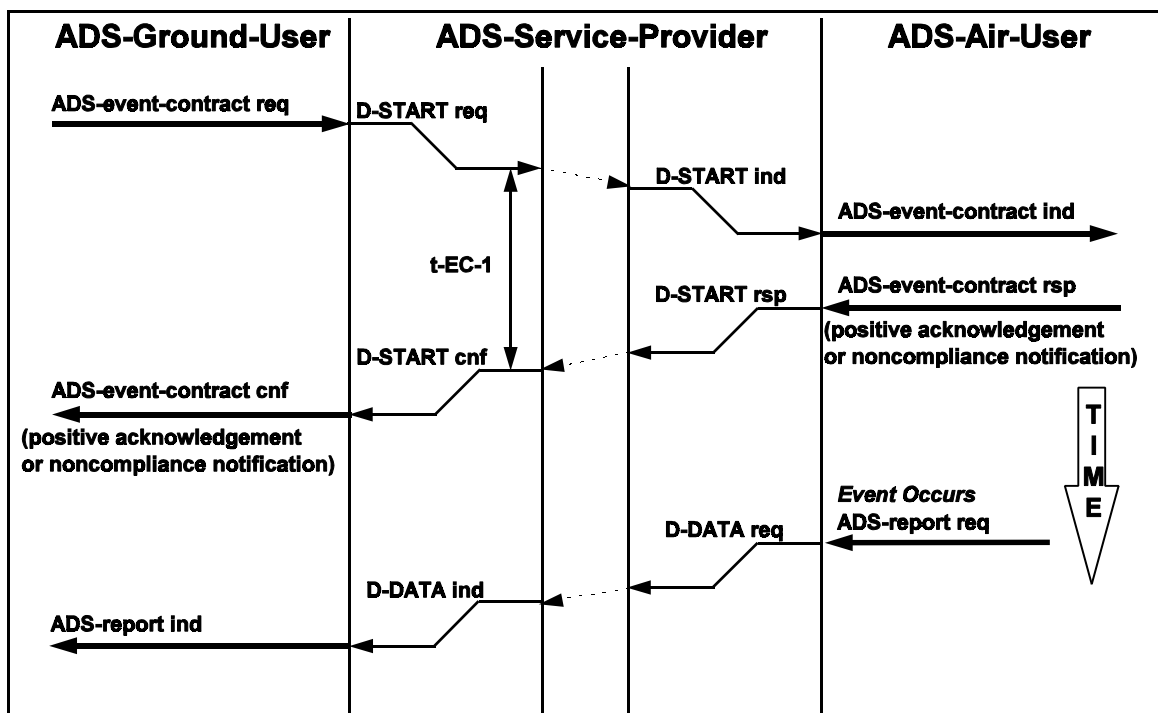
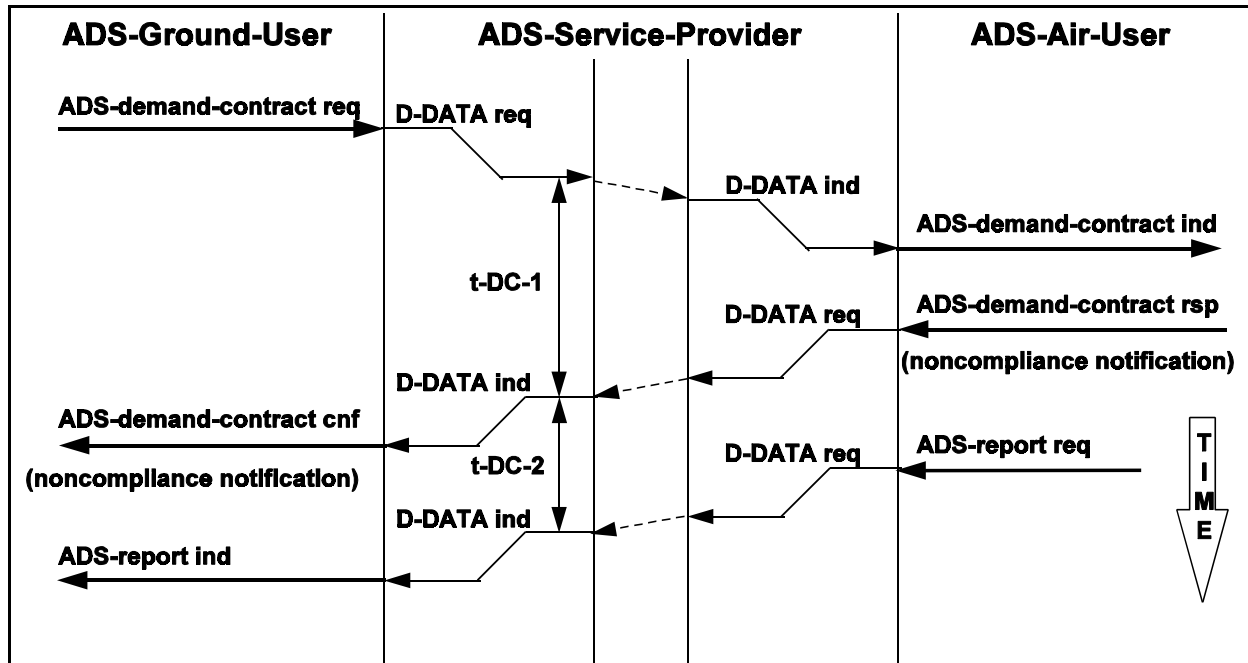


Figure 2.2.1.5-5: Use of demand contract with no dialogue existing - with noncompliance notification



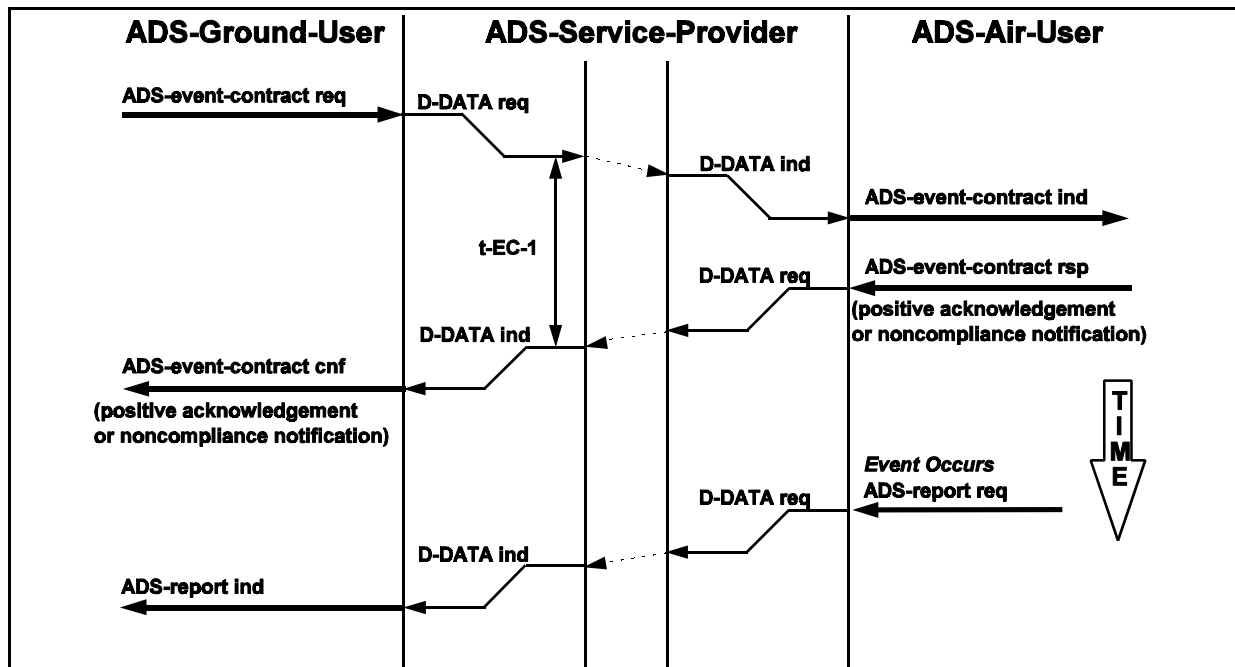


Figure 2.2.1.5-8: Use of event contract with dialogue existing - with positive acknowledgement or noncompliance notification

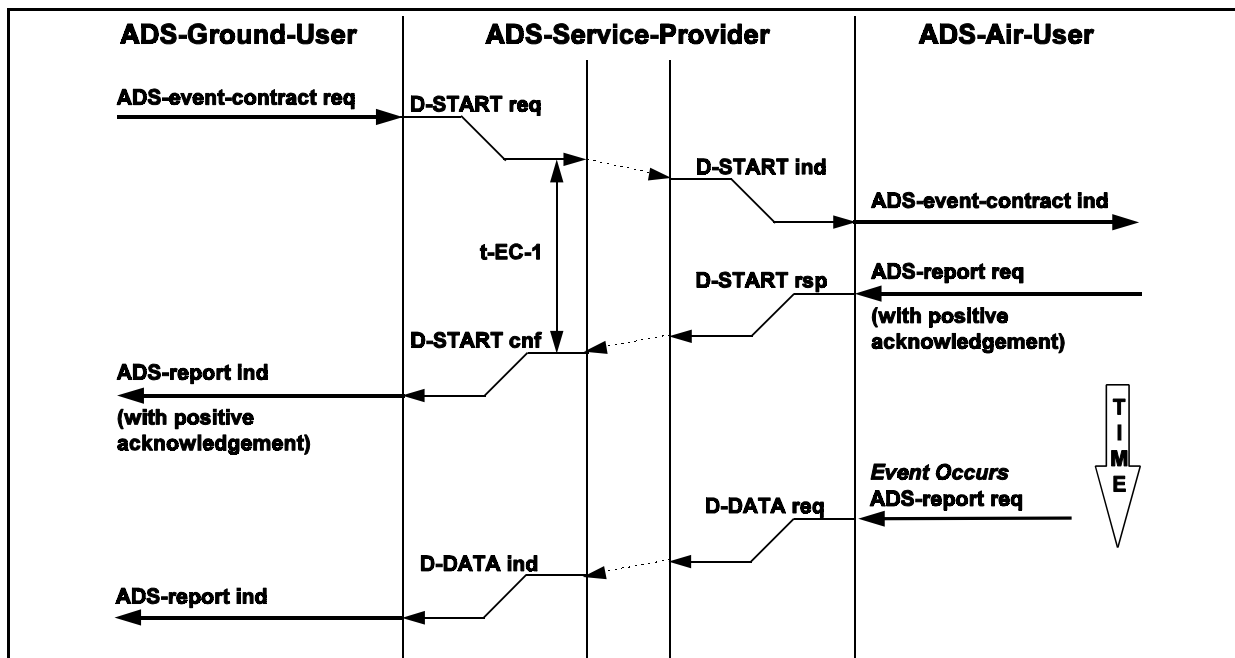


Figure 2.2.1.5-9: Use of event contract with no dialogue existing

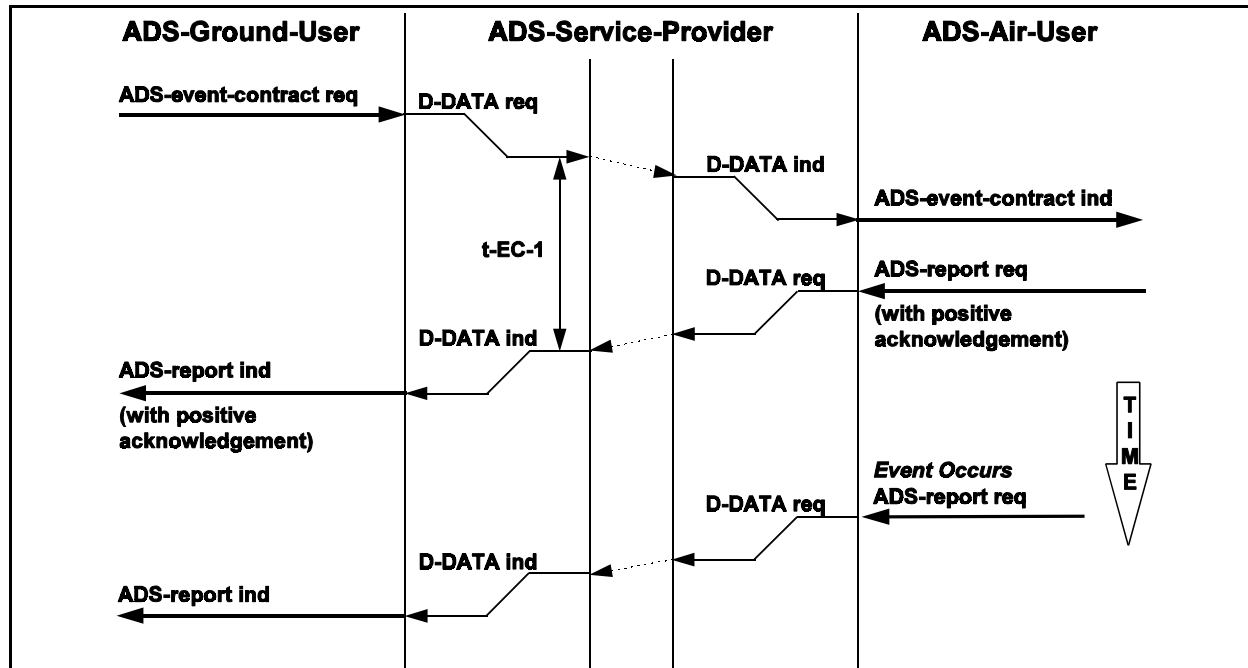


Figure 2.2.1.5-10: Use of event contract with dialogue existing

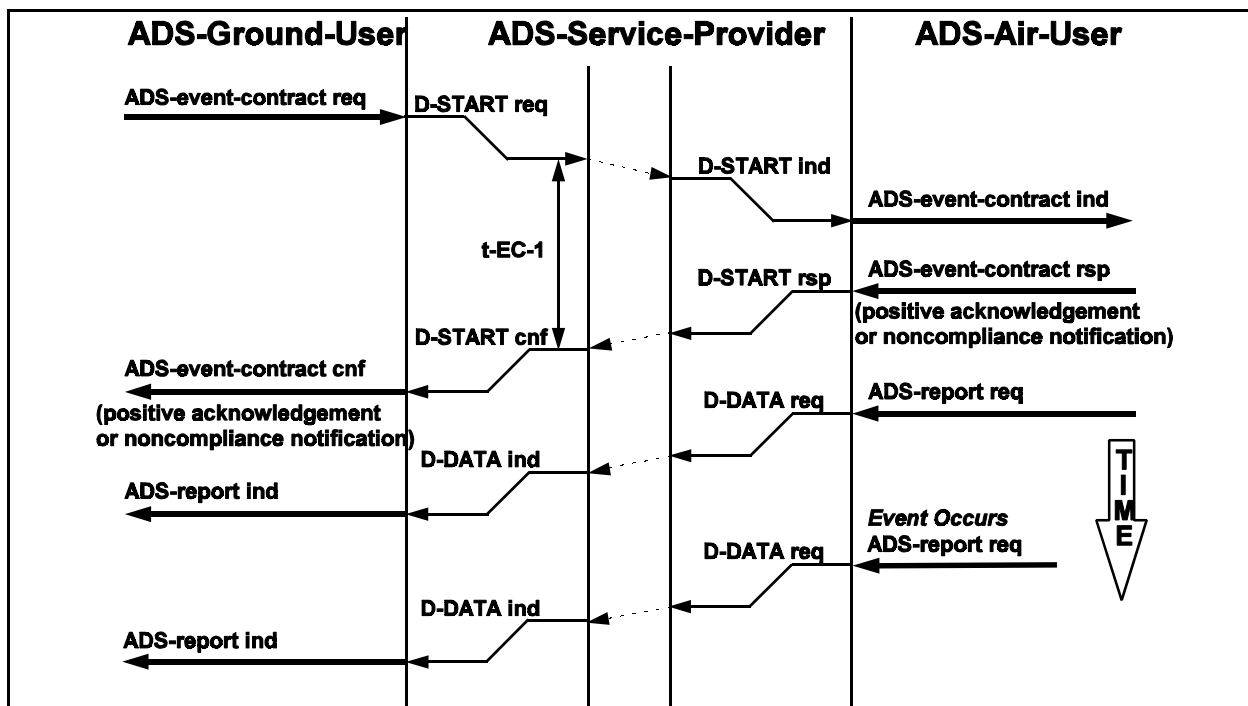


Figure 2.2.1.5-11: Use of event contract with no dialogue existing - with positive acknowledgement or noncompliance notification and immediate report

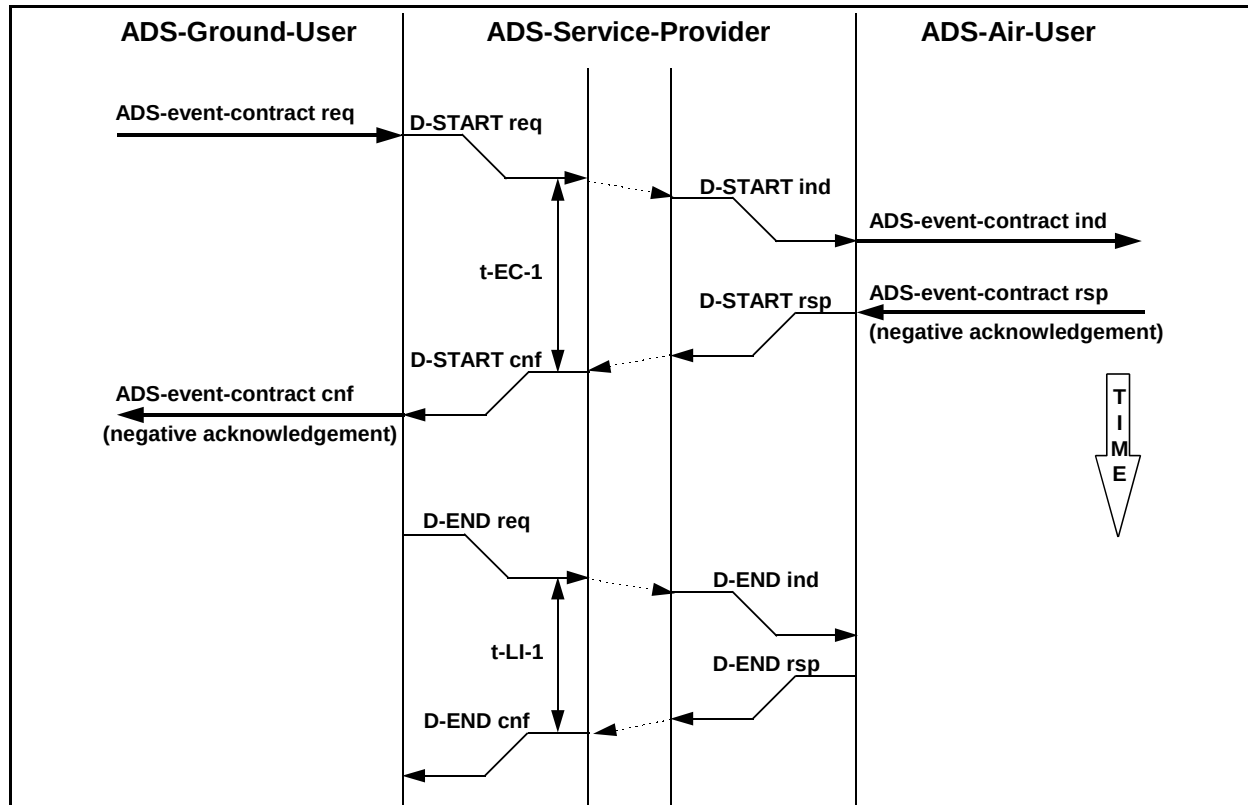
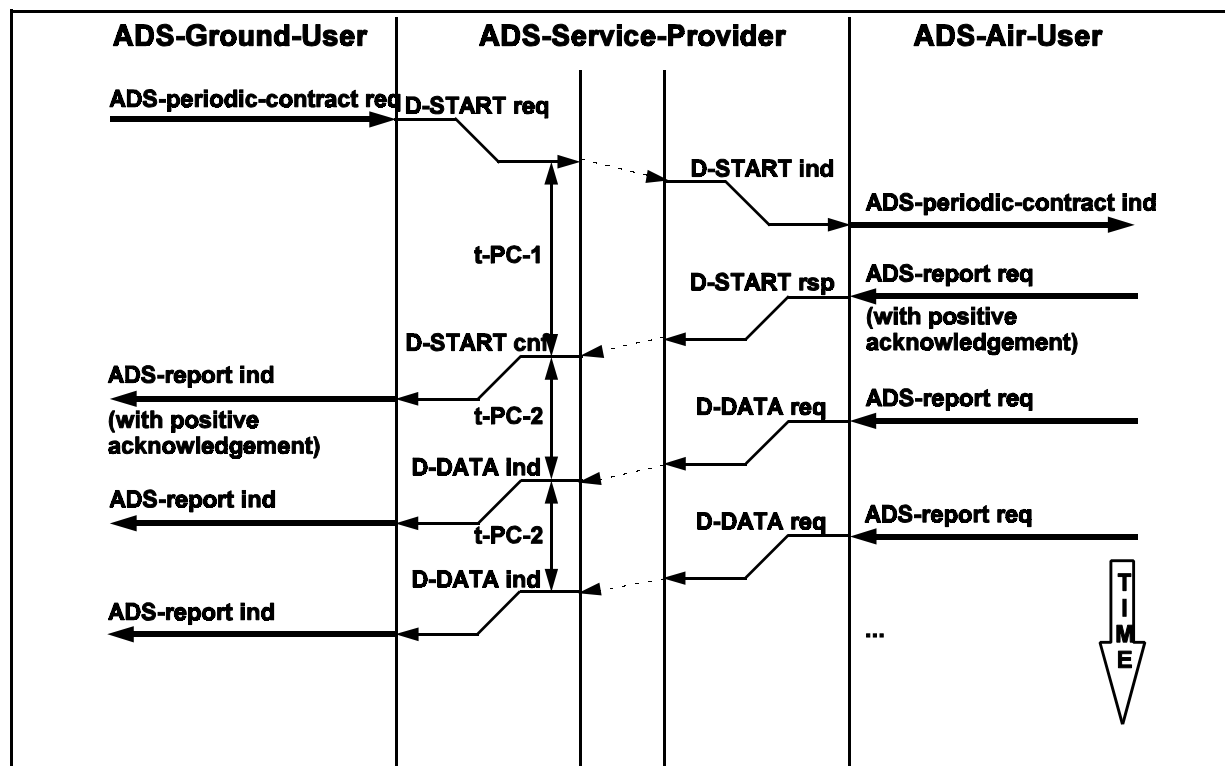
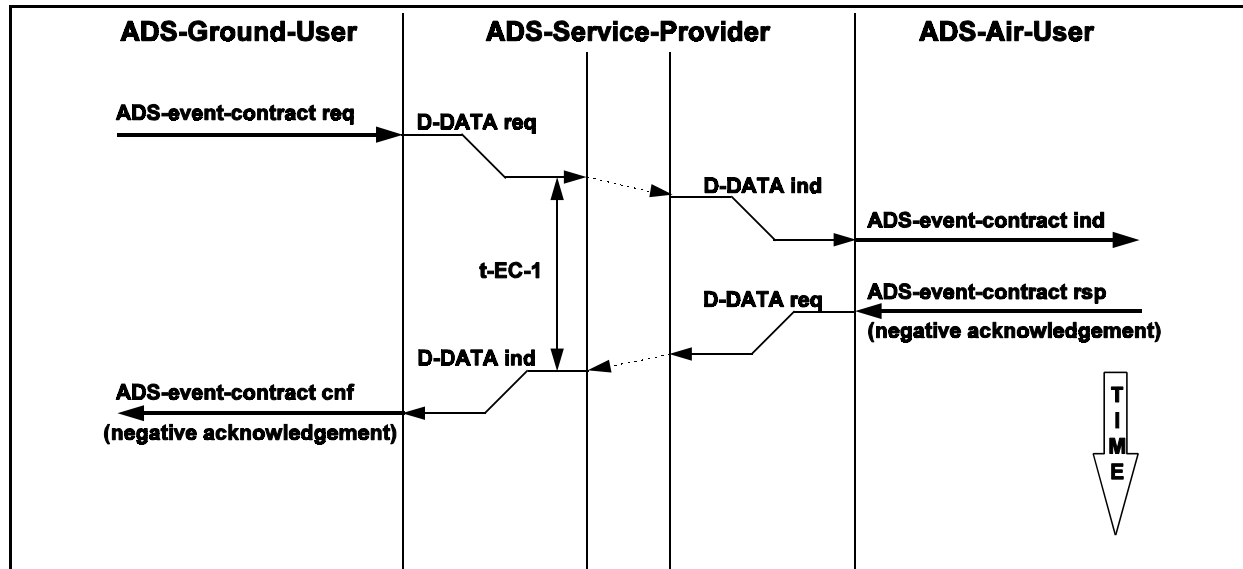


Figure 2.2.1.5-12: Use of event contract with no dialogue existing - with negative acknowledgement



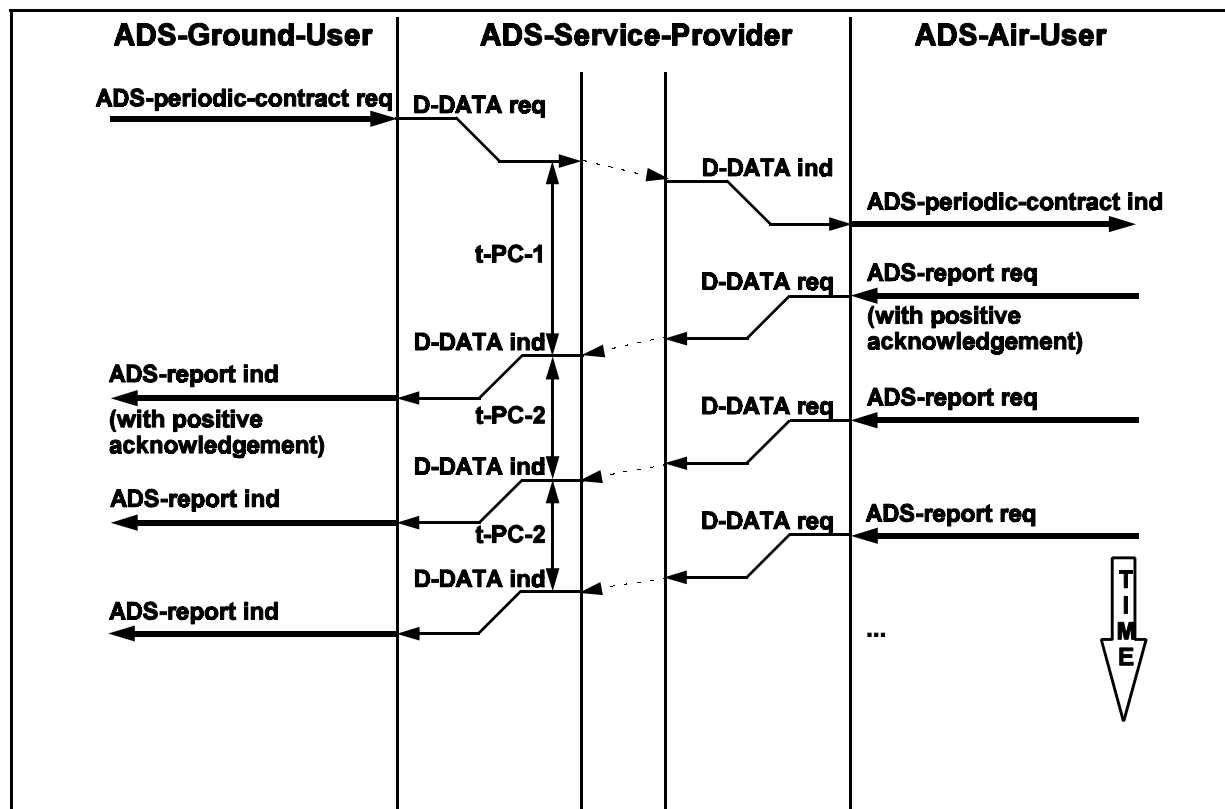


Figure 2.2.1.5-15: Use of periodic contract with a dialogue existing

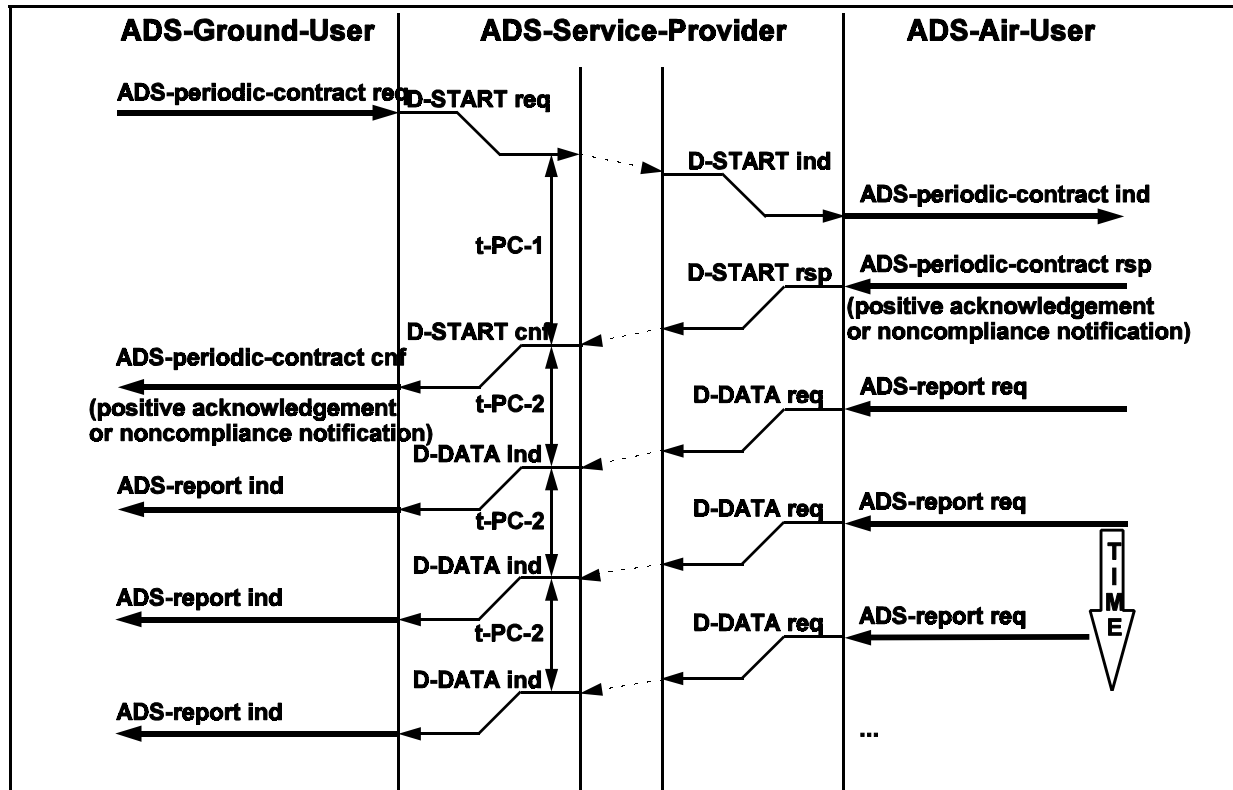


Figure 2.2.1.5-16: Use of periodic contract with no dialogue existing with positive acknowledgement or noncompliance notification

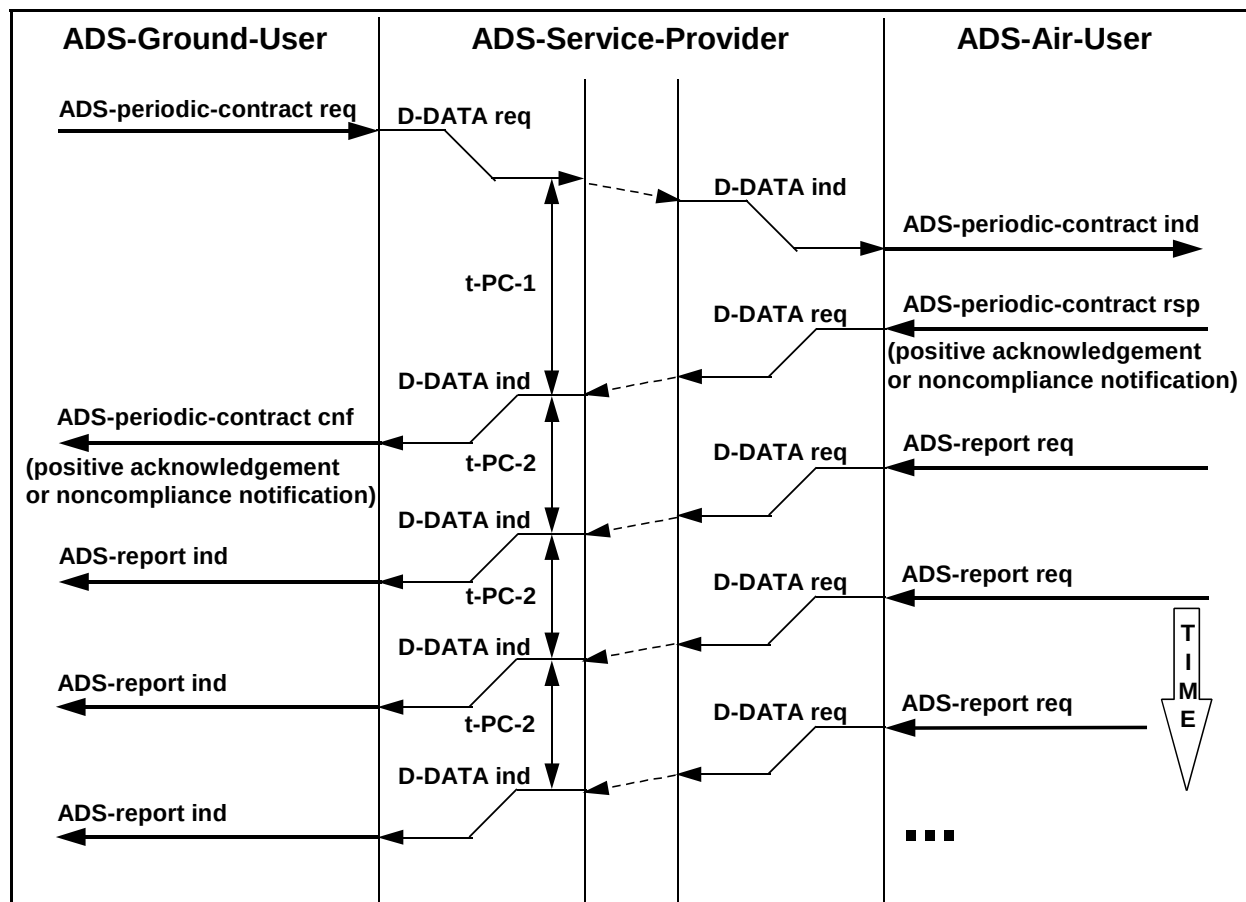


Figure 2.2.1.5-17: Use of periodic contract with dialogue existing with positive acknowledgement or noncompliance notification

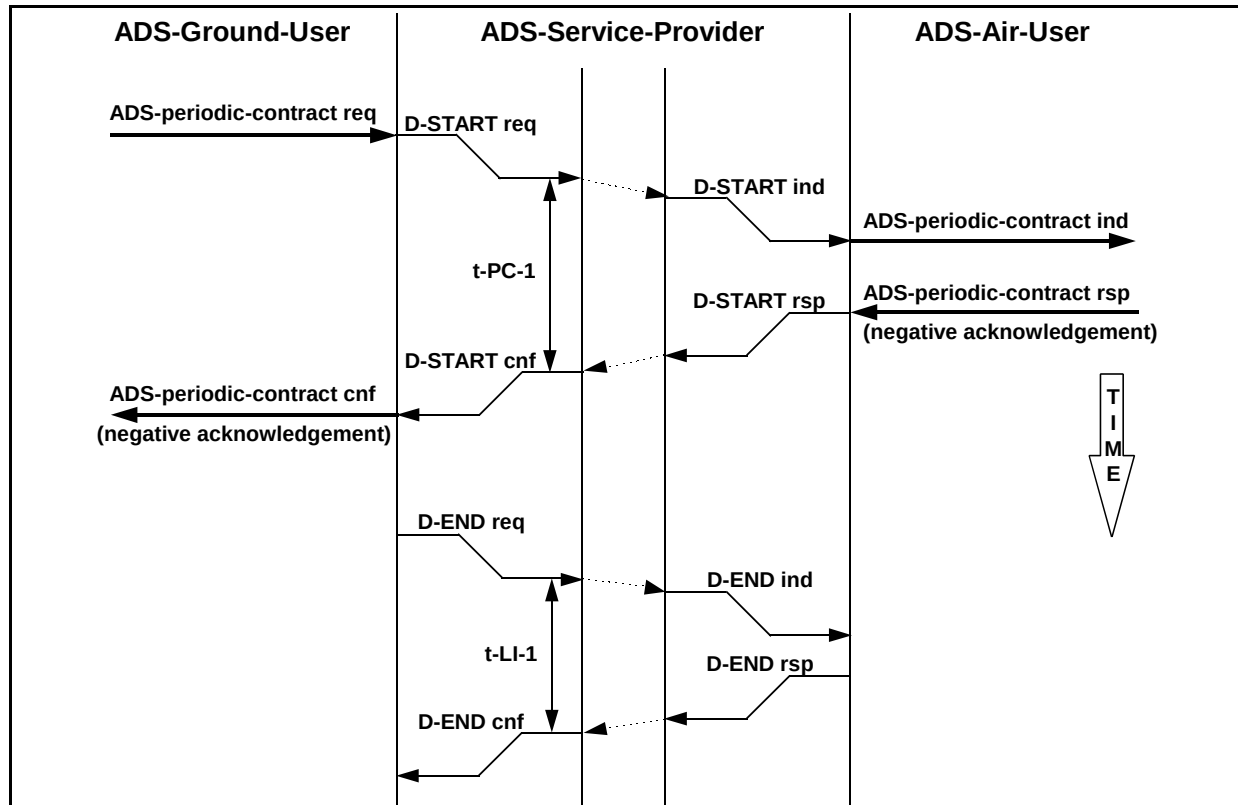


Figure 2.2.1.5-18: Use of periodic contract with no dialogue existing with negative acknowledgement

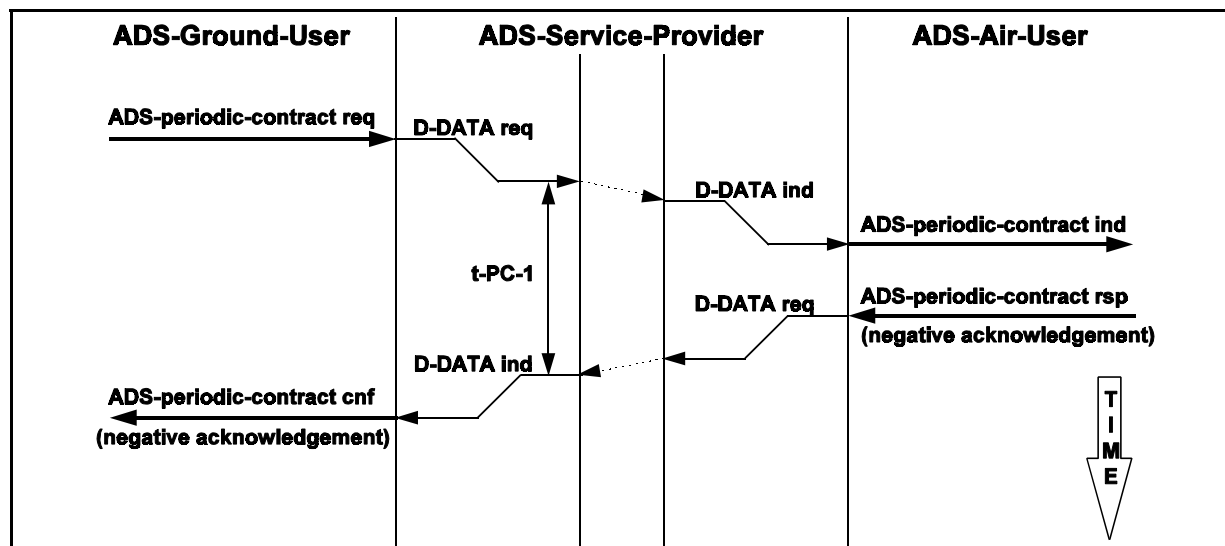


Figure 2.2.1.5-19: Use of periodic contract with a dialogue existing with negative acknowledgement

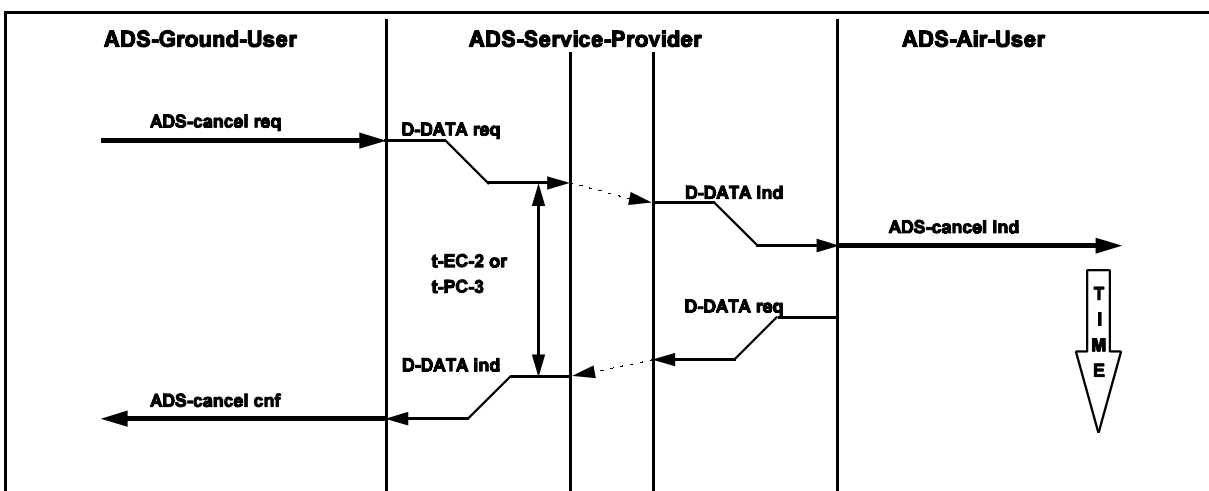


Figure 2.2.1.5-20: Use of ADS cancel contract service

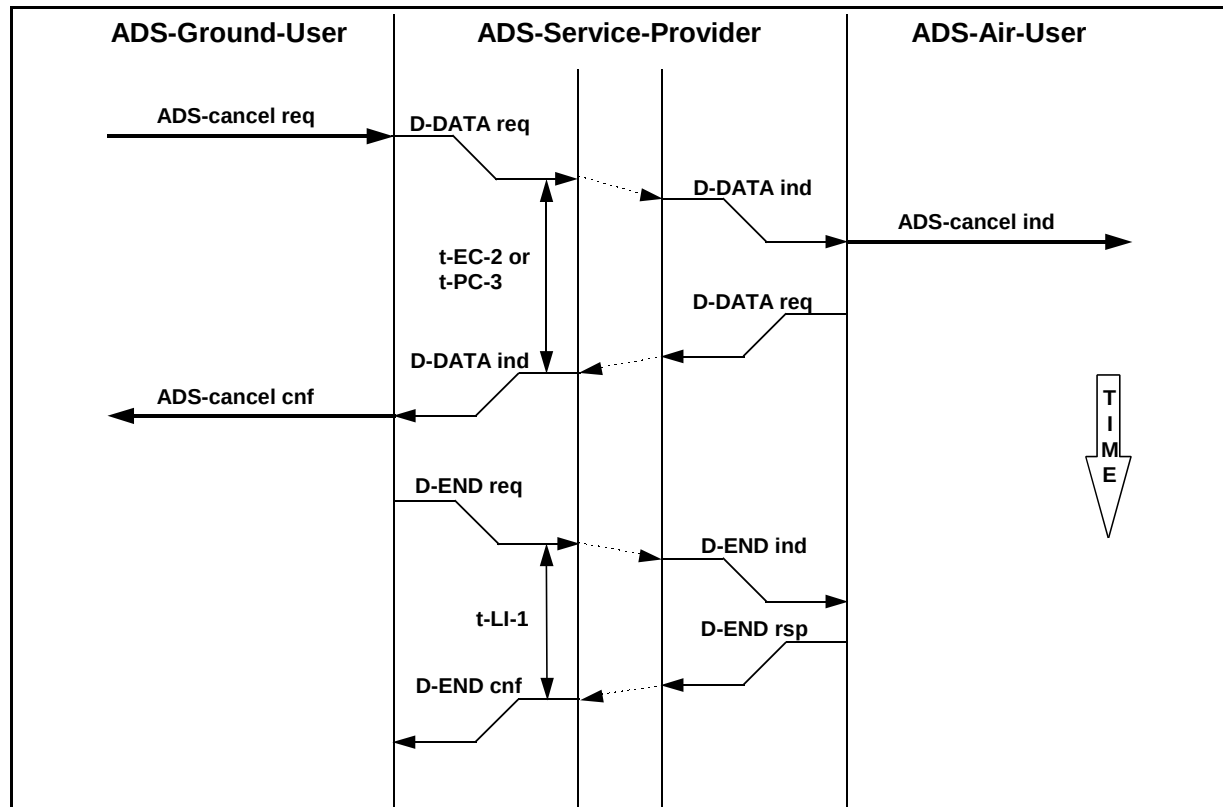


Figure 2.2.1.5-21: Use of ADS cancel contract service with only one contract

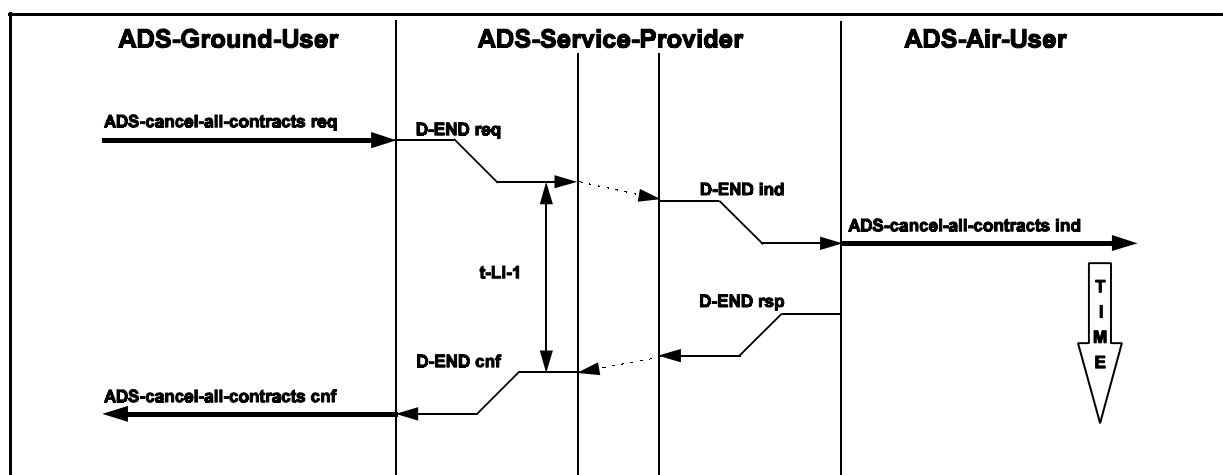


Figure 2.2.1.5-22: Use of ADS cancel all contracts service

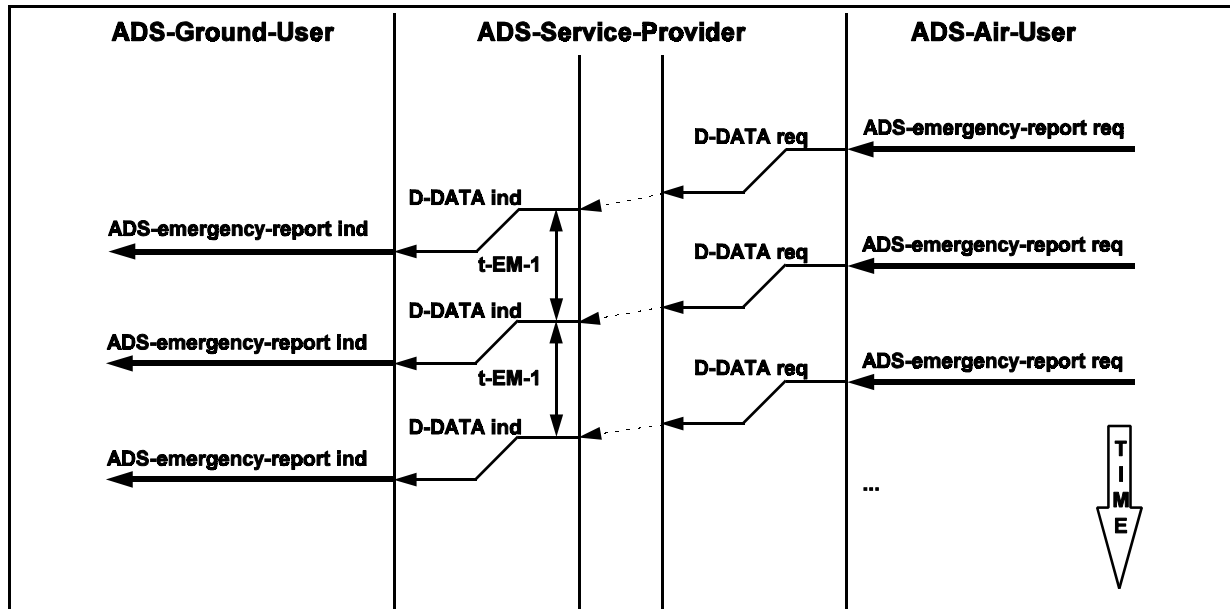


Figure 2.2.1.5-23: Use of emergency report service

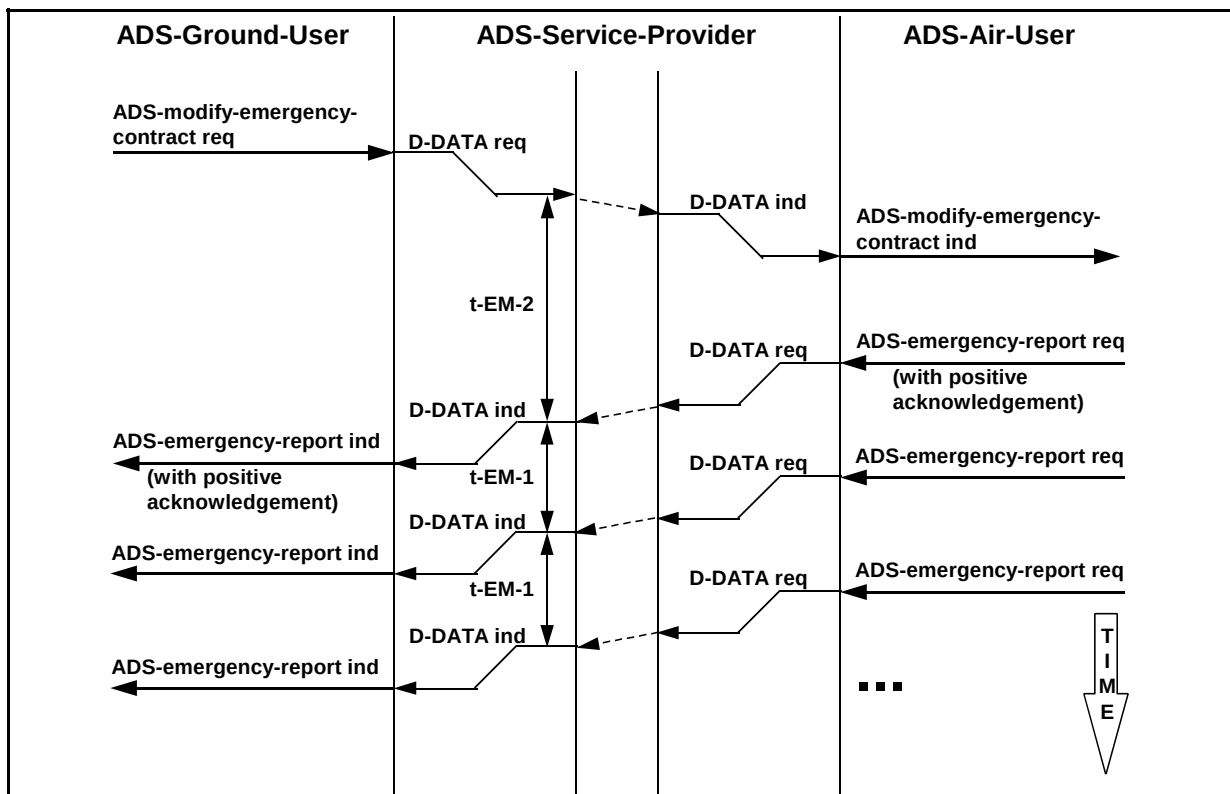


Figure 2.2.1.5-24: Modification of emergency contract

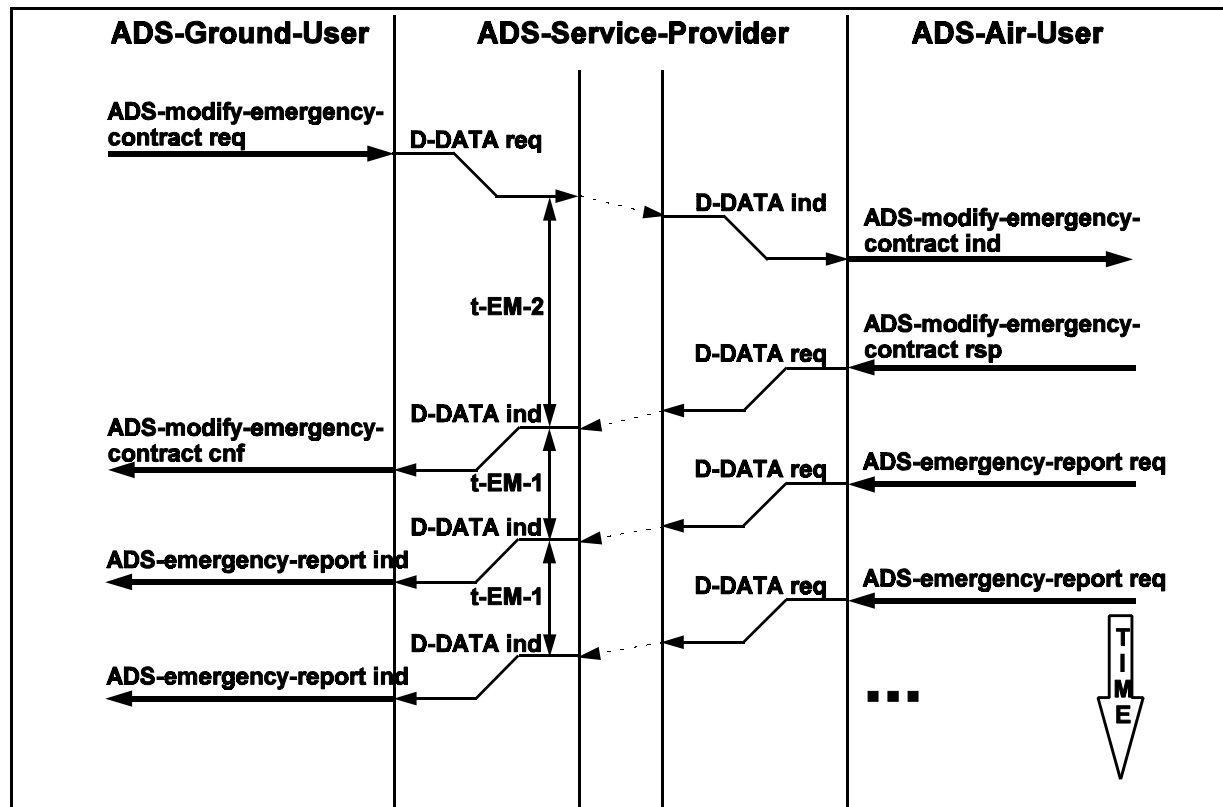


Figure 2.2.1.5-25: Modification of emergency contract rejected

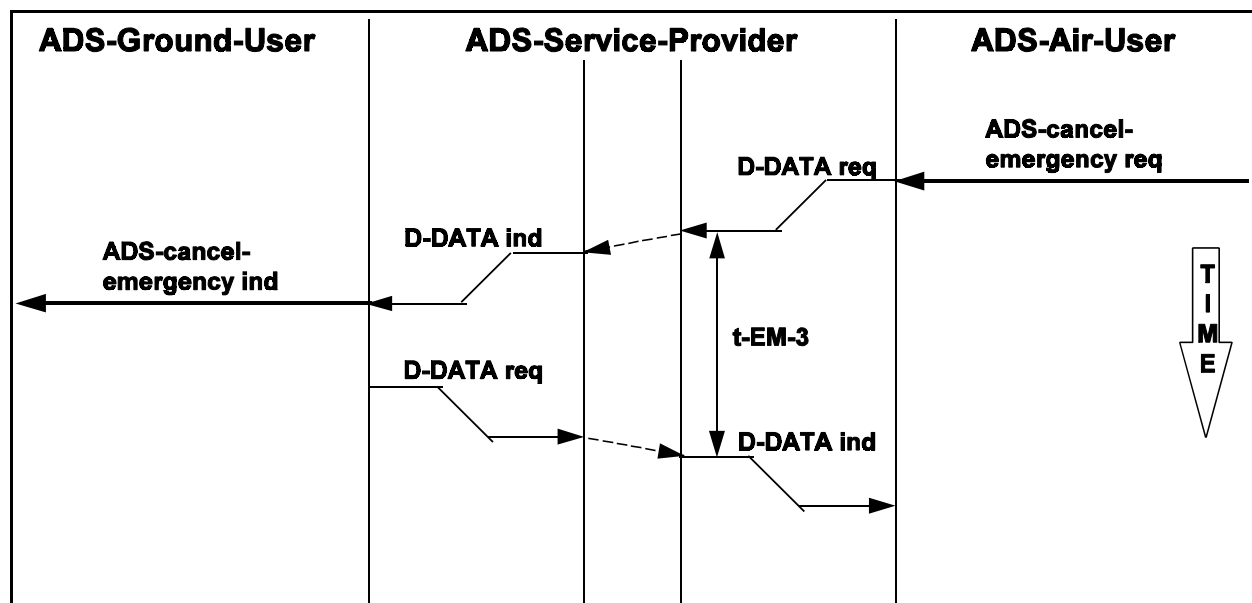


Figure 2.2.1.5-26: Cancellation of emergency contract

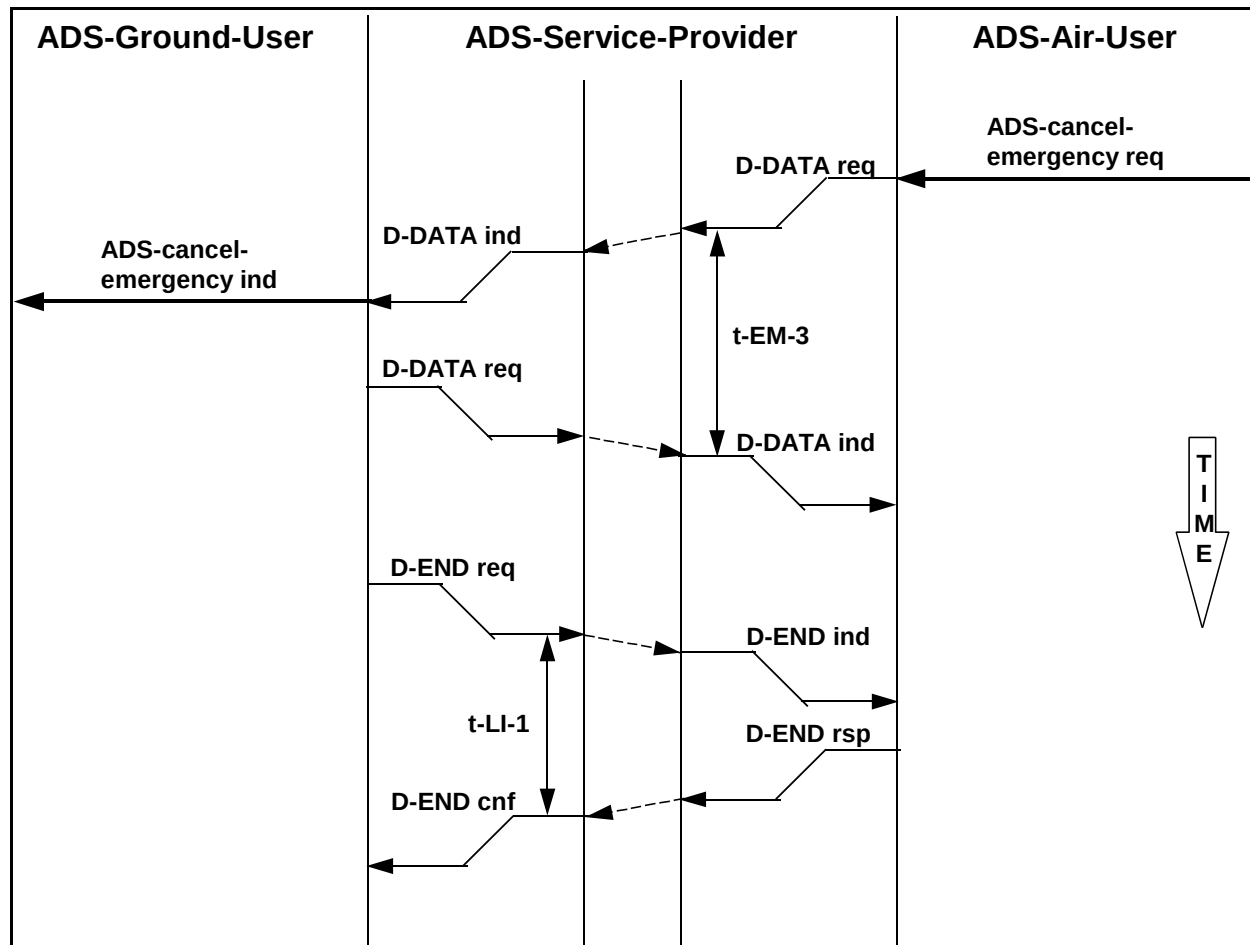


Figure 2.2.1.5-27: Cancellation of emergency contract with no other contracts in place

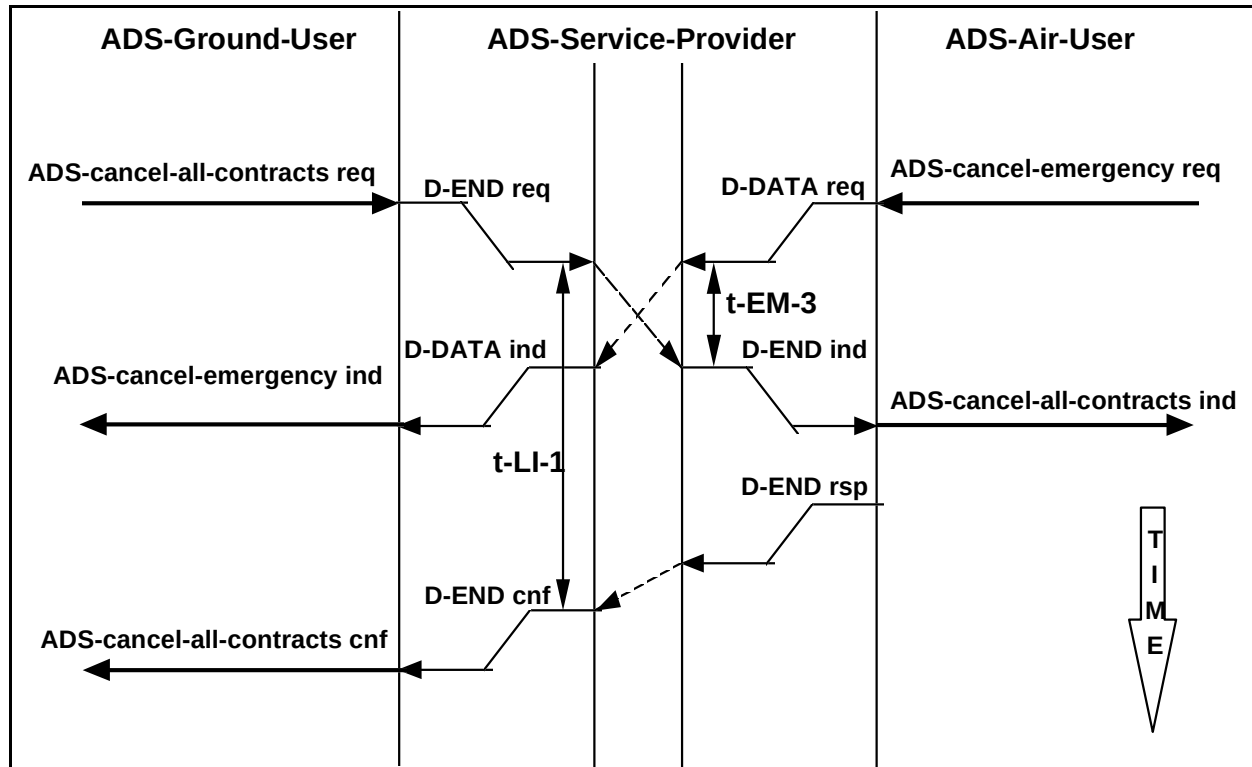


Figure 2.2.1.5-28: Crossed air emergency cancellation and cancel all contracts

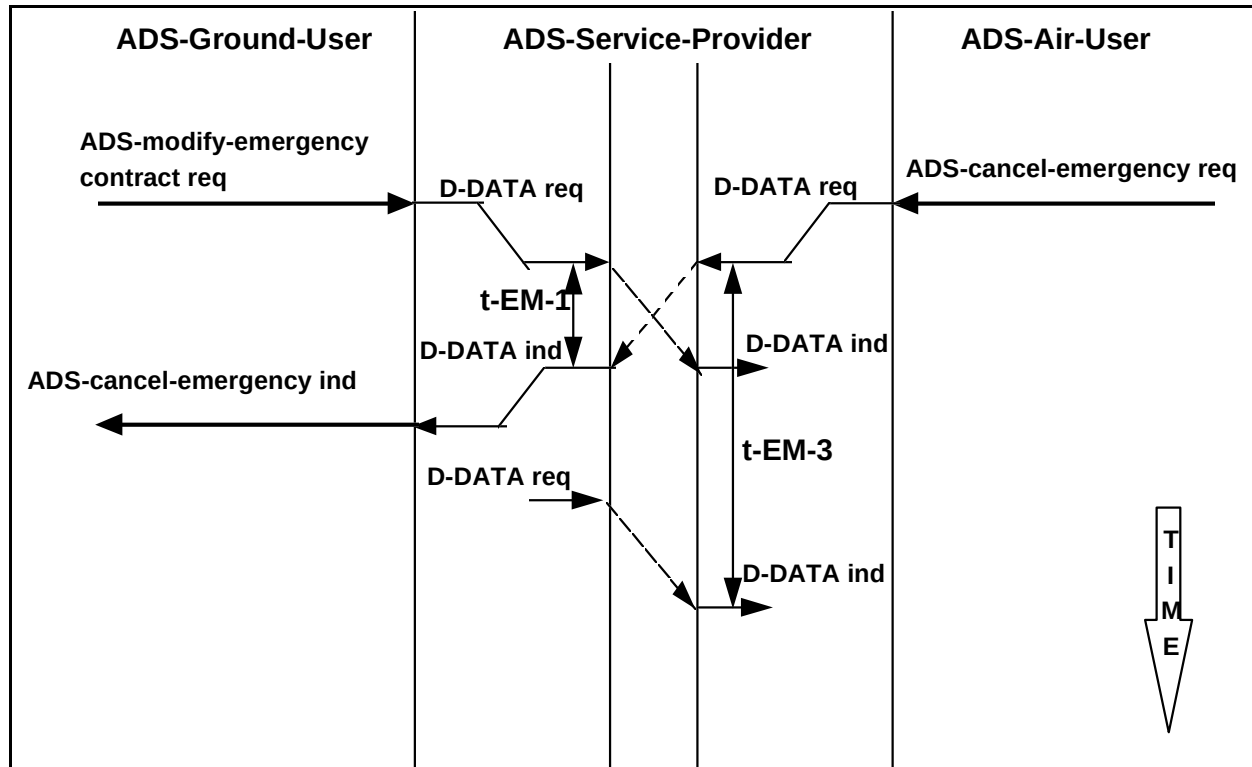


Figure 2.2.1.5-29: Crossed air emergency cancellation and modification of emergency contract with other contracts in place

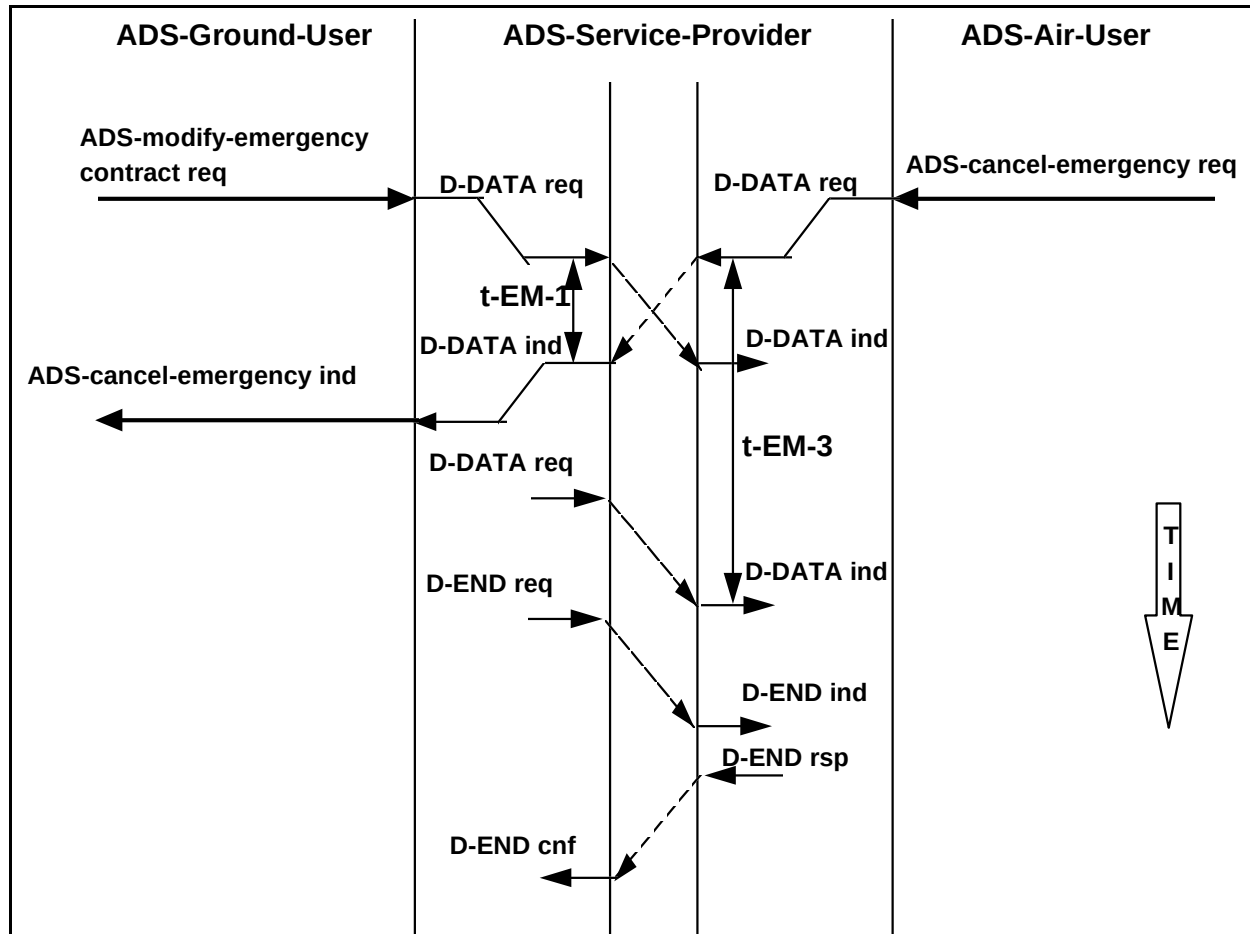


Figure 2.2.1.5-30: Crossed air emergency cancellation and modification of emergency contract with no other contracts in place

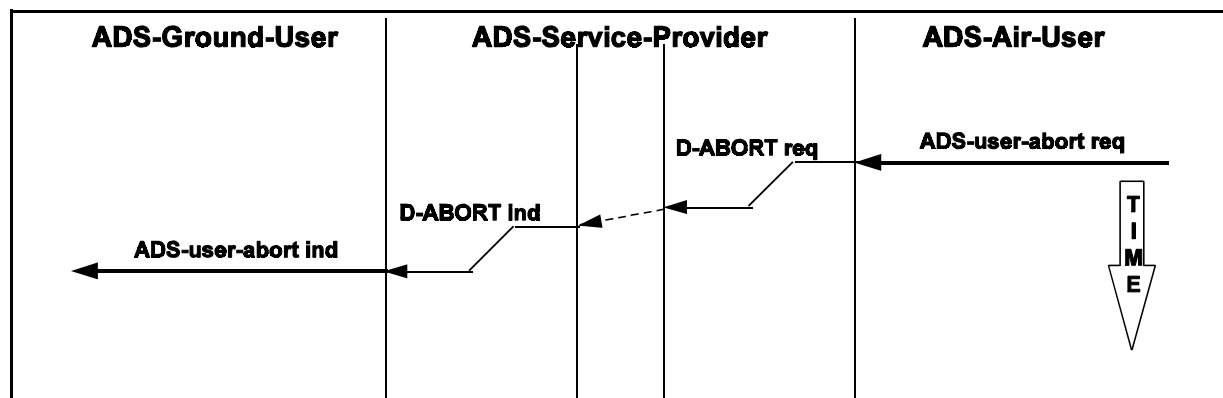


Figure 2.2.1.5-31: Air user abort service

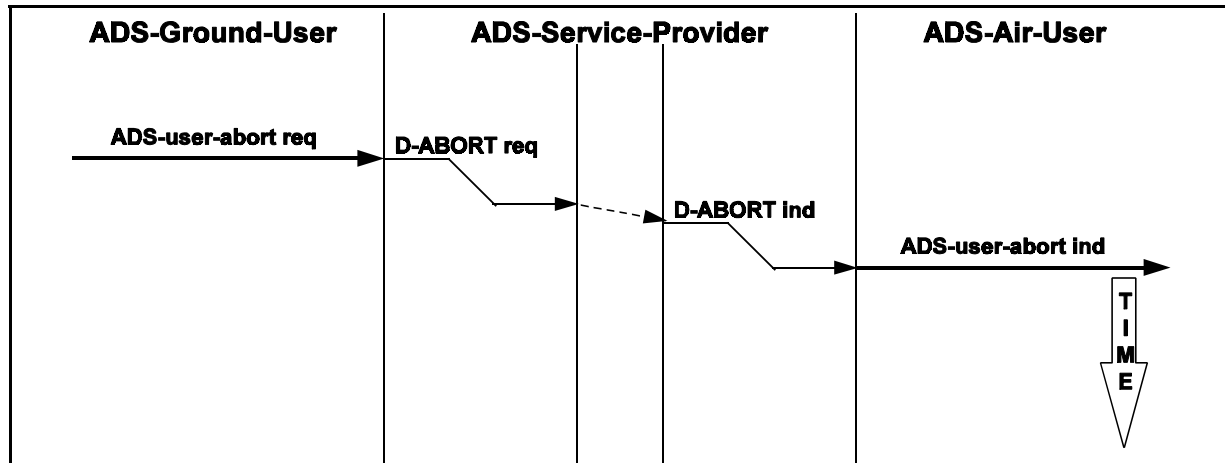


Figure 2.2.1.5-32: Ground user abort service

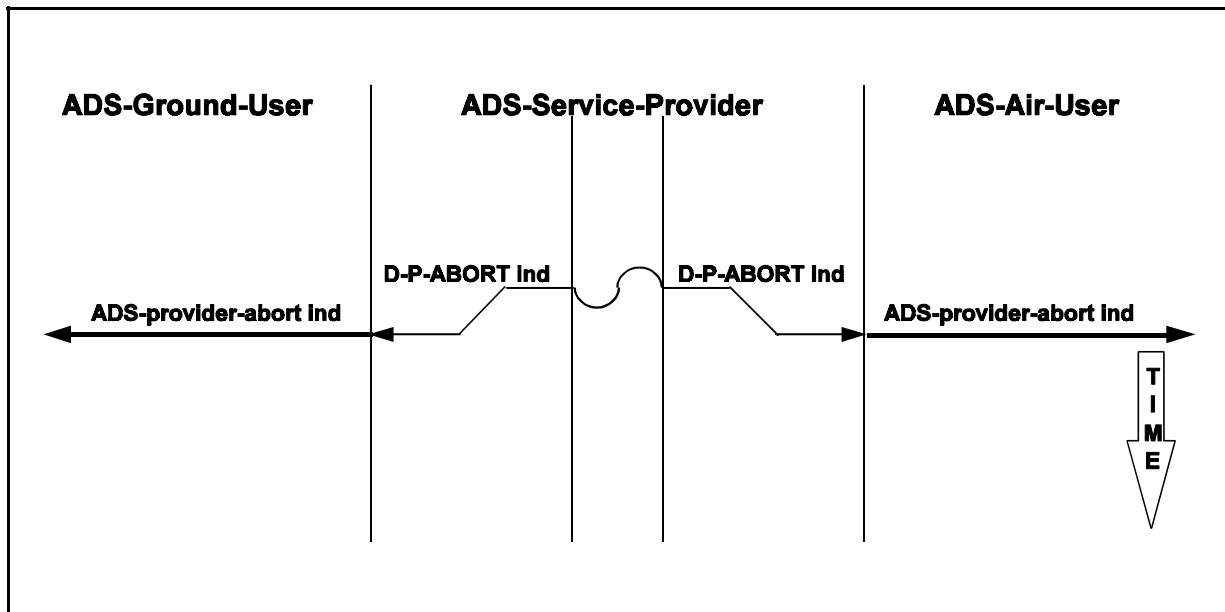


Figure 2.2.1.5-33: Dialogue service provider abort service

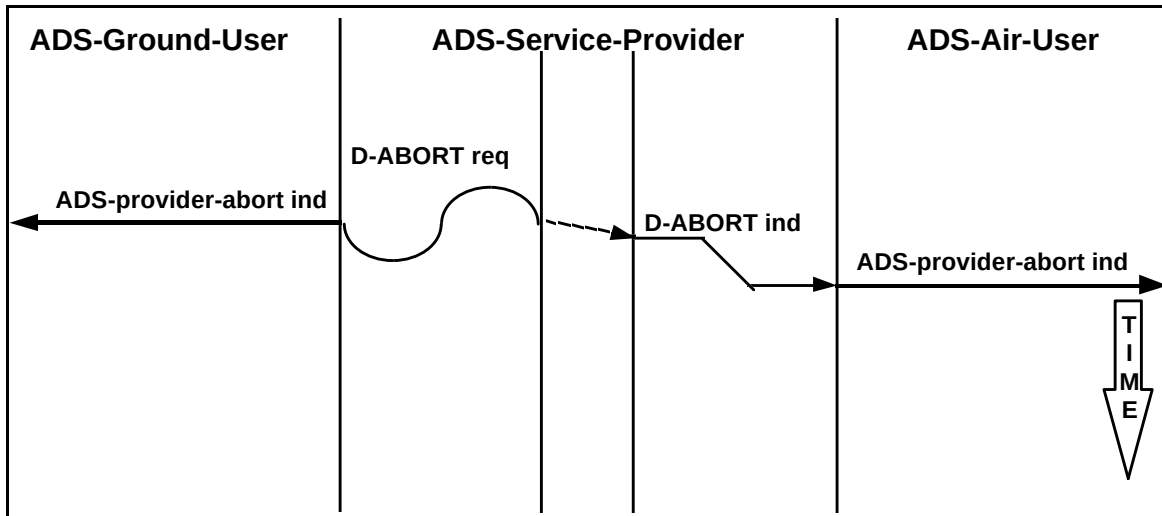


Figure 2.2.1.5-34: Ground ASE abort

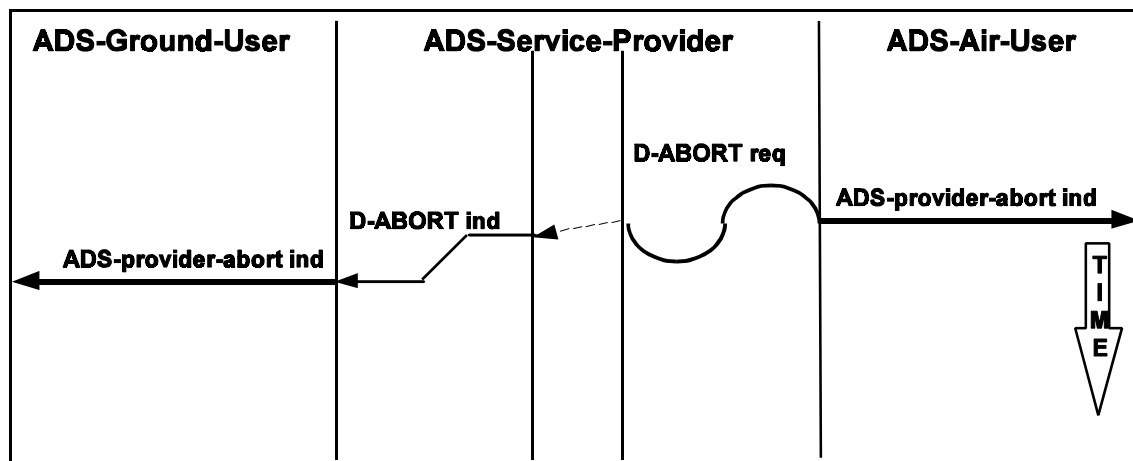


Figure 2.2.1.5-35: Air ASE abort

2.2.1.5.2 ADS Service Provider Timers

2.2.1.5.2.1 The ADS-ASE shall be capable of detecting when a timer expires.

Note 1.— Table 2.2.1.5-1 lists the time constraints related to the ADS application. Each time constraint requires a timer to be set in the ADS protocol machine.

Note 2.— If the timer expires before the final event has occurred, the ADS ASE takes the appropriate action as defined in 2.2.1.5.4.1.

2.2.1.5.2.2 **Recommendation.**—The timer values should be as indicated in Table 2.2.1.5-1.

Table 2.2.1.5-1: ADS Service Provider Timers

ADS Service	Timer	Timer Value	Timer Start Event	Timer Stop Event
ADS-demand-contract	t-DC-1	6 minutes	ADS-demand-contract request	ADS-demand-contract confirmation or ADS-report indication
	t-DC-2	3 minutes 30 seconds	ADS-demand-contract confirmation	ADS-report indication
ADS-event-contract	t-EC-1	6 minutes	ADS-event-contract request	ADS-event-contract confirmation or ADS-report indication
	t-EC-2	6 minutes	ADS-cancel request	ADS-cancel-contract confirmation
ADS-periodic-contract	t-PC-1	6 minutes	ADS-periodic-contract request	ADS-periodic-contract confirmation or ADS-report indication
	t-PC-2	reporting rate + 3 minutes	ADS-report indication or ADS-periodic-contract confirmation	ADS-report indication
	t-PC-3	6 minutes	ADS-cancel request	ADS-cancel-contract confirmation
ADS emergency contract	t-EM-1	reporting rate + 3 minutes	ADS-emergency-report indication	ADS-emergency-report indication

ADS Service	Timer	Timer Value	Timer Start Event	Timer Stop Event
	t-EM-2	6 minutes	ADS-modify-emergency-contract request	ADS-modify-emergency-contract confirmation or ADS-emergency-report indication
	t-EM-3	6 minutes	ADS-cancel-emergency request	Arrival of ADS-cancel-emergency PDU
General	t-LI-1	6 minutes	D-END request	D-END confirmation

Note.— The receipt of ADS-user-abort request, D-ABORT indication or D-P-ABORT indication are also timer stop events.

2.2.1.5.3 ADS-ASE Protocol Description

2.2.1.5.3.1 Description

2.2.1.5.3.1.1 ADS-ASE Functional Model

Note 1.— The ADS-ground-ASE is functionally made of 7 modules as shown in figure 2.2.1.5-36 and the ADS-air-ASE is functionally made of a similar 7 modules as shown in figure 2.2.1.5-37:

- a) *the High Interface Module (HI module). This module interfaces with the ASE-user through the abstract service interface as defined in 2.2.1.3.*
- b) *the ADS Demand Contract Module (DC module): the DC module manages all demand contracts with a single ground system.*
- c) *the ADS Event Contract Module (EC module): the EC module manages event contracts with a single ground system.*
- d) *the ADS Periodic Contract Module (PC module): the PC module manages periodic contracts with a single ground system.*
- e) *the ADS Emergency Module (EM module): the EM module manages emergency contracts with a single ground system.*
- f) *the ADS Abort Module (AB module): the AB module handles aborts in case of irrecoverable error.*
- g) *the Low Interface Module (LI module). This module interfaces the Dialogue Service Provider on behalf of the DC, EC, PC, EM and AB modules.*

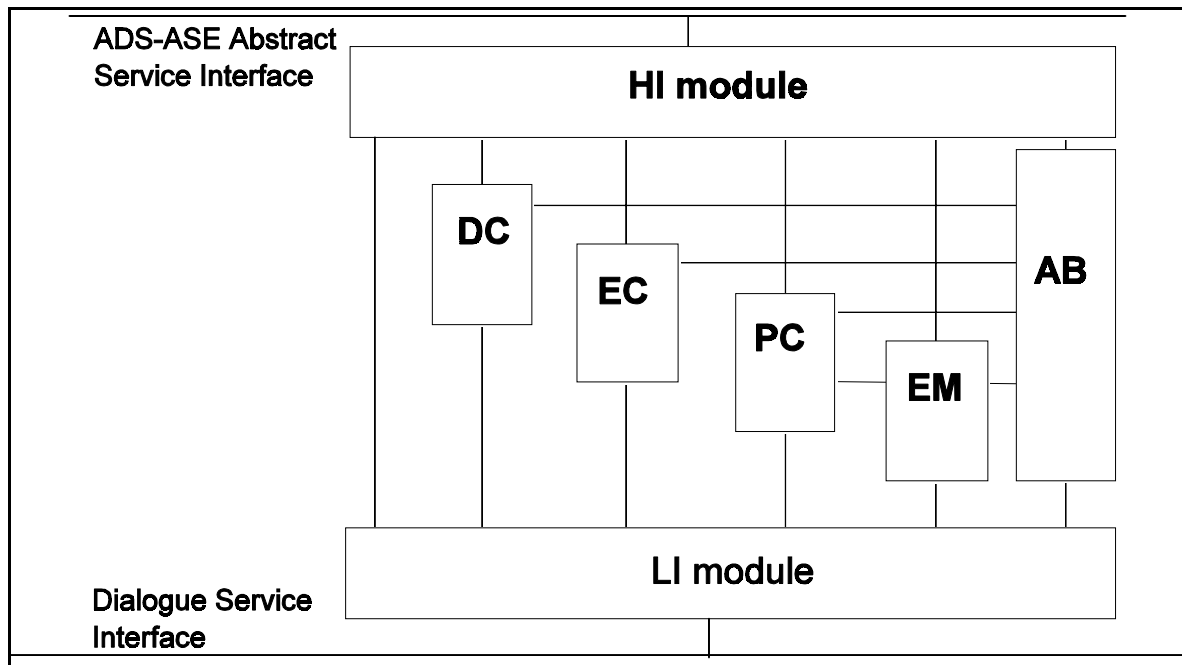


Figure 2.2.1.5-36: Functional model of the ADS-ground-ASE

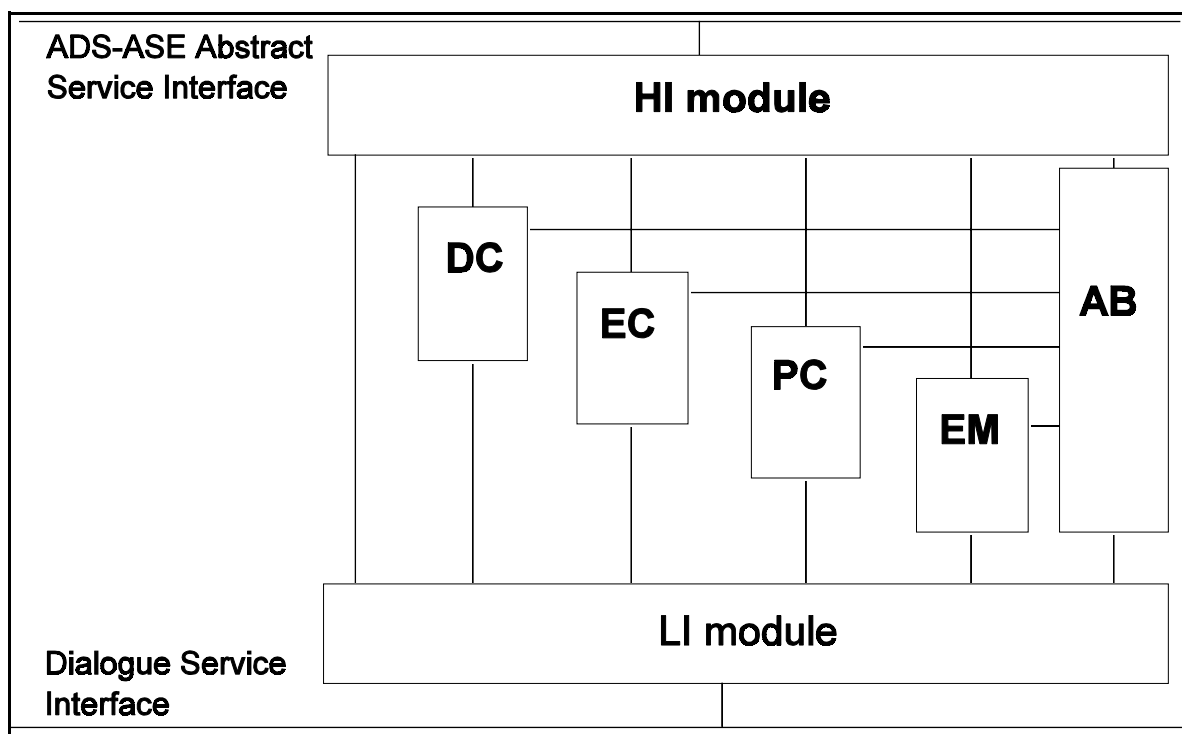


Figure 2.2.1.5-37: Functional model of the ADS-air-ASE

Note 2.— The only difference between the ADS-ground-ASE and the ADS-air-ASE functional models is that in the ADS-air-ASE, there is no communication between the PC and EM modules.

Note 3.— 2.2.1.5.3 describes the actions of the individual modules in both the air and ground systems. 2.2.1.5.6 contains state Tables for the individual modules.

Note 4.— The ADS-ground-user is considered an active user from the time at which it invokes the first ADS-demand-contract request, an ADS-event-contract request or an ADS-periodic-contract request until such time that:

- a) the ADS-ground-user receives an ADS-cancel-all-contracts confirmation,*
- b) the ADS-ground-user receives an ADS-cancel confirmation, and there are no other contracts in place,*
- c) the ADS-ground-user receives an ADS-cancel-emergency-contract indication, and there are no other contracts in place,*
- d) the ADS-ground-user receives an ADS-demand-contract confirmation, an ADS-event-contract confirmation or an ADS-periodic-contract confirmation, with the Reply parameter value set to “negative acknowledgement”, and there are no other contracts in place,*
- e) the ADS-ground-user receives an ADS-report indication, with the Contract type parameter value set to “demand contract”, and there are no other contracts in place,*
- f) the ADS-ground-user receives an ADS-user-abort indication,*
- g) the ADS-ground-user receives an ADS-provider-abort indication, or*
- h) the ADS-ground-user invokes an ADS-user-abort request.*

Note 5.— The ADS-air-user is considered an active user from the time at which it receives the first ADS-demand-contract indication, an ADS-event-contract indication or an ADS-periodic-contract indication until such time that:

- a) the ADS-air-user receives an ADS-cancel-all-contracts indication,*
- b) the ADS-air-user receives an ADS-cancel indication, and there are no other contracts in place,*
- c) the ADS-air-user invokes an ADS-cancel-emergency-contract request, and there are no other contracts in place,*
- d) the ADS-air-user invokes an ADS-demand-contract response, an ADS-event-contract response or an ADS-periodic-contract response, with the Reply parameter value set to “negative acknowledgement”, and there are no other contracts in place,*

- e) *the ADS-air-user invokes an ADS-report request, with the Contract type parameter value set to “demand contract”, and there are no other contracts in place,*
- f) *the ADS-air-user receives an ADS-user-abort indication,*
- g) *the ADS-air-user receives an ADS-provider-abort indication, or*
- h) *the ADS-air-user invokes an ADS-user-abort request.*

2.2.1.5.3.2 In 2.2.1.5.3, if no actions are described for an ADS service primitive in a particular state, then the invocation of that service primitive shall be prohibited in that state.

2.2.1.5.3.3 Possible errors arising upon Receipt of an APDU or a Dialogue Service Primitive.

2.2.1.5.3.3.1 If an APDU is not received when one is required, or one is received in an inappropriate dialogue service primitive, then exception handling procedures as described in 2.2.1.5.4.3 shall apply.

2.2.1.5.3.3.2 Upon receipt of an APDU or dialogue service primitive, if no actions are described for their arrival when in a particular state, then exception handling procedures as described in 2.2.1.5.4.4 shall apply.

2.2.1.5.3.3.3 Upon receipt of an APDU that cannot be decoded, then exception handling procedures as described in 2.2.1.5.4.7 shall apply.

2.2.1.5.3.4 Ground ADS HI Module

2.2.1.5.3.4.1 Upon receipt of a service primitive, the HI module shall pass it to the module as shown in Table 2.2.1.5-2.

Table 2.2.1.5-2: Request and response primitive to ground module mapping

Service Primitive	ADS-ground-ASE Module
ADS-demand-contract request	DC
ADS-event-contract request	EC
ADS-periodic-contract request	PC
ADS-cancel request with contract type event-contract	EC
ADS-cancel request with contract type periodic-contract	PC
ADS-cancel-all-contracts request	LI
ADS-modify-emergency-contract request	EM
ADS-user-abort request	AB

2.2.1.5.3.4.2 Upon receipt of a request to invoke a service primitive from one of the ground modules in the ADS-ground-ASE as shown in Table 2.2.1.5-3, the ground HI module shall do so.

Table 2.2.1.5-3: Indication and confirmation primitive to ground module mapping

ADS-ground-ASE Module	Service Primitive
DC	ADS-demand-contract confirmation
EC	ADS-event-contract confirmation
PC	ADS-periodic-contract confirmation
DC	ADS-report indication
EC	ADS-report indication
PC	ADS-report indication
EC	ADS-cancel confirmation
PC	ADS-cancel confirmation
LI	ADS-cancel-all-contracts confirmation
EM	ADS-emergency-report indication
EM	ADS-cancel-emergency indication
EM	ADS-modify-emergency-contract confirmation

2.2.1.5.3.4.3 On receipt of a request to invoke ADS-provider-abort indication from the ground AB module, the ground HI module shall:

- a) if the ADS-ground-user is not an active user, take no further action;
- b) if the ADS-ground-user is an active user, invoke ADS-provider-abort indication.

2.2.1.5.3.4.4 On receipt of a request to invoke ADS-user-abort indication from the ground AB module, the ground HI module shall:

- a) if the ADS-ground-user is not an active user, take no further action;
- b) if the ADS-ground-user is an active user, invoke ADS-user-abort indication.

2.2.1.5.3.4.5 The ground HI module shall reject requests and responses, apart from ADS-user-abort requests, when the ground LI module is in the LI-G-START state or the LI-G-END state.

2.2.1.5.3.5 Air ADS HI Module

2.2.1.5.3.5.1 Upon receipt of a service primitive, the air HI module shall pass it to the air module as shown in Table 2.2.1.5-4.

Table 2.2.1.5-4: Request and response primitive to air module mapping

Service Primitive	ADS-air-ASE Module
ADS-demand-contract response	DC
ADS-event-contract response	EC
ADS-periodic-contract response	PC
ADS-report request with <i>contract type</i> demand-contract	DC
ADS-report request with <i>contract type</i> event-contract	EC
ADS-report request with <i>contract type</i> periodic-contract	PC
ADS-emergency-report request	EM
ADS-cancel-emergency request	EM
ADS-modify-emergency-contract response	EM
ADS-user-abort request	AB

2.2.1.5.3.5.2 Upon receipt of a request to invoke a service primitive from one of the air modules in the ADS-air-ASE as shown in Table 2.2.1.5-5, the air HI module shall do so.

Table 2.2.1.5-5: Indication and confirmation primitive to air module mapping

ADS-air-ASE Module	Service Primitive
DC	ADS-demand-contract indication
EC	ADS-event-contract indication
PC	ADS-periodic-contract indication
EC	ADS-cancel indication
PC	ADS-cancel indication
LI	ADS-cancel-all-contracts indication
EM	ADS-modify-emergency-contract indication

2.2.1.5.3.5.3 On receipt of a request to invoke ADS-provider-abort indication from the air AB module, the air HI module shall:

- a) if the ADS-air-user is not an active user, take no further action;
- b) if the ADS-air-user is an active user, invoke ADS-provider-abort indication.

2.2.1.5.3.5.4 On receipt of a request to invoke ADS-user-abort indication from the air AB module, the air HI module shall:

- a) if the ADS-air-user is not an active user, take no further action;
- b) if the ADS-air-user is an active user, invoke ADS-user-abort indication.

2.2.1.5.3.6 Ground ADS DC Module

Note.— The states defined for the ground ADS DC module are the following:

- a) *DC-G-IDLE*
- b) *DC-G-PENDING*
- c) *DC-G-ACTIVE*

2.2.1.5.3.6.1 On initiation, the ground DC module shall be in the DC-G-IDLE state.

2.2.1.5.3.6.2 Upon receipt of an ADS-demand-contract request:

2.2.1.5.3.6.2.1 If in the DC-G-IDLE state, the ground DC module shall:

- a) create an ADS-demand-contract-PDU with elements derived as in Table 2.2.1.5-6,
- b) pass it, together with the *aircraft address* parameter value, *ICAO facility designation* parameter value and the *class of communication service* parameter value, to the ground LI module,
- c) start timer t-DC-1, and
- d) enter the DC-G-PENDING state.

Table 2.2.1.5-6

PDU Element Name	Derivation of Element Value
ADS-demand-contract-PDU	<i>contract details</i> parameter

2.2.1.5.3.6.3 Upon receipt of an ADS-demand-report-PDU containing a *positive-acknowledgement* element:

2.2.1.5.3.6.3.1 If in the DC-G-PENDING state, the ground DC module shall:

- a) stop the t-DC-1 timer,
- b) request the ground HI module to invoke ADS-report indication with parameter values derived as in Table 2.2.1.5-7,
- c) enter the DC-G-IDLE state.

Table 2.2.1.5-7

Parameter Name	Derivation of Parameter Value
Contract Type	“Demand contract”
Event Type	not supplied
Positive Acknowledgement	NULL
Report Details	<i>report</i> element of the ADS-demand-report-PDU

2.2.1.5.3.6.4 Upon receipt of an ADS-demand-report-PDU not containing a *positive-acknowledgement* element:

2.2.1.5.3.6.4.1 If in the DC-G-ACTIVE state, the ground DC module shall:

- a) stop the t-DC-2 timer,
- b) request the ground HI module to invoke ADS-report indication with parameter values derived as in Table 2.2.1.5-8, and
- c) enter the DC-G-IDLE state.

Table 2.2.1.5-8

Parameter Name	Derivation of Parameter Value
Contract Type	“Demand contract”
Event Type	not supplied
Positive Acknowledgement	not supplied
Report Details	<i>report</i> element of the ADS-demand-report-PDU

2.2.1.5.3.6.5 Upon receipt of an ADS-negative-acknowledgement-PDU:

2.2.1.5.3.6.5.1 If in the DC-G-PENDING state, the ground DC module shall:

- a) stop the t-DC-1 timer,
- b) request the ground HI module to invoke ADS-demand-contract confirmation with parameter values derived as in Table 2.2.1.5-9, and
- c) enter the DC-G-IDLE state.

Table 2.2.1.5-9

Parameter Name	Derivation of Parameter Value
Reply	<i>NegativeAcknowledgement</i> with a value derived from the <i>reason</i> element of the ADS-negative-acknowledgement-PDU

2.2.1.5.3.6.6 Upon receipt of an ADS-noncompliance-notification-PDU:

2.2.1.5.3.6.6.1 If in the DC-G-PENDING state, the ground DC module shall:

- a) stop the t-DC-1 timer,
- b) request the ground HI module to invoke an ADS-demand-contract confirmation with parameter values derived as in Table 2.2.1.5-10,
- c) start the t-DC-2 timer, and
- d) enter the DC-G-ACTIVE state.

Table 2.2.1.5-10

Parameter Name	Derivation of Parameter Value
Reply	<i>NoncomplianceNotification</i> with a value derived from the <i>demand-ncn</i> element of the ADS-noncompliance-notification-PDU

2.2.1.5.3.6.6.2 Upon expiry of the t-DC-1 timer or the t-DC-2 timer, the ground DC module shall:

- a) request the ground AB module to abort with reason *timer-expiry*, and
- b) enter the DC-G-IDLE state

2.2.1.5.3.6.6.3 Upon receipt of a request from the ground AB or ground LI module to stop operation, the ground DC module shall:

- a) stop any timers, and
- b) enter the DC-G-IDLE state.

2.2.1.5.3.7 Air ADS DC Module

Note.— The states defined for the air ADS DC module are the following:

- a) *DC-A-IDLE*
- b) *DC-A-PENDING*
- c) *DC-A-ACTIVE*

2.2.1.5.3.7.1 On initiation, the air DC module shall be in the DC-A-IDLE state.

2.2.1.5.3.7.2 Upon receipt of an ADS-demand-contract response with the *Reply* parameter value set to *negative acknowledgement*:

2.2.1.5.3.7.2.1 If in the DC-A-PENDING state, the air DC module shall:

- a) create an ADS-negative-acknowledgement-PDU with elements as defined in Table 2.2.1.5-11,
- b) pass it to the air LI module, and
- c) enter the DC-A-IDLE state.

Table 2.2.1.5-11

PDU Element Name	Derivation of Element Value
request-type	demand-contract
reason	<i>Reply</i> parameter value

2.2.1.5.3.7.3 Upon receipt of an ADS-demand-contract response with the *reply* parameter value set to *noncompliance notification*:

2.2.1.5.3.7.3.1 If in the DC-A-PENDING state, the air DC module shall:

- a) create an ADS-noncompliance-notification-PDU with elements as defined in Table 2.2.1.5-12,
- b) pass it to the air LI module, and

- c) enter the DC-A-ACTIVE state.

Table 2.2.1.5-12

PDU Element Name	Derivation of Element Value
NoncomplianceNotification	demand-ncn with a value derived from <i>Reply</i> parameter value

2.2.1.5.3.7.4 Upon receipt of an ADS-report request:

2.2.1.5.3.7.4.1 If in the DC-A-PENDING state and the *positive acknowledgement* parameter is present, the air DC module shall:

- a) create ADS-demand-report-PDU with a value derived as in Table 2.2.1.5-13,
- b) pass it to the air LI module, and
- c) enter the DC-A-IDLE state.

2.2.1.5.3.7.4.2 If in the DC-A-ACTIVE state and the *positive acknowledgement* parameter is not present, the air DC module shall:

- a) create ADS-demand-report-PDU with a value derived as in Table 2.2.1.5-14,
- b) pass it to the air LI module, and
- c) enter the DC-A-IDLE state.

Table 2.2.1.5-13

PDU Element Name	Derivation of Element Value
Report	<i>Report details</i> parameter value
Positive acknowledgement	NULL

Table 2.2.1.5-14

PDU Element Name	Derivation of Element Value
Report	<i>Report details</i> parameter value
Positive acknowledgement	not supplied

2.2.1.5.3.7.5 Upon receipt of an ADS-demand-contract-PDU:

2.2.1.5.3.7.5.1 If in the DC-A-IDLE state, the air DC module shall:

- a) request the air HI module to invoke ADS-demand-contract indication with parameters derived as in Table 2.2.1.5-15, and,
- b) enter the DC-A-PENDING state.

Table 2.2.1.5-15

PDU Element Name	Derivation of Element Value
Contract details	ADS-demand-contract-PDU
ICAO facility designation	Calling peer id, if provided by the air LI module

2.2.1.5.3.7.5.2 Upon receipt of a request from the air AB or air LI module to stop operation, the air DC module shall enter the DC-A-IDLE state.

2.2.1.5.3.8 Ground ADS EC Module

Note.— The states defined for the ground ADS EC module are the following:

- a) *EC-G-IDLE*
- b) *EC-G-START-PENDING*
- c) *EC-G-ACTIVE*
- d) *EC-G-PENDING*
- e) *EC-G-CANCEL*

2.2.1.5.3.8.1 On initiation, the ground EC module shall be in the EC-G-IDLE state.

2.2.1.5.3.8.2 Upon receipt of an ADS-event-contract request:

2.2.1.5.3.8.2.1 If in the EC-G-IDLE state, the ground EC module shall:

- a) create an ADS-event-contract-PDU with elements as defined in Table 2.2.1.5-16,
- b) pass it, together with the *aircraft address* parameter value, *ICAO facility designation* parameter value and the *class of communication service* parameter value to the ground LI module,
- c) start timer t-EC-1, and
- d) enter the EC-G-START-PENDING state.

2.2.1.5.3.8.2.2 If in the EC-G-ACTIVE state, the ground EC module shall:

- a) create an ADS-event-contract-PDU with elements as defined in Table 2.2.1.5-17,
- b) pass it to the ground LI module,
- c) start timer t-EC-1, and
- d) enter the EC-G-PENDING state.

Table 2.2.1.5-16

PDU Element Name	Derivation of Element Value
ADS-event-contract-PDU	<i>contract details</i> parameter value

Table 2.2.1.5-17

PDU Element Name	Derivation of Element Value
ADS-event-contract-PDU	<i>contract details</i> parameter value

2.2.1.5.3.8.3 Upon receipt of an ADS-cancel request:

2.2.1.5.3.8.3.1 If in the EC-G-ACTIVE state, the ground EC module shall:

- a) create an ADS-cancel-contract-PDU with elements as defined in Table 2.2.1.5-18,
- b) pass it to the ground LI module,
- c) start timer t-EC-2, and
- d) enter the EC-G-CANCEL state.

Table 2.2.1.5-18

PDU Element Name	Derivation of Element Value
CancelContract	event-contract

2.2.1.5.3.8.4 Upon receipt of an ADS-event-report-PDU containing a *positive acknowledgement*:

2.2.1.5.3.8.4.1 If in the EC-G-PENDING or the EC-G-START-PENDING state, the ground EC module shall:

- a) stop the t-EC-1 timer,

- b) request the ground HI module to invoke ADS-report indication, with parameter values derived as in Table 2.2.1.5-19, and
- c) enter the EC-G-ACTIVE state.

Table 2.2.1.5-19

Parameter Name	Derivation of Parameter Value
Contract type	“event contract”
Event type	event-type PDU element
Positive acknowledgement	NULL
Report details	report PDU element

2.2.1.5.3.8.5 Upon receipt of an ADS-event-report-PDU not containing a *positive acknowledgement*:

2.2.1.5.3.8.5.1 If in the EC-G-ACTIVE, EC-G-PENDING or EC-G-CANCEL state, the ground EC module shall:

- a) request the ground HI module to invoke ADS-report indication, with parameter values derived as in Table 2.2.1.5-20, and
- b) remain in the same state.

Table 2.2.1.5-20

Parameter Name	Derivation of Parameter Value
Contract type	“event contract”
Event type	event-type PDU element
Positive acknowledgement	not supplied
Report details	report PDU element

2.2.1.5.3.8.6 Upon receipt of an ADS-positive-acknowledgement-PDU for an event contract:

2.2.1.5.3.8.6.1 If in the EC-G-START-PENDING state or the EC-G-PENDING state, the ground EC module shall:

- a) stop the t-EC-1 timer,

- b) request the ground HI module to invoke ADS-event-contract confirmation, with parameter values derived as in Table 2.2.1.5-21, and
- c) enter the EC-G-ACTIVE state.

Table 2.2.1.5-21

Parameter Name	Derivation of Parameter Value
Reply	<i>PositiveAcknowledgement</i> with abstract value NULL

2.2.1.5.3.8.7 Upon receipt of an ADS-positive-acknowledgement-PDU for a cancel contract:

2.2.1.5.3.8.7.1 If in the EC-G-CANCEL state, the ground EC module shall:

- a) stop the t-EC-2 timer,
- b) request the ground HI module to invoke ADS-cancel-contract confirmation, with parameter values derived as in Table 2.2.1.5-22, and
- c) enter the EC-G-IDLE state.

Table 2.2.1.5-22

Parameter Name	Derivation of Parameter Value
Contract type	event-contract

2.2.1.5.3.8.8 Upon receipt of an ADS-negative-acknowledgement-PDU for an event contract:

2.2.1.5.3.8.8.1 If in the EC-G-START-PENDING state, the ground EC module shall:

- a) stop the t-EC-1 timer,
- b) request the ground HI module to invoke ADS-event-contract confirmation, with parameter values derived as in Table 2.2.1.5-23, and
- c) enter the EC-G-IDLE state.

2.2.1.5.3.8.8.2 If in the EC-G-PENDING state, the ground EC module shall:

- a) stop the t-EC-1 timer,
- b) request the ground HI module to invoke ADS-event-contract confirmation, with parameter values derived as in Table 2.2.1.5-23, and
- c) enter the EC-G-ACTIVE state.

Table 2.2.1.5-23

Parameter Name	Derivation of Parameter Value
Reply	<i>NegativeAcknowledgement</i> with value derived from reason PDU element

2.2.1.5.3.8.9 Upon receipt of an ADS-noncompliance-notification-PDU for an event contract:

2.2.1.5.3.8.9.1 If in the EC-G-START-PENDING state or the EC-G-PENDING, the ground EC module shall:

- a) stop the t-EC-1 timer,
- b) request the ground HI module to invoke ADS-event-contract confirmation, with parameter values derived as in Table 2.2.1.5-24, and
- c) enter the EC-G-ACTIVE state.

Table 2.2.1.5-24

Parameter Name	Derivation of Parameter Value
Reply	<i>NoncomplianceNotification</i> with value derived from event-ncn PDU element

2.2.1.5.3.8.9.2 Upon expiry of the t-EC-1 timer or the t-EC-2 timer, the ground EC module shall:

- a) request the ground AB module to abort with reason *timer-expiry*, and
- b) enter the EC-G-IDLE state

2.2.1.5.3.8.9.3 Upon receipt of a request from the ground AB module to stop operation, the EC module shall:

- a) stop any timers, and
- b) enter the EC-G-IDLE state.

2.2.1.5.3.9 Air ADS EC Module

Note.— The states defined for the air ADS EC module are the following:

- a) *EC-A-IDLE*
- b) *EC-A-PENDING*

- c) *EC-A-ACTIVE*
- d) *EC-A-ACTIVE-PENDING*

2.2.1.5.3.9.1 On initiation, the air EC module shall be in the EC-A-IDLE state.

2.2.1.5.3.9.2 Upon receipt of an ADS-event-contract response with the *reply* parameter value set to *positive acknowledgement*:

2.2.1.5.3.9.2.1 If in the EC-A-PENDING state or in the EC-A-ACTIVE-PENDING state, the air EC module shall:

- a) create an ADS-positive-acknowledgement-PDU with elements as defined in Table 2.2.1.5-25
- b) pass it to the air LI module, and
- c) enter the EC-A-ACTIVE state.

Table 2.2.1.5-25

PDU Element Name	Derivation of Element Value
Request type	event-contract

2.2.1.5.3.9.3 Upon receipt of an ADS-event-contract response with the *reply* parameter value set to *noncompliance notification*:

2.2.1.5.3.9.3.1 If in the EC-A-PENDING state or in the EC-A-ACTIVE-PENDING state, the air EC module shall:

- a) create an ADS-noncompliance-notification-PDU with elements as defined in Table 2.2.1.5-26,
- b) pass it to the air LI module, and
- c) enter the EC-A-ACTIVE state.

Table 2.2.1.5-26

PDU Element Name	Derivation of Element Value
NoncomplianceNotification	<i>NoncomplianceNotification</i> with value derived from <i>Reply</i> parameter

2.2.1.5.3.9.4 Upon receipt of an ADS-event-contract response with the *reply* parameter value set to *negative acknowledgement*:

2.2.1.5.3.9.4.1 If in the EC-A-PENDING state, the air EC module shall:

- a) create an ADS-negative-acknowledgement-PDU with elements as defined in Table 2.2.1.5-27,
- b) pass it to the air LI module, and
- c) enter the EC-A-IDLE state.

2.2.1.5.3.9.4.2 If in the EC-A-ACTIVE-PENDING state, the air EC module shall:

- a) create an ADS-negative-acknowledgement-PDU with elements as defined in Table 2.2.1.5-27,
- b) pass it to the air LI module, and
- c) enter the EC-A-ACTIVE state.

Table 2.2.1.5-27

PDU Element Name	Derivation of Element Value
Request Type	event-contract
Reason	<i>Reply</i> parameter value

2.2.1.5.3.9.5 Upon receipt of an ADS-report request with a *positive acknowledgement* parameter:

2.2.1.5.3.9.5.1 If in the EC-A-PENDING state or the EC-A-ACTIVE-PENDING state, the air EC module shall:

- a) create an ADS-event-report-PDU with element as defined in Table 2.2.1.5-28,
- b) pass it to the air LI module, and
- c) enter the EC-A-ACTIVE state.

Table 2.2.1.5-28

PDU Element Name	Derivation of Element Value
Event type	<i>Event type</i> parameter value
Report	<i>Report details</i> parameter value
Positive Acknowledgement	NULL

2.2.1.5.3.9.6 Upon receipt of an ADS-report request with no *positive acknowledgement* parameter:

2.2.1.5.3.9.6.1 If in the EC-A-ACTIVE state, the air EC module shall:

- a) create an ADS-event-report-PDU with element as defined in Table 2.2.1.5-29,
- b) pass it to the air LI module, and
- c) remain in the EC-A-ACTIVE state.

Table 2.2.1.5-29

PDU Element Name	Derivation of Element Value
Event type	<i>Event type</i> parameter value
Report	<i>Report details</i> parameter value
Positive Acknowledgement	not supplied

2.2.1.5.3.9.7 Upon receipt of an ADS-event-contract-PDU:

2.2.1.5.3.9.7.1 If in the EC-A-IDLE state, the air EC module shall:

- a) request the air HI module to invoke ADS-event-contract indication with parameter values derived as in Table 2.2.1.5-30, and
- b) enter the EC-A-PENDING state.

2.2.1.5.3.9.7.2 If in the EC-A-ACTIVE state, the air EC module shall:

- a) request the air HI module to invoke ADS-event-contract indication with parameter values derived as in Table 2.2.1.5-30, and
- b) enter the EC-A-ACTIVE-PENDING state.

Table 2.2.1.5-30

Parameter Name	Derivation of Parameter Value
Contract details	ADS-event-contract-PDU
ICAO facility designation	Calling peer id, if provided by the air LI module

2.2.1.5.3.9.8 Upon receipt of an ADS-cancel-contract-PDU:

2.2.1.5.3.9.8.1 If in the EC-A-ACTIVE state, the air EC module shall:

- a) request the HI module to invoke ADS-cancel indication (event-contract) with parameter values as defined in Table 2.2.1.5-31,
- b) create an ADS-positive-acknowledgement-PDU (cancel-event-contract) with elements as defined in Table 2.2.1.5-32,
- c) pass it to the air LI module, and
- d) enter the EC-A-IDLE state.

Table 2.2.1.5-31

Parameter Name	Derivation of Parameter Value
Contract type	event-contract

Table 2.2.1.5-32

PDU Element Name	Derivation of Element Value
Request type	cancel-event-contract

2.2.1.5.3.9.8.2 Upon receipt of a request from the air AB or air LI module to stop operation, the air EC module shall enter the EC-A-IDLE state.

2.2.1.5.3.10 Ground ADS PC Module

Note.— The states defined for the ground ADS PC module are the following:

- a) *PC-G-IDLE*
- b) *PC-G-START-PENDING*
- c) *PC-G-ACTIVE*
- d) *PC-G-PENDING*

e) *PC-G-CANCEL*

2.2.1.5.3.10.1 On initiation, the ground PC module shall be in the PC-G-IDLE state.

Note.— The ground PC module has a boolean variable named EMERGENCY.

2.2.1.5.3.10.2 On initiation, EMERGENCY shall be set to FALSE.

2.2.1.5.3.10.3 Upon receipt of an ADS-periodic-contract request:

2.2.1.5.3.10.3.1 If in the PC-G-IDLE state, the ground PC module shall:

- a) create an ADS-periodic-contract-PDU with elements as defined in Table 2.2.1.5-33,
- b) pass it, together with the *aircraft address* parameter value, *ICAO facility designation* parameter value and the *Class of communication service* parameter value, to the ground LI module,
- c) start timer t-PC-1, and
- d) enter the PC-G-START-PENDING state.

2.2.1.5.3.10.3.2 If in the PC-G-ACTIVE state, the PC module shall:

- a) if EMERGENCY = FALSE, stop the t-PC-2 timer,
- b) create an ADS-periodic-contract-PDU with elements as defined in Table 2.2.1.5-33,
- c) pass it to the ground LI module,
- d) start timer t-PC-1, and
- e) enter the PC-G-PENDING state.

Table 2.2.1.5-33

PDU Element Name	Derivation of Element Value
ADS-periodic-contract	<i>Contract details</i> parameter value

2.2.1.5.3.10.4 Upon receipt of an ADS-cancel request:

2.2.1.5.3.10.4.1 If in the PC-G-ACTIVE state, the ground PC module shall:

- a) stop the t-PC-2 timer,
- b) create an ADS-cancel-contract-PDU with elements as defined in Table 2.2.1.5-34,

- c) pass it to the ground LI module,
- d) start timer t-PC-3, and
- e) enter the PC-G-CANCEL state.

Table 2.2.1.5-34

PDU Element Name	Derivation of Element Value
ADS-cancel-contract-PDU	periodic-contract

2.2.1.5.3.10.5 Upon receipt of an ADS-periodic-report-PDU containing a *positive acknowledgement*:

2.2.1.5.3.10.5.1 If in the PC-G-START-PENDING state, or the PC-G-PENDING state, the ground PC module shall:

- a) stop the t-PC-1 timer,
- b) create the *report details* parameter of an ADS-report indication,
- c) request the ground HI module to invoke ADS-report indication with parameter values derived as in Table 2.2.1.5-35,
- d) if EMERGENCY = FALSE, start the t-PC-2 timer, and
- e) enter the PC-G-ACTIVE state.

Table 2.2.1.5-35

Parameter Name	Derivation of Parameter Value
Contract type	periodic-contract
Event type	not supplied
Positive acknowledgement	NULL
Report details	report PDU element

2.2.1.5.3.10.6 Upon receipt of an ADS-periodic-report-PDU not containing a *positive acknowledgement*:

2.2.1.5.3.10.6.1 If in the PC-G-PENDING state, the ground PC module shall:

- a) request the ground HI module to invoke ADS-report indication with parameter values derived as in Table 2.2.1.5-36,
- b) remain in the PC-G-PENDING state.

2.2.1.5.3.10.6.2 If in the PC-G-ACTIVE state, the ground PC module shall:

- a) if EMERGENCY = FALSE, stop the t-PC-2 timer,
- b) request the ground HI module to invoke ADS-report indication with parameter values derived as in Table 2.2.1.5-36,
- c) if EMERGENCY = FALSE, start the t-PC-2 timer, and
- d) remain in the PC-G-ACTIVE state.

2.2.1.5.3.10.6.3 If in the PC-G-CANCEL state, the ground PC module shall:

- a) request the ground HI module to invoke ADS-report indication with parameter values derived as in Table 2.2.1.5-36, and
- b) remain in the PC-G-CANCEL state.

Table 2.2.1.5-36

Parameter Name	Derivation of Parameter Value
Contract type	periodic-contract
Event type	not supplied
Positive acknowledgement	not supplied
Report details	report PDU element

2.2.1.5.3.10.7 Upon receipt of an ADS-positive-acknowledgement-PDU for a periodic contract:

2.2.1.5.3.10.7.1 If in the PC-G-START-PENDING state, or the PC-G-PENDING state, the ground PC module shall:

- a) stop the t-PC-1 timer,
- b) request the ground HI module to invoke ADS-periodic-contract confirmation with parameter values derived as in Table 2.2.1.5-37,
- c) if EMERGENCY = FALSE, start the t-PC-2 timer, and
- d) enter the PC-G-ACTIVE state.

Table 2.2.1.5-37

Parameter Name	Derivation of Parameter Value
Reply	<i>PositiveAcknowledgement</i> with value NULL

2.2.1.5.3.10.8 Upon receipt of an ADS-positive-acknowledgement-PDU for an cancel contract:

2.2.1.5.3.10.8.1 If in the PC-G-CANCEL state, the ground PC module shall:

- a) stop the t-PC-3 timer,
- b) request the ground HI module to invoke ADS-cancel-contract confirmation with parameter values derived as in Table 2.2.1.5-38, and
- c) enter the PC-G-IDLE state.

Table 2.2.1.5-38

Parameter Name	Derivation of Parameter Value
Contract type	periodic-contract

2.2.1.5.3.10.9 Upon receipt of an ADS-negative-acknowledgement-PDU for a periodic contract:

2.2.1.5.3.10.9.1 If in the PC-G-START-PENDING state, the ground PC module shall:

- a) stop the t-PC-1 timer,
- b) request the ground HI module to invoke ADS-periodic-contract confirmation with parameter values derived as in Table 2.2.1.5-39, and
- c) enter the PC-G-IDLE state.

2.2.1.5.3.10.9.2 If in the PC-G-PENDING state, the ground PC module shall:

- a) stop the t-PC-1 timer,
- b) request the ground HI module to invoke ADS-periodic-contract confirmation with parameter values derived as in Table 2.2.1.5-39, and
- c) if EMERGENCY = FALSE, start the t-PC-2 timer, and
- d) enter the PC-G-ACTIVE state.

Table 2.2.1.5-39

Parameter Name	Derivation of Parameter Value
Reply	<i>NegativeAcknowledgement</i> with value derived from the reason PDU element

2.2.1.5.3.10.10 Upon receipt of an ADS-noncompliance-notification-PDU for a periodic contract:

2.2.1.5.3.10.10.1 If in the PC-G-START-PENDING state, or the PC-G-PENDING state, the ground PC module shall:

- a) stop the t-PC-1 timer,
- b) request the ground HI module to invoke ADS-periodic-contract confirmation with parameter values derived as in Table 2.2.15-40, and
- c) if EMERGENCY = FALSE, start the t-PC-2 timer, and
- d) enter the PC-G-ACTIVE state.

Table 2.2.1.5-40

Parameter Name	Derivation of Parameter Value
Reply	<i>NoncomplianceNotification</i> with value derived from the periodic-ncn PDU element

2.2.1.5.3.10.11 Upon receipt of a request to suspend periodic contracts from the ground EM module, the PC module shall:

- a) if in the PC-G-ACTIVE state, stop the t-PC-2 timer,
- b) set EMERGENCY to be TRUE, and
- c) remain in the same state.

2.2.1.5.3.10.12 Upon receipt of a request to reinstate periodic contracts from the ground EM module, the ground PC module shall:

- a) if in the PC-G-ACTIVE state, start the t-PC-2 timer, based on the most recent value of the period of the contract,
- b) set EMERGENCY to be FALSE, and
- c) remain in the same state.

2.2.1.5.3.10.13 Upon receipt of a request from the ground AB or ground LI module to stop operation, the ground PC module shall:

- a) stop any timers,
- b) set EMERGENCY to be FALSE, and
- c) enter the PC-G-IDLE state.

2.2.1.5.3.10.14 Upon expiry of the t-PC-1 timer, t-PC-2 timer, or t-PC-3 timer, the ground PC module shall:

- a) request the ground AB module to abort with reason timer-expiry, and
- b) enter the PC-G-IDLE state.

2.2.1.5.3.11 Air ADS PC Module

Note.— The states defined for the air ADS PC module are the following:

- a) *PC-A-IDLE*
- b) *PC-A-PENDING*
- c) *PC-A-ACTIVE*
- d) *PC-A-ACTIVE-PENDING*

2.2.1.5.3.11.1 On initiation, the air PC module shall be in the PC-A-IDLE state.

2.2.1.5.3.11.2 Upon receipt of an ADS-periodic-contract response with the *reply* parameter value set to *positive acknowledgement* or *noncompliance notification*, then:

2.2.1.5.3.11.2.1 If in the PC-A-PENDING state or PC-A-ACTIVE-PENDING state, the air PC module shall:

- a) If the *reply* parameter value is a *positive acknowledgement*, create an ADS-positive-acknowledgement-PDU with elements as defined in Table 2.2.1.5-41, or
- b) If the *reply* parameter value is a *noncompliance notification*, create an ADS-noncompliance-notification-PDU with elements as defined in Table 2.2.1.5-42,
- c) pass it to the air LI module, and
- d) enter the PC-A-ACTIVE state.

Table 2.2.1.5-41

PDU Element Name	Derivation of Element Value
ADS-positive-acknowledgement-PDU	periodic-contract

Table 2.2.1.5-42

PDU Element Name	Derivation of Element Value
NoncomplianceNotification	<i>Reply</i> parameter value

2.2.1.5.3.11.3 Upon receipt of an ADS-periodic-contract response with the *reply* parameter value set to *negative acknowledgement*, then:

2.2.1.5.3.11.3.1 If in the PC-A-PENDING state, the air PC module shall:

- a) create an ADS-negative-acknowledgement-PDU with elements as defined in Table 2.2.1.5-43,
- b) pass it to the air LI module, and
- c) enter the PC-A-IDLE state.

2.2.1.5.3.11.3.2 If in the PC-A-ACTIVE-PENDING state, the air PC module shall:

- a) create an ADS-negative-acknowledgement-PDU with elements as defined in Table 2.2.1.5-43
- b) pass it to the air LI module, and
- c) enter the PC-A-ACTIVE state.

Table 2.2.1.5-43

PDU Element Name	Derivation of Element Value
request-type	periodic-contract
reason	<i>Reply</i> parameter value

2.2.1.5.3.11.4 Upon receipt of an ADS-report request with a *positive acknowledgement* parameter, then:

2.2.1.5.3.11.4.1 If in the PC-A-PENDING state or PC-A-ACTIVE-PENDING, the air PC module shall:

- a) create ADS-periodic-report-PDU with elements as defined in Table 2.2.1.5-44,
- b) pass it to the air LI module, and

- c) enter the PC-A-ACTIVE state.

Table 2.2.1.5-44

PDU Element Name	Derivation of Element Value
report	<i>Report details</i> parameter value
positive-acknowledgement	NULL

2.2.1.5.3.11.5 Upon receipt of an ADS-report request with no *positive acknowledgement* parameter, then:

2.2.1.5.3.11.5.1 If in the PC-A-ACTIVE state, the air PC module shall:

- a) create ADS-periodic-report-PDU with elements as defined in Table 2.2.1.5-45,
- b) pass it to the air LI module, and
- c) remain in the PC-A-ACTIVE state.

Table 2.2.1.5-45

PDU Element Name	Derivation of Element Value
report	<i>Report details</i> parameter value
positive-acknowledgement	not present

2.2.1.5.3.11.6 Upon receipt of an ADS-periodic-contract-PDU:

2.2.1.5.3.11.6.1 If in the PC-A-IDLE state, the air PC module shall:

- a) request the air HI module to invoke ADS-periodic-contract indication with parameter values derived as in Table 2.2.1.5-46, and
- b) enter the PC-A-PENDING state.

2.2.1.5.3.11.6.2 If in the PC-A-ACTIVE state, the air PC module shall:

- a) request the air HI module to invoke ADS-periodic-contract indication with parameter values derived as in Table 2.2.1.5-46, and
- b) enter the PC-A-ACTIVE-PENDING state.

Table 2.2.1.5-46

Parameter Name	Derivation of Parameter Value
Contract details	ADS-periodic-contract-PDU
ICAO facility designation	Calling peer id, if provided by the air LI module

2.2.1.5.3.11.7 Upon receipt of an ADS-cancel-contract-PDU, then:

2.2.1.5.3.11.7.1 If in the PC-A-ACTIVE state, the air PC module shall:

- a) request the air HI module to invoke ADS-cancel indication with parameter values derived as in Table 2.2.1.5-47,
- b) create an ADS-positive-acknowledgement-PDU with elements as defined in Table 2.2.1.5-48,
- c) pass it to the air LI module, and
- d) enter the PC-A-IDLE state.

Table 2.2.1.5-47

Parameter Name	Derivation of Parameter Value
Contract type	periodic-contract

Table 2.2.1.5-48

PDU Element Name	Derivation of Element Value
RequestType	cancel-periodic-contract

2.2.1.5.3.11.7.2 Upon receipt of a request from the air AB or air LI module to stop operation, the air PC module shall enter the PC-A-IDLE state.

2.2.1.5.3.12 Ground ADS EM Module

Note.— The states defined for the ground ADS EM module are the following:

- a) *EM-G-IDLE*
- b) *EM-G-ACTIVE*
- c) *EM-G-MODIFY*

2.2.1.5.3.12.1 On initiation, the ground EM module shall be in the EM-G-IDLE state.

2.2.1.5.3.12.2 Upon receipt of an ADS-modify-emergency-contract request:

2.2.1.5.3.12.2.1 If in the EM-G-ACTIVE state, the ground EM module shall:

- a) stop the t-EM-1 timer,
- b) create an ADS-modify-emergency-contract-PDU with elements as defined in Table 2.2.1.5-49,
- c) pass it to the ground LI module,
- d) start the t-EM-2 timer, and
- e) enter the EM-G-MODIFY state.

Table 2.2.1.5-49

PDU Element Name	Derivation of Element Value
ModifyEmergency	<i>reporting interval</i> parameter value

2.2.1.5.3.12.3 Upon receipt of an ADS-emergency-report-PDU containing a positive acknowledgement:

2.2.1.5.3.12.3.1 If in the EM-G-MODIFY state, the ground EM module shall:

- a) stop the t-EM-2 timer,
- b) request the ground HI module to invoke invoke ADS-emergency-report indication with parameter values derived as in Table 2.2.1.5-50,
- c) start the t-EM-1 timer, and
- d) enter the EM-G-ACTIVE state.

Table 2.2.1.5-50

Parameter Name	Derivation of Parameter Value
Positive acknowledgement of modification	NULL
Emergency report details	emergency-report element of ADSEmergency

2.2.1.5.3.12.4 Upon receipt of an ADS-emergency-report-PDU not containing a positive acknowledgement:

2.2.1.5.3.12.4.1 If in the EM-G-IDLE state, the ground EM module shall:

- a) request the PC module to suspend operation,

- b) request the ground HI module to invoke ADS-emergency-report indication with parameter values derived as in Table 2.2.1.5-51, and
- c) start the t-EM-1 timer, and
- d) enter the EM-G-ACTIVE state.

2.2.1.5.3.12.4.2 If in the EM-G-ACTIVE state, the ground EM module shall:

- a) stop the t-EM-1 timer,
- b) request the ground HI module to invoke ADS-emergency-report indication with parameter values derived as in Table 2.2.1.5-51, and
- c) start the t-EM-1 timer, and
- d) enter the EM-G-ACTIVE state.

2.2.1.5.3.12.4.3 If in the EM-G-MODIFY state, the ground EM module shall:

- a) request the ground HI module to invoke ADS-emergency-report indication with parameter values derived as in Table 2.2.1.5-51, and
- b) remain in the EM-G-MODIFY state.

Table 2.2.1.5-51

Parameter Name	Derivation of Parameter Value
Positive acknowledgement of modification	not provided
Emergency report details	emergency-report element of ADSEmergency

2.2.1.5.3.12.5 Upon receipt of an ADS-cancel-emergency-PDU:

2.2.1.5.3.12.5.1 If in the EM-G-ACTIVE state, the ground EM module shall:

- a) stop the t-EM-1 timer,
- b) request the ground HI module to invoke ADS-cancel-emergency indication,
- c) create ADS-cancel-emergency-acknowledgement-PDU,
- d) pass it to the ground LI module,
- e) request the ground PC module to reinstate any periodic contracts, and
- f) enter the EM-G-IDLE state.

2.2.1.5.3.12.5.2 If in the EM-G-MODIFY state, the ground EM module shall:

- a) stop the t-EM-2 timer,
- b) request the ground HI module to invoke ADS-cancel-emergency indication,
- c) create ADS-cancel-emergency-acknowledgement-PDU,
- d) pass it to the ground LI module,
- e) request the ground PC module to reinstate any periodic contracts, and
- f) enter the EM-G-IDLE state.

2.2.1.5.3.12.6 Upon receipt of an ADS-negative-acknowledgement-PDU for a modify-emergency-contract:

2.2.1.5.3.12.6.1 If in the EM-G-MODIFY state, the ground EM module shall:

- a) stop the t-EM-2 timer,
- b) request the ground HI module to invoke ADS-modify-emergency-contract confirmation,
- c) start the t-EM-1 timer, and
- d) enter the EM-G-ACTIVE state.

2.2.1.5.3.12.7 Upon expiry of the t-EM-1 timer or the t-EM-2 timer, the ground EM module shall:

- a) request the ground AB module to abort with reason *timer-expiry*, and
- b) enter the EM-G-IDLE state.

2.2.1.5.3.12.7.1 Upon receipt of a request from the ground AB or ground LI module to stop operation, the ground EM module shall:

- a) stop any timers, and
- b) enter the EM-G-IDLE state.

2.2.1.5.3.13 Air ADS EM Module

Note.— The states defined for the air ADS EM module are the following:

- a) *EM-A-IDLE*
- b) *EM-A-ACTIVE*
- c) *EM-A-MODIFY*

d) *EM-A-CANCEL*

2.2.1.5.3.13.1 On initiation, the air EM module shall be in the EM-A-IDLE state.

2.2.1.5.3.13.2 Upon receipt of an ADS-emergency-report request with no *positive acknowledgement* parameter:

2.2.1.5.3.13.2.1 If in the EM-A-IDLE state, the air EM module shall:

- a) create an ADS-emergency-report-PDU with elements as defined in Table 2.2.1.5-52,
- b) pass it to the air LI module, and
- c) enter the EM-A-ACTIVE state.

2.2.1.5.3.13.2.2 If in the EM-A-ACTIVE state, the air EM module shall:

- a) create an ADS-emergency-report-PDU with elements as defined in Table 2.2.1.5-52,
- b) pass it to the air LI module, and
- c) remain in the EM-A-ACTIVE state.

Table 2.2.1.5-52

Parameter Name	Derivation of Parameter Value
emergency-report	<i>emergency report details</i> parameter value
positive-acknowledgements	not provided

2.2.1.5.3.13.3 Upon receipt of an ADS-emergency-report request with a *positive acknowledgement* parameter:

2.2.1.5.3.13.3.1 If in the EM-A-MODIFY state, the air EM module shall:

- a) create an ADS-emergency-report-PDU with elements as defined in Table 2.2.1.5-53,
- b) pass it to the air LI module, and
- c) enter the EM-A-ACTIVE state.

Table 2.2.1.5-53

Parameter Name	Derivation of Parameter Value
emergency-report	<i>emergency report details</i> parameter value
positive-acknowledgements	NULL

2.2.1.5.3.13.4 Upon receipt of an ADS-cancel-emergency request:

2.2.1.5.3.13.4.1 If in the EM-A-ACTIVE state, the air EM module shall:

- a) create an ADS-cancel-emergency-PDU,
- b) pass it to the air LI module,
- c) start the t-EM-3 timer, and
- d) enter the EM-A-CANCEL state.

2.2.1.5.3.13.5 Upon receipt of an ADS-modify-emergency-contract response:

2.2.1.5.3.13.5.1 If in the EM-A-MODIFY state, the air EM module shall:

- a) create an ADS-negative-acknowledgement-PDU with elements as defined in Table 2.2.1.5-54,
- b) pass it to the air LI module, and
- c) enter the EM-A-ACTIVE state.

Table 2.2.1.5-54

PDU Element Name	Derivation of Element Value
request-type	modify-emergency-contract
reason	cannot-meet-reporting-rate

2.2.1.5.3.13.6 Upon receipt of an ADS-cancel-emergency-acknowledgement-PDU:

2.2.1.5.3.13.6.1 If in the EM-A-CANCEL state, the air EM module shall:

- a) stop the t-EM-3 timer, and
- b) enter the EM-A-IDLE state.

2.2.1.5.3.13.7 Upon receipt of an ADS-modify-emergency-contract-PDU:

2.2.1.5.3.13.7.1 If in the EM-A-ACTIVE state, the air EM module shall:

- a) request the HI module to invoke ADS-modify-emergency-contract indication, and
- b) enter the EM-A-MODIFY state.

2.2.1.5.3.13.7.2 If in the EM-A-CANCEL state, the air EM module:

- a) shall remain in the EM-A-CANCEL state.

2.2.1.5.3.13.8 Upon expiry of the t-EM-3, the air EM module shall:

- a) request the air AB module to abort with reason *timer-expiry*, and
- b) enter the EM-A-IDLE state.

2.2.1.5.3.13.9 Upon receipt of a request from the air AB or air LI module to stop operation, the air EM module shall:

- a) stop any timers, and
- b) enter the EM-A-IDLE state.

2.2.1.5.3.14 Ground and Air ADS AB Modules

Note.— All statements in 2.2.1.5.3.14 apply to both the ADS ground AB module and the ADS air AB module.

2.2.1.5.3.14.1 Upon receipt of an ADS-user-abort request, the AB module shall:

- a) request the DC, EC, PC and EM modules to stop operation, and
- b) request the LI module to invoke D-ABORT with parameter values derived as in Table 2.2.1.5-55.

Table 2.2.1.5-55

Parameter Name	Derivation of Parameter Value
originator	user
user data	not provided

2.2.1.5.3.14.2 Upon receipt of a request to abort from the LI, DC, EC, PC or EM modules, the AB module shall:

- a) request the DC, EC, PC and EM modules to stop operation,
- b) create an ADS-provider-abort-PDU with elements as defined in Table 2.2.1.5-56,
- c) request the LI module to invoke D-ABORT with parameter values derived as in Table 2.2.1.5-57, and
- d) request the HI module to invoke ADS-provider-abort with parameter values derived as in Table 2.2.1.5-58.

Table 2.2.1.5-56

PDU Element Name	Derivation of Element Value
AbortReason	Value provided by DC, EC, PC or EM module

Table 2.2.1.5-57

Parameter Name	Derivation of Parameter Value
originator	provider
user data	the ADS-provider-abort-PDU

Table 2.2.1.5-58

Parameter Name	Derivation of Parameter Value
Reason	Value provided by DC, EC, PC or EM module

2.2.1.5.3.14.3 Upon receipt of a D-P-ABORT indication, the AB module shall:

- a) request the DC, EC, PC and EM modules to stop operation, and
- b) request the HI module to invoke ADS-provider-abort indication with parameter values derived as in Table 2.2.1.5-59.

Table 2.2.1.5-59

Parameter Name	Derivation of Parameter Value
Reason	<i>communications-service-failure</i>

2.2.1.5.3.14.4 Upon receipt of a D-ABORT indication with the *originator* parameter value set to the abstract value “user” and with data in the *User Data* parameter, the AB module shall:

- a) request the DC, EC, PC and EM modules to stop operation, and
- b) request the HI module to invoke ADS-user-abort indication.

2.2.1.5.3.14.5 Upon receipt of a D-ABORT indication with the *originator* parameter value set to the abstract value “provider”, the AB module shall:

- a) request the DC, EC, PC and EM modules to stop operation, and
- b) request the HI module to invoke ADS-provider-abort indication with parameter values derived as in Table 2.2.1.5-60.

Table 2.2.1.5-60

Parameter Name	Derivation of Parameter Value
Reason	value of the <i>user data</i> parameter

2.2.1.5.3.15 Ground ADS LI Module

Note.— The states defined for the ground ADS LI module are the following:

- a) *LI-G-IDLE*
- b) *LI-G-START*
- c) *LI-G-ACTIVE*
- d) *LI-G-END*

2.2.1.5.3.15.1 On initiation, the ground LI module shall be in the *LI-G-IDLE* state.

2.2.1.5.3.15.2 Upon receipt of an ADS-demand-contract-PDU, an ADS-event-contract-PDU, or an ADS-periodic-contract-PDU from the ground DC, EC or PC modules:

2.2.1.5.3.15.2.1 If in the *LI-G-IDLE* state, the ground LI module shall:

- a) Invoke D-START request using parameter values as shown in Table 2.2.1.5-61, and
- b) enter the *LI-G-START* state.

Table 2.2.1.5-61: D-START request parameter values

D-START parameter	Source
<i>Called peer Id</i>	<i>Aircraft address</i> parameter value from contract request
<i>Calling peer Id</i>	<i>ICAO facility designation</i> parameter value from contract request
<i>DS-user version number</i>	Not used
<i>Security requirements</i>	Not used
<i>Quality of service</i>	Routing class:ATSC, with value from <i>Class of communication service</i> parameter value from contract request Priority: High priority flight safety messages RER: Low
<i>User data</i>	The contract PDU passed to LI

2.2.1.5.3.15.2.2 If in the *LI-G-ACTIVE* state, the ground LI module shall:

- a) Invoke D-DATA request with the PDU as the *user data* parameter value, and
- b) remain in the *LI-G-ACTIVE* state.

2.2.1.5.3.15.3 Upon receipt of an ADS-cancel-all-contracts request :

2.2.1.5.3.15.3.1 If in the LI-G-ACTIVE state, the ground LI module shall:

- a) Invoke D-END req with ADS-cancel-all-contracts-PDU in the user data,
- b) start the t-LI-1 timer, and
- c) enter the LI-G-END state.

2.2.1.5.3.15.4 Upon receipt of an ADS-cancel-contract-PDU, ADS-modify-emergency-contract-PDU, or ADS-cancel-emergency-acknowledgement-PDU from the ground EC, PC or EM modules:

2.2.1.5.3.15.4.1 If in the LI-G-ACTIVE state, the ground LI module shall:

- a) invoke D-DATA req with the PDU in the user data,
- b) if the DC module is in the DC-G-IDLE state, and the EC module is in the EC-G-IDLE state, and the PC module is in the PC-G-IDLE state, and the ground EM module is in the EM-G-IDLE state, then:
 - 1) invoke D-END req with no user data,
 - 2) start the t-LI-1 timer, and
 - 3) enter the LI-G-END state; or
- c) otherwise, remain in the LI-G-ACTIVE state.

2.2.1.5.3.15.4.2 If in the LI-G-END state, the ground LI module shall remain in the LI-G-END state.

2.2.1.5.3.15.5 Upon receipt of request to invoke D-ABORT from the ground AB module:

2.2.1.5.3.15.5.1 If in the LI-G-START state, the LI-G-ACTIVE state or the LI-G-END state, the ground LI module shall:

- a) invoke D-ABORT request with the parameter values supplied, and
- b) enter the LI-G-IDLE state.

2.2.1.5.3.15.5.2 If in the LI-G-IDLE state, the ground LI module shall ignore the request.

2.2.1.5.3.15.6 Upon receipt of a D-START confirmation with the *Result* parameter value containing the abstract value accepted:

2.2.1.5.3.15.6.1 If in the LI-G-START state, the ground LI module shall:

- a) pass the user data to the module as defined in Table 2.2.1.5-62,

- b) if, after processing the PDU (i.e. the ground HI module has issued the appropriate indication or confirmation), the ground DC module is in the DC-G-IDLE state, and the ground EC module is in the EC-G-IDLE state, and the ground PC module is in the PC-G-IDLE state, and the ground EM module is in the EM-G-IDLE state, then:
 - 1) invoke D-END req with no user data, and
 - 2) start the t-LI-1 timer, and
 - 3) enter the LI-G-END state; or
- c) if, after processing the PDU, the ground DC module is not in the DC-G-IDLE state, or the ground EC module is not in the EC-G-IDLE state, or the ground PC module is not in the PC-G-IDLE state, or the ground EM module is not in the EM-G-IDLE state, then:
 - 1) enter the LI-G-ACTIVE state.

2.2.1.5.3.15.7 Upon receipt of a D-DATA indication:

2.2.1.5.3.15.7.1 If in the LI-G-ACTIVE state, the ground LI module shall:

- a) pass the user data to the module as defined in Table 2.2.1.5-62,
- b) if, after processing the PDU (i.e. the ground HI module has issued the appropriate indication or confirmation), the ground DC module is in the DC-G-IDLE state, and the ground EC module is in the EC-G-IDLE state, and the ground PC module is in the PC-G-IDLE state, and the ground EM module is in the EM-G-IDLE state, then:
 - 1) invoke D-END req with no user data, and
 - 2) start the T-LI-1 timer, and
 - 3) enter the LI-G-END state; or
- c) otherwise, remain in the LI-G-ACTIVE state.

2.2.1.5.3.15.7.2 If in the LI-G-END state, the ground LI module shall:

- a) pass the user data to the module as defined in Table 2.2.1.5-62, and
- b) remain in the LI-G-END state.

Table 2.2.1.5-62: PDU to ground module mapping

PDU	Subfield	Ground Module
ADS-demand-report-PDU		DC
ADS-event-report-PDU		EC
ADS-periodic-report-PDU		PC
ADS-emergency-report-PDU		EM
ADS-cancel-emergency-PDU		EM
ADS-positive-acknowledgement-PDU	RequestType	
	cancel-event-contract	EC
	cancel-periodic-contract	PC
	demand-contract	DC
	event-contract	EC
	modify-emergency-contract	EM
	periodic-contract	PC
ADS-noncompliance-notification-PDU	contract-type	
	demand-contract	DC
	event-contract	EC
	periodic-contract	PC
ADS-negative-acknowledgement-PDU	RequestType	
	demand-contract	DC
	event-contract	EC
	modify-emergency-contract	EM
	periodic-contract	PC

2.2.1.5.3.15.8 Upon receipt of a D-END confirmation with the *result* parameter value containing the abstract value *accepted*:

2.2.1.5.3.15.8.1 If in the LI-G-END state, the ground LI module shall:

- a) stop the t-LI-1 timer,
- b) if the user data parameter value contains an ADS-positive-acknowledgement-PDU (cancel-all-contracts), then request the ground HI module to invoke ADS-cancel-all-contracts confirmation,
- c) request the DC, EC, PC and EM modules to stop operation, and
- d) enter the LI-G-IDLE state.

2.2.1.5.3.15.9 Upon receipt of a D-ABORT indication, the ground LI module shall:

- a) pass the D-ABORT indication to the ground AB module, and
- b) enter the LI-G-IDLE state.

2.2.1.5.3.15.10 Upon receipt of a D-P-ABORT indication, the ground LI module shall:

- a) pass the D-P-ABORT indication to the ground AB module, and
- b) enter the LI-G-IDLE state.

2.2.1.5.3.15.11 Upon expiry of the t-LI-1 timer, the ground LI module shall:

- a) request the air AB module to abort with reason *timer-expiry*, and
- b) remain in the same state.

2.2.1.5.3.16 Air ADS LI Module

Note.— The states defined for the air ADS LI module are the following:

- a) *LI-A-IDLE*
- b) *LI-A-START*
- c) *LI-A-ACTIVE*

2.2.1.5.3.16.1 On initiation, the air LI module shall be in the LI-A-IDLE state.

2.2.1.5.3.16.2 Upon receipt of an ADS-demand-report-PDU, or an ADS-event-report-PDU, or an ADS-periodic-report-PDU, or an ADS-positive-acknowledgement-PDU, or an ADS-negative-acknowledgement-PDU, or an ADS-noncompliance-notification-PDU from the air DC, EC, PC or EM modules:

2.2.1.5.3.16.2.1 If in the LI-A-START state, the air LI module shall:

- a) Invoke D-START response using parameter values as shown in Table 2.2.1.5-63, and
- b) enter the LI-A-ACTIVE state.

Table 2.2.1.5-63: D-START response parameter values

D-START parameter	Source
<i>DS-user version number</i>	Not used
<i>Security requirements</i>	Not used
<i>Quality of service</i>	Not used
<i>Result</i>	Accepted
<i>User data</i>	The PDU passed to the air LI module

2.2.1.5.3.16.2.2 If in the LI-A-ACTIVE state, the air LI module shall:

- a) Invoke D-DATA request using the PDU as the *user data* parameter value, and
- b) remain in the LI-A-ACTIVE state.

2.2.1.5.3.16.3 Upon receipt of an ADS-cancel-emergency-PDU, or an ADS-emergency-report-PDU from the EM module:

2.2.1.5.3.16.3.1 If in the LI-A-ACTIVE state, the air LI module shall:

- a) Invoke D-DATA request using the PDU as the *user data* parameter value, and
- b) remain in the LI-A-ACTIVE state.

2.2.1.5.3.16.4 Upon receipt of a request to invoke D-ABORT from the air AB module:

2.2.1.5.3.16.4.1 The air LI module shall:

- a) If a dialogue exists, invoke D-ABORT request with the parameter values supplied, and
- b) enter the LI-A-IDLE state.

2.2.1.5.3.16.5 Upon receipt of a D-START indication:

2.2.1.5.3.16.5.1 If in the LI-A-IDLE state, and the application service priority parameter value is “high priority flight safety messages”, the RER quality of service parameter is the abstract value “low”, the Routing Class parameter identifies the traffic category “Air Traffic Service Communications (ATSC)” and the Calling Peer ID parameter is a valid four to eight character facility designation, the air LI module shall:

- a) pass the user data and the calling peer id parameter value to the module as defined in Table 2.2.1.5-64, and
- b) enter LI-A-START state.

2.2.1.5.3.16.6 Upon receipt of a D-DATA indication:

2.2.1.5.3.16.6.1 If in the LI-A-ACTIVE state, the air LI module shall:

- a) pass the user data to the module as defined in Table 2.2.1.5-64, and
- b) remain in the LI-A-ACTIVE state.

Table 2.2.1.5-64: PDU to air module mapping

PDU type	Sub-element	Air Module
ADS-demand-contract-PDU		DC
ADS-event-contract-PDU		EC
ADS-periodic-contract-PDU		PC
ADS-modify-emergency-contract-PDU		EM
ADS-cancel-emergency-acknowledgement-PDU		EM
ADS-cancel-all-contracts-PDU		HI
ADS-cancel-contract-PDU	CancelContract	
	event-contract	EC
	periodic-contract	PC

2.2.1.5.3.16.7 Upon receipt of a D-END indication:

2.2.1.5.3.16.7.1 If in the LI-A-ACTIVE state:

2.2.1.5.3.16.7.1.1 If the user data parameter value contains an ADS-cancel-all-contracts-PDU, the air LI module shall:

- a) pass it to the air HI module,
- b) invoke D-END response with the *Result* parameter value set to *accepted* and ADS-positive-acknowledgement-PDU (cancel-all-contracts) in the *user data*,

- c) request the DC, EC, PC and EM modules to stop operation, and
- d) enter LI-A-IDLE state.

2.2.1.5.3.16.7.1.2 If there is no user data, the air LI module shall:

- a) invoke D-END response with the *result* parameter value set to *accepted* and no *user data*, and
- b) enter LI-A-IDLE state.

2.2.1.5.3.16.8 Upon receipt of a D-ABORT indication, the air LI module shall:

- a) stop any timer, and
- b) pass the D-ABORT indication to the air AB module, and
- c) enter the LI-A-IDLE state.

2.2.1.5.3.16.9 Upon receipt of a D-P-ABORT indication, the air LI module shall:

- a) stop any timer, and
- b) pass the D-P-ABORT indication to the air AB module, and
- c) enter the LI-A-IDLE state.

2.2.1.5.4 Exception Handling

2.2.1.5.4.1 Timer Expires

2.2.1.5.4.1.1 When any of the timers in any of the modules stated in 2.2.1.5.2 reaches its maximum time, the module shall request the air or ground AB module to abort with reason *timer-expiry*.

2.2.1.5.4.2 Unrecoverable System Error

2.2.1.5.4.2.1 **Recommendation.**—When any module has an unrecoverable system error, the module should request the air or ground AB module to abort with reason *unrecoverable-system-error*.

2.2.1.5.4.3 Invalid PDU

2.2.1.5.4.3.1 When the user data parameter value of a D-START indication is a valid APDU and is not an ADS-demand-contract-PDU, an ADS-event-contract-PDU or an ADS-periodic-contract-PDU, the air LI module shall request the AB module to abort with reason *invalid-PDU*.

2.2.1.5.4.3.2 When the user data parameter value of a D-START confirmation is a valid APDU and is not an ADS-demand-report-PDU, an ADS-event-report-PDU, an ADS-periodic-report-PDU, an ADS-positive-acknowledgement-PDU, and ADS-negative-acknowledgement-PDU or an ADS-noncompliance-notification-PDU the ground LI module shall request the AB module to abort with reason *invalid-PDU*.

2.2.1.5.4.3.3 When the user data parameter value of a D-DATA indication is a valid APDU and is not an ADS-demand-contract-PDU, an ADS-event-contract-PDU, an ADS-periodic-contract-PDU, an ADS-modify-emergency-contract-PDU, an ADS-cancel-emergency-acknowledgement-PDU or an ADS-cancel-contract-PDU, the air LI module shall request the AB module to abort with reason *invalid-PDU*.

2.2.1.5.4.3.4 When the user data parameter value of a D-DATA indication is a valid APDU and is not an ADS-demand-report-PDU, an ADS-event-report-PDU, an ADS-periodic-report-PDU, an ADS-positive-acknowledgement-PDU, and ADS-negative-acknowledgement-PDU, an ADS-noncompliance-notification-PDU, an ADS-emergency-report-PDU or an ADS-cancel-emergency-PDU the ground LI module shall request the AB module to abort with reason *invalid-PDU*.

2.2.1.5.4.3.5 When the user data parameter value of a D-END indication is present, but does not contain an ADS-cancel-all-contracts-PDU, the air LI module shall request the AB module to abort with reason *invalid-PDU*.

2.2.1.5.4.3.6 When the user data parameter value of a D-ABORT indication is a valid APDU and is not an ADS-provider-abort-PDU, the air LI module or the ground LI module shall request the AB module to abort with reason *invalid-PDU*.

2.2.1.5.4.3.7 When the user data parameter value of a D-END confirmation is present, but does not contain an ADS-positive-acknowledgement-PDU (cancel-all-contracts), the ground LI module shall request the AB module to abort with reason *invalid-PDU*.

2.2.1.5.4.4 Sequence Error

2.2.1.5.4.4.1 When a PDU is passed to a module for which there are no instructions in 2.2.1.5.3 (i.e. the PDU has arrived out of sequence), the air or ground AB module shall be requested to abort with reason *sequence-error*.

2.2.1.5.4.4.2 Upon receipt of a Dialogue service primitive for which there are no instruction in 2.2.1.5.3 (i.e. the primitive was not expected or was expected under other conditions or with other parameter values), the air or ground AB module shall be requested to abort with reason *sequence-error*.

2.2.1.5.4.5 D-START Rejection

2.2.1.5.4.5.1 Upon receipt of a D-START confirmation with the *result* parameter value containing the abstract value rejected (transient) or rejected (permanent), and the *reject source* parameter value containing the abstract value DS user, the ground LI module shall:

- a) request the ground AB module to abort with reason *sequence-error*; and
- b) enter the LI-G-IDLE state.

2.2.1.5.4.5.2 Upon receipt of a D-START confirmation with the *result* parameter value containing the abstract value rejected (transient) or rejected (permanent), and the *reject source* parameter value containing the abstract value DS provider, the ground LI module shall:

- a) request the ground AB module to abort with reason *cannot-establish-contact*; and
- b) enter the LI-G-IDLE state.

2.2.1.5.4.6 D-END Rejection

2.2.1.5.4.6.1 Upon receipt of a D-END confirmation with the *result* parameter value containing the abstract value rejected, the ground AB module shall be requested to abort with reason *dialogue-end-not-accepted*.

2.2.1.5.4.7 Decoding Error

2.2.1.5.4.7.1 When the air LI module or the ground LI module fails to decode an APDU, the LI module shall request the AB module to abort with reason *decoding-error*.

2.2.1.5.4.7.2 Upon receipt of a D-START indication, a D-START confirmation or a D-DATA indication with no user data, the air or ground AB module shall be requested to abort with reason *decoding-error*.

2.2.1.5.4.8 Invalid QOS

2.2.1.5.4.8.1 Upon receipt of a D-START indication with the application service priority parameter set to a value other than the abstract value “high priority flight safety messages”, or the RER quality of service parameter set to a value other than the abstract value “low” or the Routing Class quality of service parameter set to a value other than one identifying the traffic category “Air Traffic Service Communications (ATSC)”, the air LI module shall request the air AB module to abort with reason *invalid-qos-parameter*.

2.2.1.5.5 ADS-ASE State Tables

2.2.1.5.5.1 Priority

2.2.1.5.5.1.1 If the state tables for the ADS-air-ASE and the ADS-ground-ASE shown below conflict with textual statements made elsewhere in this document, the textual statements shall take precedence.

Note 1.— In the following state tables, the statement “cannot occur” means that if the implementation conforms to the SARPs, it is impossible for this event to occur. If the event does occur, this implies that there is an error in the implementation. If such a situation is detected, it is suggested that the ASE aborts with the error “unrecoverable error”.

Note 2.— In the following state tables, the statement “not permitted” means that the implementation must prevent this event from occurring through some local means. If the event does occur this implies that there is an error in the implementation. If such a situation is detected, it is suggested that the ASE performs a local rejection of the request rather than aborting the dialogue.

Table 2.2.1.5-65: ADS ground DC module state table

State →	DC-G-IDLE (Initial State)	DC-G-PENDING	DC-G-ACTIVE
Event ↓			
<i>Primitive Requests and Responses</i>			
ADS-demand-contract req	Send ADS-demand-contract-P DU Start t-DC-1 DC-G-PENDING	Not permitted	Not permitted
<i>ADS downlink PDUs</i>			
ADS-demand-report-PDU (with positive acknowledgement)	request AB to abort	Stop t-DC-1 ADS-report ind DC-G-IDLE	request AB to abort
ADS-demand-report-PDU	request AB to abort	request AB to abort	stop t-DC-2 ADS-report ind DC-G-IDLE
ADS-negative-acknowledgement-PDU (demand contract)	request AB to abort	stop t-DC-1 ADS-demand-contract cnf DC-G-IDLE	request AB to abort
ADS-noncompliance-notification-PDU (demand-ncn)	request AB to abort	stop t-DC-1 ADS-demand-contract cnf start t-DC-2 DC-G-ACTIVE	request AB to abort
<i>Requests from other modules</i>			
Request to stop operation	DC-G-IDLE	stop t-DC-1 DC-G-IDLE	stop t-DC-2 DC-G-IDLE
<i>Timer expiry</i>			
t-DC-1	cannot occur	request AB to abort	cannot occur
t-DC-2	cannot occur	cannot occur	request AB to abort

Table 2.2.1.5-66: ADS air DC module state table

State →	DC-A-IDLE (Initial State)	DC-A-PENDING	DC-A-ACTIVE
Event ↓			
<i>Primitive Requests and Responses</i>			
ADS-demand-contract rsp (negative acknowledgement)	Not permitted	Send ADS-negative-acknowledgement-PDU DC-A-IDLE	Not permitted
ADS-demand-contract rsp (noncompliance notification)	Not permitted	Send ADS-noncompliance-notification-PDU DC-A-ACTIVE	Not permitted
ADS-report req (demand contract with positive acknowledgement)	Not permitted	Send ADS-demand-report-PDU DC-A-IDLE	Not permitted
ADS-report req (demand contract)	Not permitted	Not permitted	Send ADS-demand-report-PDU DC-A-IDLE
<i>Requests from other modules</i>			
Requests to stop operation	DC-A-IDLE	DC-A-IDLE	DC-A-IDLE
<i>ADS uplink PDUs</i>			
ADS-demand-contract-PD U	ADS-demand-contract ind DC-A-PENDING	request AB to abort	request AB to abort

Table 2.2.1.5-67: ADS ground EC module state table

State →	EC-G-IDLE (Initial State)	EC-G-START-PENDING	EC-G-ACTIVE	EC-G-PENDING	EC-G-CANCEL
Event ↓					
<i>Primitive Requests and Responses</i>					
ADS-event-contract req	Send ADS-event-contract -PDU Start t-EC-1 EC-G-START-PENDING	Not permitted	Send ADS-event-contract -PDU Start t-EC-1 EC-G-PENDING	Not permitted	Not permitted
ADS-cancel req (event contract)	Not permitted	Not permitted	Send ADS-cancel-PDU Start t-EC-2 EC-G-CANCEL	Not permitted	Not permitted
<i>ADS Aircraft PDUs</i>					
ADS-event-report-PDU (with positive acknowledgement)	request AB to abort	Stop t-EC-1 ADS-report ind EC-G-ACTIVE	request AB to abort	Stop t-EC-1 ADS-report ind EC-G-ACTIVE	request AB to abort
ADS-event-report-PDU	request AB to abort	request AB to abort	ADS-report ind EC-G-ACTIVE	ADS-report ind EC-G-PENDING	ADS-report ind EC-G-CANCEL
ADS-positive-acknowledgement-PDU (event-contract)	request AB to abort	stop t-EC-1 ADS-event-contract cnf EC-G-ACTIVE	request AB to abort	stop t-EC-1 ADS-event-contract cnf EC-G-ACTIVE	request AB to abort
ADS-positive-acknowledgement-PDU (cancel-contract-event)	request AB to abort	request AB to abort	request AB to abort	request AB to abort	Stop t-EC-2 ADS-cancel-contract cnf EC-G-IDLE
ADS-negative-acknowledgement-PDU (event-contract)	request AB to abort	stop t-EC-1 ADS-event-contract cnf EC-G-IDLE	request AB to abort	stop t-EC-1 ADS-event-contract cnf EC-G-ACTIVE	request AB to abort
ADS-noncompliance-notification-PDU (event-ncn)	request AB to abort	stop t-EC-1 ADS-event-contract cnf EC-G-ACTIVE	request AB to abort	stop t-EC-1 ADS-event-contract cnf EC-G-ACTIVE	request AB to abort
<i>Requests from other modules</i>					
Requests to stop operation	EC-G-IDLE	stop t-EC-1 EC-G-IDLE	EC-G-IDLE	stop t-EC-1 EC-G-IDLE	stop t-EC-2 EC-G-IDLE
<i>Timer expiry</i>					
t-EC-1	cannot occur	request AB to abort	cannot occur	request AB to abort	cannot occur
t-EC-2	cannot occur	cannot occur	cannot occur	cannot occur	request AB to abort

Table 2.2.1.5-68: ADS air EC module state table

<i>State</i> ⇒	<i>EC-A-IDLE</i> (Initial State)	<i>EC-A-PENDING</i>	<i>EC-A-ACTIVE</i>	<i>EC-A-ACTIVE-PENDING</i>
<i>Event</i> ∕				
<i>Primitive Requests and Responses</i>				
ADS-event-contract rsp (positive acknowledgement or noncompliance notification)	Not permitted	Send ADS-positive-acknowledgement-PDU or ADS-noncompliance-notification-PDU <i>EC-A-ACTIVE</i>	Not permitted	Send ADS-positive-acknowledgement-PDU or ADS-noncompliance-notification-PDU <i>EC-A-ACTIVE</i>
ADS-event-contract rsp (negative acknowledgement)	Not permitted	Send ADS-negative-acknowledgement-PDU <i>EC-A-IDLE</i>	Not permitted	Send ADS-negative-acknowledgement-PDU <i>EC-A-ACTIVE</i>
ADS-report req (with positive acknowledgement) (event contract)	Not permitted	Send ADS-event-report-PDU <i>EC-A-ACTIVE</i>	Not permitted	Send ADS-event-report-PDU <i>EC-A-ACTIVE</i>
ADS-report req (event contract)	Not permitted	Not permitted	Send ADS-event-report-PDU <i>EC-A-ACTIVE</i>	Not permitted
<i>Requests from other modules</i>				
Requests to stop operation	<i>EC-A-IDLE</i>	<i>EC-A-IDLE</i>	<i>EC-A-IDLE</i>	<i>EC-A-IDLE</i>
<i>ADS Ground PDUs</i>				
ADS-event-contract-PDU	ADS-event-contract ind <i>EC-A-PENDING</i>	request AB to abort	ADS-event-contract ind <i>EC-A-ACTIVE-PENDING</i>	request AB to abort
ADS-cancel-PDU (event contract)	request AB to abort	request AB to abort	ADS-cancel ind Send ADS-positive-acknowledgement <i>EC-A-IDLE</i>	request AB to abort

Table 2.2.1.5-69: ADS-periodic-contract ground based state table

State →	PC-G-IDLE (Initial State)	PC-G-START-PENDING	PC-G-ACTIVE	PC-G-PENDING	PC-G-CANCEL
Event ↓					
<i>Primitive Requests and Responses</i>					
ADS-periodic-contract req	Send ADS-periodic-contract -PDU Start t-PC-1 <i>PC-G-START-PENDING</i>	Not permitted	If emergency=FALSE then Stop t-PC-2 Send ADS-periodic-contract -PDU Start t-PC-1 <i>PC-G-PENDING</i>	Not permitted	Not permitted
ADS-cancel req	Not permitted	Not permitted	Stop t-PC-2 Send ADS-cancel-PDU Start t-PC-3 <i>PC-G-CANCEL</i>	Not permitted	Not permitted
<i>ADS Aircraft PDUs</i>					
ADS-periodic-report-PDU (with positive acknowledgement)	request AB to abort	Stop t-PC-1 ADS-report ind If emergency=FALSE then Start t-PC-2 <i>PC-G-ACTIVE</i>	request AB to abort	Stop t-PC-1 ADS-report ind If emergency=FALSE then Start t-PC-2 <i>PC-G-ACTIVE</i>	request AB to abort
ADS-periodic-report-PDU (with no positive acknowledgement)	request AB to abort	request AB to abort	If emergency=FALSE then Stop t-PC-2 ADS-report ind If emergency=FALSE then Start t-PC-2 <i>PC-G-ACTIVE</i>	ADS-report ind <i>PC-G-PENDING</i>	ADS-report ind <i>PC-G-CANCEL</i>
ADS-positive-acknowledgement-PDU (periodic-contract)	request AB to abort	stop t-PC-1 ADS-periodic-contract cnf If emergency=FALSE then Start t-PC-2 <i>PC-G-ACTIVE</i>	request AB to abort	stop t-PC-1 ADS-periodic-contract cnf If emergency=FALSE then Start t-PC-2 <i>PC-G-ACTIVE</i>	request AB to abort
ADS-positive-acknowledgement-PDU (cancel-contract - periodic)	request AB to abort	request AB to abort	request AB to abort	request AB to abort	Stop t-PC-3 ADS-cancel-contract cnf <i>PC-G-IDLE</i>

<i>State</i> ⇒	<i>PC-G-IDLE</i> (Initial State)	<i>PC-G-START-PENDING</i>	<i>PC-G-ACTIVE</i>	<i>PC-G-PENDING</i>	<i>PC-G-CANCEL</i>
<i>Event</i> ↴					
ADS-negative-acknowledgement-PDU (periodic-contract)	request AB to abort	stop t-PC-1 ADS-periodic-contract cnf <i>PC-G-IDLE</i>	request AB to abort	stop t-PC-1 ADS-periodic-contract cnf If emergency=FALSE then Start t-PC-2 <i>PC-G-ACTIVE</i>	request AB to abort
ADS-noncompliance-notification-PDU	request AB to abort	stop t-PC-1 ADS-periodic-contract cnf If emergency=FALSE then Start t-PC-2 <i>PC-G-ACTIVE</i>	request AB to abort	stop t-PC-1 ADS-periodic-contract cnf If emergency=FALSE then Start t-PC-2 <i>PC-G-ACTIVE</i>	request AB to abort
<i>Requests from other modules</i>					
Request to stop operation	set emergency=FALSE <i>PC-G-IDLE</i>	stop t-PC-1 set emergency=FALSE <i>PC-G-IDLE</i>	If emergency=FALSE then stop t-PC-2 set emergency=FALSE <i>PC-G-IDLE</i>	stop t-PC-1 set emergency=FALSE <i>PC-G-IDLE</i>	stop t-PC-3 set emergency=FALSE <i>PC-G-IDLE</i>
Request to suspend periodic contract	set emergency=TRUE	set emergency=TRUE	set emergency=TRUE stop t-PC-2	set emergency=TRUE	set emergency=TRUE
Request to reinstate periodic contract	set emergency=FALSE	set emergency=FALSE	set emergency=FALSE start t-PC-2	set emergency=FALSE	set emergency=FALSE
<i>Timer expiry</i>					
t-PC-1	cannot occur	request AB to abort	cannot occur	request AB to abort	cannot occur
t-PC-2	cannot occur	cannot occur	request AB to abort	cannot occur	cannot occur
t-PC-3	cannot occur	cannot occur	cannot occur	cannot occur	request AB to abort

Table 2.2.1.5-70: ADS air PC module state table

State ⇒	PC-A-IDLE (Initial State)	PC-A-PENDING	PC-A-ACTIVE	PC-A-ACTIVE-PENDING
Event ↴				
<i>Primitive Requests and Responses</i>				
ADS-periodic-contract rsp (positive acknowledgement or noncompliance notification)	Not permitted	Send ADS-positive- acknowledgement-PDU or ADS- noncompliance-notification-PDU <i>PC-A-ACTIVE</i>	Not permitted	Send ADS-positive- acknowledgement-PDU or ADS-noncompliance- notification-PDU <i>PC-A-ACTIVE</i>
ADS-periodic-contract rsp (negative acknowledgement)	Not permitted	Send ADS-negative- acknowledgement-PDU <i>PC-A-IDLE</i>	Not permitted	Send ADS-negative- acknowledgement-PDU <i>PC-A-ACTIVE</i>
ADS-report req - with positive acknowledgement (periodic contract)	Not permitted	Send ADS-periodic-report-PDU <i>PC-A-ACTIVE</i>	Not permitted	Send ADS-periodic-report-PDU <i>PC-A-ACTIVE</i>
ADS-report req (periodic contract)	Not permitted	Not permitted	Send ADS-periodic-report-PDU <i>PC-A-ACTIVE</i>	Not permitted
<i>Requests from other modules</i>				
Requests to stop operation	<i>PC-A-IDLE</i>	<i>PC-A-IDLE</i>	<i>PC-A-IDLE</i>	<i>PC-A-IDLE</i>
<i>ADS uplink PDUs</i>				
ADS-periodic-contract-PDU	ADS-periodic-contract ind <i>PC-A-PENDING</i>	request AB to abort	ADS-periodic-contract ind <i>PC-A-ACTIVE-PENDING</i>	request AB to abort
ADS-cancel-PDU (periodic contract)	request AB to abort	request AB to abort	ADS-cancel ind Send ADS-positive- acknowledgement <i>PC-A-IDLE</i>	request AB to abort

Table 2.2.1.5-71: ADS ground EM module state table

State $\Rightarrow$	EM-G-IDLE	EM-G-ACTIVE	EM-G-MODIFY
Event $\Downarrow$			
<i>Primitive Requests and Responses</i>			
ADS-modify-emergency-contract req	Not permitted	Stop t-EM-1 Send ADS-modify-emergency-contract-PDU Start t-EM-2 <i>EM-G-MODIFY</i>	Not permitted
<i>ADS Aircraft PDUs</i>			
ADS-emergency-report-PDU (with positive acknowledgement)	request AB to abort	request AB to abort	Stop t-EM-2 ADS-emergency-report ind Start t-EM-1 <i>EM-G-ACTIVE</i>
ADS-emergency-report-PDU	Suspend periodic contract ADS-emergency-report ind Start t-EM-1 <i>EM-G-ACTIVE</i>	Stop t-EM-1 ADS-emergency-report ind Start t-EM-1 <i>EM-G-ACTIVE</i>	ADS-emergency-report ind <i>EM-G-MODIFY</i>
ADS-cancel-emergency-PDU	request AB to abort	Stop t-EM-1 ADS-cancel-emergency ind Send ADS-cancel-emergency-acknowledgement-PDU Re-instate periodic contracts <i>EM-G-IDLE</i>	Stop t-EM-2 ADS-cancel-emergency ind Send ADS-cancel-emergency-acknowledgement-PDU Re-instate periodic contracts <i>EM-G-IDLE</i>
ADS-negative-acknowledgement-PDU (modify-emergency-contract)	request AB to abort	request AB to abort	Stop t-EM-2 ADS-modify-emergency-contract cnf Start t-EM-1 <i>EM-G-ACTIVE</i>
<i>Requests from other modules</i>			
Requests to stop operation	<i>EM-G-IDLE</i>	stop t-EM-1 <i>EM-G-IDLE</i>	stop t-EM-2 <i>EM-G-IDLE</i>
<i>Timer expiry</i>			
t-EM-1	cannot occur	request AB to abort	cannot occur
t-EM-2	cannot occur	cannot occur	request AB to abort

Table 2.2.1.5-72: ADS air EM module state table

State $\Rightarrow$	EM-A-IDLE	EM-A-ACTIVE	EM-A-MODIFY	EM-A-CANCEL
Event $\Downarrow$				
<i>Primitive Requests and Responses</i>				
ADS-emergency-report req	Send ADS-emergency-report-PDU EM-A-ACTIVE	Send ADS-emergency-report-PDU EM-A-ACTIVE	Not permitted	Not permitted
ADS-emergency-report req (with positive acknowledgement)	Not permitted	Not permitted	Send ADS-emergency-report-PDU EM-A-ACTIVE	Not permitted
ADS-cancel-emergency req	Not permitted	Send ADS-cancel-emergency-PDU Start t-EM-3 EM-A-CANCEL	Not permitted	Not permitted
ADS-modify-emergency-contract rsp	Not permitted	Not permitted	Send ADS-negative-acknowledgement-PDU EM-A-ACTIVE	Not permitted
<i>ADS Ground PDUs</i>				
ADS-cancel-emergency-acknowledgement-PDU	request AB to abort	request AB to abort	request AB to abort	Stop t-EM-3 EM-A-IDLE
ADS-modify-emergency-contract-PDU	request AB to abort	ADS-modify-emergency-contract ind EM-A-MODIFY	request AB to abort	EM-A-CANCEL
<i>Requests from other modules</i>				
Requests to stop operation	EM-A-IDLE	EM-A-IDLE	EM-A-IDLE	stop t-EM-3 EM-A-IDLE
<i>Timer expiry</i>				
t-EM-3	cannot occur	cannot occur	cannot occur	request AB to abort

Table 2.2.1.5-73: Ground ADS LI module state table

State →	LI-G-IDLE (Initial State)	LI-G-START	LI-G-ACTIVE	LI-G-END
Event ↓				
<i>Data and requests passed from other modules</i>				
ADS-demand-contract-request-PDU, ADS-event-contract-request-PDU, or ADS-periodic-contract-request-PDU,	D-START req LI-G-START	Not permitted	D-DATA req LI-G-ACTIVE	Not permitted
ADS-cancel-all-contracts req	Not permitted	Not permitted	start t-LI-1 D-END req LI-G-END	Not permitted
ADS-cancel-contract-PDU, ADS-modify-emergency-contract-PDU or ADS-cancel-emergency-acknowledgement-PDU	Not permitted	Not permitted	D-DATA req [1]	LI-G-END
ADS-provider-abort-PDU	LI-G-IDLE	D-ABORT req LI-G-IDLE	D-ABORT req LI-G-IDLE	D-ABORT req LI-G-IDLE
ADS-forward-contract-response-PDU	Not permitted	Not permitted	Not permitted	Not permitted
<i>Primitive Indications and Confirmations</i>				
D-START ind	pass user data to appropriate module LI-G-START-R	cannot occur	cannot occur	cannot occur
D-START cnf	cannot occur	pass user data to appropriate module [1]	cannot occur	cannot occur
D-DATA ind	cannot occur	cannot occur	pass user data to appropriate module [1]	pass user data to appropriate module LI-G-END
D-END-ind	cannot occur	cannot occur	if ADS-end-forward-service-PDU, pass to HI module D-END rsp LI-G-IDLE	cannot occur
D-END cnf	cannot occur	cannot occur	cannot occur	stop t-LI-1 ADS-cancel-all-contracts cnf LI-G-IDLE
D-ABORT ind or D-P-ABORT ind	cannot occur	pass to AB module LI-G-IDLE	pass to AB module LI-G-IDLE	pass to AB module LI-G-IDLE
Timer expiry				
t-LI-1	cannot occur	cannot occur	cannot occur	request AB to abort

[1] If DC, EC, PC, and EM modules are all in their idle state then
 Invoke D-END req with no user data
 LI-G-END
 else
 LI-G-ACTIVE

Table 2.2.1.5-74: Air ADS LI module state table

<i>State</i> ⇒	<i>LI-A-IDLE</i> (Initial State)	<i>LI-A-START</i>	<i>LI-A-ACTIVE</i>
<i>Event</i> ↓			
<i>Data and requests passed from other modules</i>			
ADS-demand-report-PDU, ADS-negative-acknowledgement-PDU, ADS-event-report-PDU, ADS-periodic-report-PDU, ADS-positive-acknowledgement-PDU, ADS-noncompliance-notification-PDU	Not permitted	D-START rsp <i>LI-A-ACTIVE</i>	D-DATA req <i>LI-A-ACTIVE</i>
ADS-cancel-emergency-PDU, ADS-emergency-report-PDU	Not permitted	Not permitted	D-DATA req <i>LI-A-ACTIVE</i>
<i>Primitive Indications and Confirmations</i>			
D-START ind	pass to appropriate module <i>LI-A-START</i>	cannot occur	cannot occur
D-DATA ind	cannot occur	cannot occur	pass user data to appropriate module <i>LI-A-ACTIVE</i>
D-END ind	cannot occur	cannot occur	if ADS-cancel-all-contracts-PDU, pass to HI module D-END rsp <i>LI-A-IDLE</i>
D-ABORT ind or D-P-ABORT ind	cannot occur	pass to AB module <i>LI-A-IDLE</i>	pass to AB module <i>LI-A-IDLE</i>

2.2.1.6 Communication Requirements

2.2.1.6.1 Encoding Rules

2.2.1.6.1.1 The ADS application shall use PER as defined in ISO/IEC 8825-2, using the Basic Unaligned variant to encode/decode the ASN.1 message structure and content specified in 2.2.1.4.

2.2.1.6.2 Dialogue Service Requirements

2.2.1.6.2.1 Primitive Requirements

2.2.1.6.2.1.1 Where dialogue service primitives, that is D-START, D-END, D-ABORT, D-P-ABORT and D-DATA are described as being invoked in 2.2.1.5, the ADS-ground-ASE and the ADS-air-ASE shall exhibit external behaviour consistent with the dialogue service, as described in 4.2, having been implemented and its primitives invoked.

2.2.1.6.2.2 Quality of Service Requirements

2.2.1.6.2.2.1 The application service priority for ADS shall have the abstract value of “high priority flight safety messages”.

2.2.1.6.2.2.2 The RER quality of service parameter of the D-START request shall be set to the abstract value of “low”.

2.2.1.6.2.2.3 The ADS-ASE shall map the class of communication service abstract values to the ATSC routing class abstract value part of the D-START QOS parameter as presented in Table 2.2.1.6-1.

Table 2.2.1.6-1. Mapping between class of communication and routing class abstract values

Class of Communication Abstract Value	Routing Class Abstract Value
A	Traffic follows Class A ATSC route(s)
B	Traffic follows Class B ATSC route(s)
C	Traffic follows Class C ATSC route(s)
D	Traffic follows Class D ATSC route(s)
E	Traffic follows Class E ATSC route(s)
F	Traffic follows Class F ATSC route(s)
G	Traffic follows Class G ATSC route(s)
H	Traffic follows Class H ATSC route(s)

Note.— ATSC values are defined in 1.3.

2.2.1.7 ADS User Requirements

2.2.1.7.1 General

2.2.1.7.1.1 General Requirements

2.2.1.7.1.1.1 The ADS-ground-user shall only establish a demand contract, an event contract or a periodic contract with an ADS-air-user.

2.2.1.7.1.1.2 The ADS-air-user shall invoke ADS-report requests only at the rate specified and containing only the information required to meet the contract as specified in 2.2.1.7.

2.2.1.7.1.2 General Parameter Requirements

Note 1.— When an ADS-ground-user invokes ADS-demand-contract request, ADS-event-contract request, ADS-periodic-contract request or ADS-forward-contract request and requires a particular class of communication service, it provides the class of communication service parameter.

Note 2.— When an ADS-ground-user invokes ADS-demand-contract request, ADS-event-contract request, ADS-periodic-contract request or ADS-forward-contract request, and does not provide the class of communications service parameter, this indicates no routing preference.

Note 3.— When an ADS-ground-user specifies the class of communications service parameter and there is an ADS contract in place, the parameter is ignored.

2.2.1.7.1.2.1 When providing the air speed (as part of the air vector parameter), the ADS-air-user shall:

- a) if available, provide Mach number,
- b) if available, provide indicated air speed, or
- c) if available, provide both Mach number and indicated air speed.

2.2.1.7.1.3 Timing Requirements

2.2.1.7.1.3.1 **Recommendation.**— *When an ADS-air-user or ADS-ground-user receives an indication that requires a response, it should invoke the response within 0.5 seconds.*

2.2.1.7.1.3.2 **Recommendation.**— *When a periodic contract or an emergency contract is in place, the ADS-air-user should invoke ADS-report request or ADS-emergency report request (as described below) within 0.5 second of the reporting interval as measured from the sending of the previous report.*

2.2.1.7.1.4 Error Handling Requirements

2.2.1.7.1.4.1 If the ADS-air-user or ADS-ground-user has an unrecoverable system error, then it shall:

- a) cease the operation of all contracts with peer system(s) which are effected by the error, and

- b) for each effected peer system, invoke ADS-user-abort request.

2.2.1.7.1.4.2 If the ADS-user receives an ADS-user-abort indication or an ADS-provider-abort indication, then it shall cease operation of all ADS contracts with the peer system to which the indication is related.

2.2.1.7.1.5 Miscellaneous Air User Requirements

2.2.1.7.1.5.1 With the permissible exception of ADS-user-abort and ADS-provider-abort, the ADS-air-user shall respond to indications and confirmations in the order in which they are received.

2.2.1.7.1.5.2 The ADS-air-user shall be capable of supporting contracts from at least four different ATC ground systems at the same time.

Note.— The ADS-air-user may use the class of communications service indicated in the D-START indication quality of service parameter value in order to determine if the ground systems is an ATC ground system. How the ADS-air-user finds out what class of communications service parameter value is used is a local matter.

2.2.1.7.1.5.3 If the ADS-air-user receives an ADS-demand-contract indication, or an ADS-event-contract indication or an ADS-periodic-contract indication which exceeds its capacity for supporting ground systems, then it shall:

- a) reject the contract with the *reply* parameter set to *negative acknowledgement*,
- b) set the *reason* element of *negative acknowledgement* to *maximum-capacity-exceeded*, and
- c) include the set of ICAO facility designations of all the ground systems with which it has contracts in the *maximum-capacity-exceeded* element.

2.2.1.7.1.5.4 If the ICAO facility designation parameter is provided in an ADS-demand-contract indication, an ADS-event-contract indication or an ADS-periodic-contract indication, and if this ICAO facility designation is equal to the ICAO facility designations of any other ground system with which the aircraft has one or more contracts, the ADS-air-user shall invoke ADS-user-abort in place of the normal response.

Note.— The intention is that the new connection will be aborted; the existing connection and all the contracts on it will be retained.

2.2.1.7.1.5.5 If, after accepting a contract, the ADS-air-user is unable to provide the information required, either because it is unavailable, invalid or because its validity is uncertain, then:

- a) if the information forms part of position, timestamp or FOM, then:
 - 1) the ADS-air-user shall continue to send ADS reports as required with the FOM set to “0”; and
 - 2) if all of the information is again found to be valid and available, the FOM shall be reset to its actual value, and

- b) if the information is the aircraft-address, the ADS-air-user shall omit the aircraft-address from any ADS-reports or ADS-emergency-reports that require it, and
- c) if the information is part of the projected-profile, the ADS-air-user shall omit the projected-profile from any ADS-reports that require it, and
- d) if the information is part of the ground-vector, the ADS-air-user shall omit the ground-vector from any ADS-reports or ADS-emergency-reports that require it, and
- e) if the information is part of the air-vector, the ADS-air-user shall omit the air-vector from any ADS-reports that require it, and
- f) if the information is part of the weather, the ADS-air-user shall omit the weather from any ADS-reports that require it, and
- g) if the information is part of the short-term-intent, the ADS-air-user shall omit the short-term-intent from any ADS-reports that require it, and
- h) if the information is part of the extended-projected-profile, the ADS-air-user shall omit the extended-projected-profile from any ADS-reports that require it.

Note 1.— If information is not available for more than one optional field, then both are omitted.

Note 2.— The ADS-air-user must be able to detect when information becomes unavailable. The ADS-air-user must be able to detect if the information is invalid, or its validity is uncertain.

Note 3.— The ADS-ground-user will know what information is expected in any ADS-report or ADS-emergency-report. It is therefore able to tell when the information is unavailable or possibly invalid.

2.2.1.7.1.6 Miscellaneous Ground User Requirements

2.2.1.7.1.6.1 With the permissible exception of ADS-user-abort and ADS-provider-abort, the ADS-ground-user shall respond to indications and confirmations in the order in which they are received.

Note.— The ADS-ground-user checks the contents of the ADS reports received, for conformance to the contracts in place, and flags any non-conformance.

2.2.1.7.2 Establishment and operation of a Demand Contract

Note 1.— 2.2.1.7.2 details the actions taken by the ADS-ground-user and the ADS-air-user in the establishment and operation of a demand contract.

Note 2.— When the ADS-ground-user requires to establish a demand contract with the ADS-air-user, it invokes ADS-demand-contract request.

2.2.1.7.2.1 When the ADS-air-user receives an ADS-demand-contract indication, and is not able to accept the contract, the ADS-air-user shall invoke an ADS-demand-contract response with the *reply* parameter set to *negative-acknowledgement*, and the *reason* parameter set to the value indicating the reason that it cannot accept the contract.

2.2.1.7.2.2 When the ADS-air-user receives an ADS-demand-contract indication, and it is able to accept the contract in full, the ADS-air-user shall invoke an ADS-report request including a *positive acknowledgement* parameter.

2.2.1.7.2.3 When the ADS-air-user receives an ADS-demand-contract indication, and it is able to accept the contract, but is not able to supply all the requested information,

2.2.1.7.2.3.1 the ADS-air-user shall:

- a) invoke ADS-demand-contract response, with the *reply* parameter set to *noncompliance-notification*, and the *demand-ncn* parameter element containing an indication of the reports that were requested but cannot be provided, and
- b) invoke ADS-report request containing the information that it is able to send, with the *positive acknowledgement* parameter absent.

2.2.1.7.2.3.2 **Recommendation.**— *the ADS-air-user should invoke ADS-report request containing the information that it is able to send, with the positive acknowledgement parameter absent, within 0.5 seconds.*

2.2.1.7.2.4 Forming the ADS-report request

2.2.1.7.2.4.1 Subject to the restrictions stated in 2.2.1.7.1.5.5, the ADS-air-user invokes ADS-report request in response to an ADS-demand-contract, and only when requested in the ADS-demand-contract indication and not indicated as being unavailable in an ADS-demand-contract response (where the *reply* parameter was set to *noncompliance-notification*), then the ADS-air-user shall form the *report details* parameter with the following information:

- a) *aircraft address*,
- b) *projected-profile*,
- c) *ground-vector*,
- d) *air-vector*,
- e) *weather*,
- f) *short-term-intent*, and
- g) *extended-projected-profile*.

2.2.1.7.2.4.2 When *short-term-intent* is provided it shall cover the time period indicated in *short-term-intent*.

2.2.1.7.2.4.3 When *number-of-way-points* was provided in the ADS-demand-contract indication, and *extended-projected-profile* is provided in the subsequent ADS-report request, the *extended-projected-profile* shall cover the number of way points indicated in *number-of-way-points* or the number of way points stored in the avionics, which ever is the lesser.

2.2.1.7.2.4.4 When *time-interval* was provided in the ADS-demand-contract indication, and *extended-projected-profile* is provided in the subsequent ADS-report request, the *extended-projected-profile* shall cover the time interval indicated in *time-interval* or the time interval covered by way points stored in the avionics, which ever is the lesser.

2.2.1.7.2.5 When the ADS-air-user invokes ADS-report request in response to an ADS-demand-contract indication, the *contract type* parameter shall be set to *demand-contract*, and the *event type* parameter not provided.

2.2.1.7.3 Establishment and operation of an Event Contract

Note 1.— 2.2.1.7.3 details the actions taken by the ADS-ground-user and the ADS-air-user in the establishment and operation of a event contract.

Note 2.— When the ADS-ground-user requires to establish an event contract with the ADS-air-user, it invokes ADS-event-contract request.

2.2.1.7.3.1 When invoking the ADS-event-contract request, the ADS-ground-user shall specify at least one event type.

2.2.1.7.3.2 When the ADS-air-user receives an ADS-event-contract indication, and it is not able to accept the contract, then the ADS-air-user shall invoke an ADS-event-contract response with the *reply* parameter set to *negative-acknowledgement* and *reason* set to the value indicating the reason that it cannot accept the contract.

Note.— In the event of the new event contract not being accepted, any existing event contract will remain in place.

2.2.1.7.3.3 When the ADS-air-user receives an ADS-event-contract indication, and it is able to accept the contract in full, the ADS-air-user shall:

- a) if the terms of the contract require an ADS-report as baseline information, and the ADS-air-user is able to invoke an ADS-report request within 0.5 seconds, then invoke an ADS-report request including a *positive acknowledgement* parameter, or
- b) if the terms of the contract do not require an ADS-report as baseline information, or the ADS-air-user is not able to invoke an ADS-report request within 0.5 seconds, then invoke ADS-event-contract response with *reply* set to *positive acknowledgement*.

2.2.1.7.3.4 When the ADS-air-user receives an ADS-event-contract indication, and it is able partially to fulfill the contract, because it is not able to detect some of the events in the contract, then the ADS-air-user shall invoke ADS-event-contract response with the *reply* parameter set to *noncompliance-notification*, and the *event-ncn* element set to the events that cannot be complied with.

2.2.1.7.3.5 If the ADS-air-user accepts the event contract with a *noncompliance-notification*, or with a *positive-acknowledgement* (either in an ADS-event-contract response or an ADS-report request) then the ADS-air-user shall:

- a) cancel any other event contract with that ground system, and
- b) if one or more of the following event types are in the ADS-event-contract indication and not present in the noncompliance notification if sent, then invoke ADS-report request with the contract type set to *event-report*, the event-type set to *baseline*, and air-vector and ground-vector included in the report details parameter:
 - 1) air-speed-change,
 - 2) ground-speed-change,
 - 3) heading-change,
 - 4) track-angle-change and/or
 - 5) level-change.

Note.— This provides a baseline reference against which possible deviations are compared.

2.2.1.7.3.5.1 Subject to the restrictions stated in 2.2.1.7.1.5.5, when *lateral-deviation-change* is provided in the ADS-event-contract *contract details* parameter, and not indicated in the noncompliance notification if sent, then for the duration of the event contract, only while the lateral deviation of the aircraft relative to the active route of flight is more than the value of *lateral-deviation-change*, the ADS-air-user shall invoke ADS-report requests at a rate of once every 60 seconds, including the *ground-vector* element in the *report details* parameter.

2.2.1.7.3.5.2 Subject to the restrictions stated in 2.2.1.7.1.5.5, when *vertical-rate-change* is provided in the ADS-event-contract *contract details* parameter with a zero or positive value, and not indicated in the noncompliance notification if sent, then for the duration of the event contract, only when the aircraft's rate of climb is greater than the value of *vertical-rate-change*, the ADS-air-user shall invoke ADS-report requests at a rate of once every 60 seconds, including the *ground-vector* element in the *report details* parameter.

2.2.1.7.3.5.3 Subject to the restrictions stated in 2.2.1.7.1.5.5, when *vertical-rate-change* is provided in the ADS-event-contract *contract details* parameter with a negative value, and not indicated in the noncompliance notification if sent, then for the duration of the event contract, only when the aircraft's rate of descent is greater than the absolute value of *vertical-rate-change*, the ADS-air-user shall invoke ADS-report requests at a rate of once every 60 seconds, including the *ground-vector* element in the *report details* parameter.

2.2.1.7.3.5.4 Subject to the restrictions stated in 2.2.1.7.1.5.5, when *level threshold* is provided in the ADS-event-contract *contract details* parameter, and not indicated in the noncompliance notification if sent, then for the duration of the event contract, only when the aircraft's level is greater than the value of *ceiling*, or less than the value of *floor*, the ADS-air-user shall invoke ADS-report requests at a rate of once every 60 seconds, including the *ground-vector* element in the *report details* parameter.

2.2.1.7.3.5.5 Subject to the restrictions stated in 2.2.1.7.1.5.5, when *way-point-change* is provided in the ADS-event-contract *contract details* parameter, and not indicated in the noncompliance notification if sent, then for the duration of the event contract, whenever the aircraft's next way-point changes, the ADS-air-user shall invoke ADS-report request, including the *projected-profile* element in the *report details* parameter.

2.2.1.7.3.5.6 Subject to the restrictions stated in 2.2.1.7.1.5.5, when *fom-change* is provided in the ADS-event-contract *contract details* parameter, then for the duration of the event contract, whenever the aircraft's navigational accuracy, navigational system redundancy or airborne collision avoidance system (ACAS) availability changes, the ADS-air-user shall invoke ADS-report request.

2.2.1.7.3.5.7 Subject to the restrictions stated in 2.2.1.7.1.5.5, when *extended-projected-profile-change* is provided in the ADS-event-contract *contract details* parameter, and contains the *time-interval* element, and is not indicated in the noncompliance notification if sent, then for the duration of the event contract, whenever one or more way-points on the active route of flight within the *time-interval* as measured from the current time changes, the ADS-air-user shall invoke ADS-report request including the *extended-projected-profile* element containing way-points covering the *time-interval* from the current time, or the time interval stored in the avionics, which ever is the lesser timer interval, in the ADS-report request *report details* parameter.

2.2.1.7.3.5.8 Subject to the restrictions stated in 2.2.1.7.1.5.5, when *extended-projected-profile-change* is provided in the ADS-event-contract *contract details* parameter, and contains the *number-of-way-points* element, and is not indicated in the noncompliance notification if sent, then for the duration of the event contract, whenever one or more way-points on the active route of flight that are in the next *number-of-way-points*, the ADS-air-user shall invoke ADS-report request including the *extended-projected-profile* element containing the next *number-of-way-points* or the number of way points stored in the avionics, which ever is the lesser.

2.2.1.7.3.5.9 Subject to the restrictions stated in 2.2.1.7.1.5.5, when *air-speed-change* is provided in the ADS-event-contract in the *contract details* parameter, and is not indicated in the noncompliance notification if sent, then for the duration of the event contract, whenever the absolute value of the difference between the aircraft's airspeed and the airspeed transmitted in the most recent ADS-report request that contained an air-vector element, is greater than or equal to the value of *air-speed-change*, then the ADS-air-user shall invoke ADS-report request including the *air-vector* element in the *report details* parameter.

2.2.1.7.3.5.10 Subject to the restrictions stated in 2.2.1.7.1.5.5, when *ground-speed-change* is provided in the ADS-event-contract in the *contract details* parameter, and is not indicated in the noncompliance notification if sent, then for the duration of the event contract, whenever the absolute value of the difference between the aircraft's ground speed and the ground speed transmitted in the most recent ADS-report request that contained a *ground-vector* element is greater than or equal to the value of *ground-speed-change*, then the ADS-air-user shall invoke ADS-report request including the *ground-vector* element in the *report details* parameter.

2.2.1.7.3.5.11 Subject to the restrictions stated in 2.2.1.7.1.5.5, when *track-angle-change* is provided in the ADS-event-contract in the *contract details* parameter, and is not indicated in the noncompliance notification if sent, then for the duration of the event contract, whenever the absolute value of the difference between the aircraft's track angle and the track angle transmitted in the most recent ADS-report request that contained a *ground-vector* element, is greater than or equal to the value of *track-angle-change*, then the ADS-air-user shall invoke ADS-report request including the *ground-vector* element in the *report details* parameter.

2.2.1.7.3.5.12 Subject to the restrictions stated in 2.2.1.7.1.5.5, when *level-change* is provided in the ADS-event-contract in the *contract details* parameter, and is not indicated in the noncompliance notification if sent, then for the duration of the event contract, whenever the absolute value of the difference between the aircraft's *level* and the level transmitted in the most recent ADS-report request, is greater than or equal to the value of *level-change*, then the ADS-air-user shall invoke ADS-report request including the *ground-vector* element in the *report details* parameter.

2.2.1.7.3.5.13 Subject to the restrictions stated in 2.2.1.7.1.5.5, when *heading-change* is provided in the ADS-event-contract *contract details* parameter, and not indicated in the noncompliance notification if sent, then for the duration of the event contract, whenever the aircraft's heading differs negatively or positively from the value transmitted in the previous ADS report containing an air-vector element by an amount exceeding the value of the *heading-change* element specified in the event contract request, then the ADS-air-user shall invoke ADS-report request including the *air-vector* element in the *report details* parameter.

2.2.1.7.3.5.14 If the ability of the aircraft to detect the occurrence of events changes during the event contract to the extent that it may effect the ability of the aircraft to meet the terms of the event contract, the ADS-air-user shall invoke ADS-report request including the *ability-to-detect-events-impaired* element in the *report details* parameter.

Note 1.— If more than one of the events described above occurs at the same time, the ADS-air-user invokes separate ADS-report requests as described above, for each event independently (i.e. the same report cannot be used to report on more than one event, even if the same information is being transmitted.)

Note 2.— Apart from circumstances detailed in 2.2.1.7.3.3, the positive acknowledgement parameter is not present in any ADS-report request made in response to an ADS-event-contract indication.

2.2.1.7.3.6 When the ADS-air-user invokes ADS-report request in response to an ADS-event-contract indication, the *contract type* parameter shall be set to *event-contract*.

2.2.1.7.3.7 When the ADS-air-user invokes ADS-report request in response to an ADS-event-contract indication, the *event type* parameter shall be set to indicate the type of event in the contract that this report is in response to, or to indicate a baseline report.

2.2.1.7.4 Establishment and operation of a Periodic Contract

Note 1.— 2.2.1.7.4 details the actions taken by the ADS-ground-user and the ADS-air-user in the establishment and operation of a periodic contract while no emergency contract exists.

Note 2.— When the ADS-ground-user requires to establish a periodic contract with the ADS-air-user it invokes ADS-periodic-contract request.

2.2.1.7.4.1 When the ADS-air-user receives an ADS-periodic-contract indication, and it is not able to accept the contract, then the ADS-air-user shall invoke an ADS-periodic-contract response with the *reply* parameter set to *negative-acknowledgement* and *reason* set to the value indicating the reason that it cannot accept the contract.

Note.— In the event of the new contract not being accepted, any existing contract will remain in place.

2.2.1.7.4.2 When the ADS-air-user receives an ADS-periodic-contract indication, and it is able to accept the contract in full, then:

2.2.1.7.4.2.1 If the ADS-air-user is able to send the first ADS-report request of the contract within 0.5 seconds, then the ADS-air-user shall invoke the first ADS-report request of the contract, including a *positive acknowledgement* parameter.

2.2.1.7.4.2.2 If the ADS-air-user is not able to send the first ADS-report request of the contract within 0.5 seconds, then:

2.2.1.7.4.2.2.1 The ADS-air-user shall:

- a) invoke ADS-periodic-contract response with the *reply* parameter set to *positive acknowledgement*, and
- b) if no emergency contract exists, send the first ADS-report request of the contract.

2.2.1.7.4.2.2.2 **Recommendation.**— *The ADS-air-user should:*

- a) *invoke ADS-periodic-contract response with the reply parameter set to positive acknowledgement within 0.5 seconds, and*
- b) *if no emergency contract exists, send the first ADS-report request of the contract within 30 seconds from the receipt of the ADS-periodic-contract request.*

2.2.1.7.4.3 When the ADS-air-user receives an ADS-periodic-contract indication, and it is able to supply some of the information required in the contract, but is not able generate all the report elements, or it is not able to meet the requested reporting rate, or both, then the ADS-air-user shall invoke ADS-periodic-contract response with the *reply* parameter set to *noncompliance-notification*, and with *periodic-ncn* set to indicate the reports that cannot be generated and/or that the reporting rate cannot be met.

2.2.1.7.4.4 If the ADS-air-user accepts the periodic contract with an ADS-periodic-contract response with the *Reply* parameter value set to *noncompliance-notification*, or *positive-acknowledgement*, or the ADS-air-user accepts the periodic contract with an ADS-report with the *Positive acknowledgement* parameter present then:

2.2.1.7.4.4.1 The ADS-air-user shall cancel any periodic contract in force with the ground system.

2.2.1.7.4.4.2 If the ADS-air-user accepted the contract with a *noncompliance-notification* that indicated that the reporting rate could not be met, then the ADS-air-user shall set the reporting rate to be 60 seconds.

2.2.1.7.4.4.3 If the ADS-air-user accepted the contract by a means other than a *noncompliance-notification* that indicated that the reporting rate could not be met, then the ADS-air-user shall set the reporting rate to be the *reporting-interval* from the *contract details* parameter of the ADS-periodic-contract indication.

2.2.1.7.4.4.4 The ADS-air-user shall invoke ADS-report requests at the reporting rate, until such time as the contract is cancelled, or suspended due to an emergency.

Note.— If an emergency contract is already in place, the periodic contract will be immediately suspended due to the provisions stated in 2.2.1.7.

2.2.1.7.4.4.5 Subject to the restrictions stated in 2.2.1.7.1.5.5, the ADS-air-user invokes ADS-report request in response to a ADS-periodic-contract indication, then, for each row in Table 2.2.1.7-1 :

- a) if the modulus is present in the *contract details* parameter of the ADS-periodic-contract indication;
- b) if the ADS-air-user did not accept the contract by means of a *noncompliance-notification* that indicated that it is not able to generate that report element; and
- c) if the number of ADS-report requests already invoked in response to this contract is exactly divisible by the value of the modulus parameter.

then the ADS-air-user shall include the *report details* element as indicated in Table 2.2.1.7-1.

Table 2.2.1.7-1: Inclusion of optional report details in ADS-reports

Modulus in the Contract Details Parameter	ADS-report Report Details Element
aircraft address modulus	aircraft address
projected-profile-modulus	projected-profile
ground-vector-modulus	ground-vector
air-vector-modulus	air-vector
weather-modulus	weather
short-term-intent-modulus	short-term-intent
extended-projected-profile-modulus	extended-projected-profile

Note 1.— For example, if aircraft address has the value 2 and ground-vector-modulus has the value 3, then the aircraft address element will be included in the 1st, 3rd, 5th, 7th etc. ADS-reports, and the ground-vector will be included in the 1st, 4th, 7th, 10th etc. ADS-reports.

Note 2.— Position, time-stamp and fom will always be included in every ADS-report.

2.2.1.7.4.4.6 When short-term-intent is included in the *report details* parameter of an ADS-report, the ADS-air-user shall insert a value that covers way points and estimated times of arrival for the following *intent-projection-time* as measured from the timestamp on the ADS-report.

2.2.1.7.4.4.7 When *number-of-way-points* was provided in the *extended-projected-profile-modulus* of the ADS-periodic-contract indication, and *extended-projected-profile* is provided in the subsequent ADS-report request, the *extended-projected-profile* shall cover the number of way points indicated in *number-of-way-points* or the number of way points stored in the avionics, which ever is the lesser.

2.2.1.7.4.4.8 When *time-interval* was provided in the *extended-projected-profile-modulus* of the ADS-periodic-contract indication, and *extended-projected-profile* is provided in the subsequent ADS-report request, the *extended-projected-profile* shall cover the time interval indicated in *time-interval* or the time interval covered by way points stored in the avionics, which ever is the lesser.

2.2.1.7.4.4.9 When the ADS-air-user invokes ADS-report request in response to an ADS-periodic-contract indication, the *contract type* parameter shall be set to *periodic-contract*.

Note 1.— Apart from circumstances detailed in 2.2.1.7.4.2.1, the positive acknowledgement parameter is not present in the ADS-report request.

Note 2.— When the ADS-air-user invokes ADS-report request in response to an ADS-periodic-contract indication, the event type parameter is not be included in the ADS-report request.

2.2.1.7.5 Ground Cancellation of Contracts

Note 1.— 2.2.1.7.5 details the actions taken by the ADS-ground-user and the ADS-air-user in the cancellation of contracts.

Note 2.— When an ADS-ground-user requires to cancel an event contract or a periodic contract, then it either invokes ADS-cancel request with the contract type parameter set to event-contract or periodic-contract respectively. When an ADS-ground-user requires to cancel all contracts with the aircraft it invokes ADS-cancel-all-contracts request

2.2.1.7.5.1 If the ADS-air-user receives an ADS-cancel-contract with *contract type* parameter set to *event-contract*, the ADS-air-user shall cancel any event contract with that ground system.

2.2.1.7.5.2 If the ADS-air-user receives an ADS-cancel-contract with *contract type* parameter set to *periodic-contract*, the ADS-air-user shall cancel any periodic contract with that ground system.

2.2.1.7.5.3 When the ADS-air-user receives an ADS-cancel-all-contracts indication, it shall cancel all contracts (event, periodic and emergency) with that ground system.

Note.— There is no provision for cancellation of demand contracts.

2.2.1.7.6 Establishment and Operation of Emergency Contracts

Note 1.— 2.2.1.7.6 details the actions taken by the ADS-ground-user and the ADS-air-user in the establishment and operation of emergency contracts.

Note 2.— The emergency contract is only air user activated, and may be initiated either by human or automatically by the aircraft system.

2.2.1.7.6.1 On emergency contract initiation, the ADS-air-user shall establish an emergency contract with every ground system with which it has an event contract or a periodic contract (or both).

2.2.1.7.6.2 When an ADS-periodic-contract indication or ADS-event-contract indication occurs during an emergency, from an ADS-ground-user with which the aircraft has not got an event contract or a periodic contract, then the ADS-air-user shall:

- a) acknowledge the contract in the manner indicated in 2.2.1.7.3 and 2.2.1.7.4, with either a response or an ADS-report request, and
- b) if a negative acknowledgement is not sent, establish an emergency contract with the ADS-ground-user.

2.2.1.7.6.3 When the ADS-air-user establishes an emergency contract with an ADS-ground-user, then:

2.2.1.7.6.3.1 If the ADS-air-user has a periodic contract with the ADS-ground-user, then the ADS-air-user shall suspend the operation of the periodic contract.

2.2.1.7.6.3.2 If the ADS-air-user has no periodic contract with the ADS-ground-user at the time of establishing the emergency contract, then the ADS-air-user shall set the emergency reporting rate to be 60 seconds.

2.2.1.7.6.3.3 If the ADS-air-user has a periodic contract with the ADS-ground-user at the time of establishing the emergency contract, then the ADS-air-user shall set the emergency reporting rate to be as indicated in Table 2.2.1.7-2.

Table 2.2.1.7-2: Emergency reporting rate calculation when a periodic contract is in place

Existing periodic reporting rate	Emergency reporting rate
1 second	1 second
greater than two minutes	60 seconds
less than or equal to two minutes and greater than 1 second	half the reporting rate of the periodic contract rounded down to the nearest second

2.2.1.7.6.3.4 The ADS-air user shall invoke ADS-emergency-report request at the emergency reporting rate.

2.2.1.7.6.3.5 Subject to the restrictions stated in 2.2.1.7.1.5.5, the ADS-air user shall include the following elements in the emergency report details parameter in the ADS-emergency-report request:

- a) *position, timestamp* and *fom*; and
- b) *ground-vector* and *aircraft address* for the first ADS-emergency-report request after the emergency contract has been established, and subsequently every fifth ADS-emergency-report request.

Note 1.— That is, on the 1st, 6th, 11th, 16th etc. ADS-emergency-report request.

Note 2.— Apart from conditions specified in 2.2.1.7.7, the positive acknowledgement parameter is not present in any ADS-emergency-report request.

2.2.1.7.6.3.6 If the ADS-air-user receives an ADS-periodic-contract indication from an ADS-ground-user with which the ADS-air-user has an emergency contract, then the ADS-air-user shall invoke ADS-periodic-contract response giving a reply that would be appropriate if the emergency contract were not in operation.

Note.— This implies that a reply of an ADS-report is not possible.

2.2.1.7.6.3.7 For each ground system, the ADS-air-user shall store details of the most recent of the following:

- a) ADS-periodic-contract indication, which had a positive acknowledgement as a response;
- b) ADS-periodic-contract indication, which had a noncompliance notification as a response; and
- c) ADS-cancel-contract indication with value *periodic-contract*.

Note.— This information is used to re-establish the periodic contract after the emergency is over.

2.2.1.7.7 Modifying an Emergency Contract

Note 1.— 2.2.1.7.7 details the actions taken by the ADS-ground-user and the ADS-air-user when modifying an emergency contract.

Note 2.— When the ADS-ground-user requires to modify the reporting rate of an emergency contract it invokes ADS-modify-emergency-contract request.

2.2.1.7.7.1 When the ADS-air-user receives an ADS-modify-emergency-contract indication, and it is able to comply with the request, it shall:

- a) change the emergency reporting rate to the time indicated in the *reporting interval* parameter; and
- b) include a *positive acknowledgement* parameter in the next ADS-emergency-report request.

Note 1.— The existing five ADS-emergency-report cycle remains regardless of any reporting rate modification, moreover, the position within the cycle also remains unaffected. For example, if the second ADS-emergency-report request was invoked before the modification of emergency reporting rate, the following ADS-emergency-report request will be the third.

2.2.1.7.7.1.1 **Recommendation.**— *The ADS-air-user should invoke the next ADS-emergency-report request within 0.5 seconds.*

2.2.1.7.7.2 When the ADS-air-user receives an ADS-modify-emergency-contract indication, and it is not able comply with the request, then the ADS-air-user shall invoke ADS-modify-emergency-contract response.

2.2.1.7.7.2.1 **Recommendation.**— *The ADS-air-user should invoke the next ADS-emergency-report request within 0.5 second.*

Note.— *The emergency reporting rate remains unchanged.*

2.2.1.7.8 Cancellation of an Emergency Contract

Note 1.— *2.2.1.7.8 details the actions taken by the ADS-ground-user and the ADS-air-user when the ADS-air-user cancels emergency contracts.*

Note 2.— *The initiation of the cancellation of an emergency contract may only be done by human intervention in the aircraft.*

2.2.1.7.8.1 When the ADS-air-user cancels emergency contracts, it shall cancel the emergency contract with each ADS-ground-user with which it has an emergency contract.

2.2.1.7.8.2 When the ADS-air-user cancels an emergency contract, it shall:

- a) invoke ADS-cancel-emergency request,
- b) if a periodic contract was in operation before the emergency contract was established and no ADS-periodic-contract indications were accepted and no ADS-cancel-contract indications (with a value of periodic contract) were received during the emergency contract, then resume operation of the periodic contract, and
- c) if the latest event to be stored (as indicated in 2.2.1.7.6.3) was an ADS-periodic-contract indication, then initiate the operation of that periodic contract.

Note.— *If the latest event to be stored (as indicated in 2.2.1.7.6.3) was an ADS-cancel-contract indication (with a value of periodic-contract) then no periodic contract is started or resumed.*

2.2.1.7.8.2.1 **Recommendation.**— *If the ADS-air-user reinstates a periodic contract or initiates a new periodic contract, it should invoke the next ADS-report request within 0.5 second.*

2.2.1.7.8.2.2 If the ADS-air-user reinstates a periodic contract, it shall restart in the same position in the cycle of reports as it was in when the emergency contract was established.

2.2.1.7.9 Operation of Aborts

Note 1.— *2.2.1.7.9 details the actions taken by an ADS-ground-user and the ADS-air-user aborts occur.*

Note 2.— *When an ADS-ground-user or an ADS-air-user requires to abort the current contracts, it initiates ADS-user-abort request.*

2.2.1.7.9.1 When the ADS-air-user or the ADS-ground-user receives an ADS-user-abort indication or an ADS-provider-abort-indication, it shall cancel all contracts with the peer ADS-user.

2.2.1.7.10 Parameter Value Unit, Range and Resolution

2.2.1.7.10.1 An ADS user shall interpret ADS parameter value unit, range and resolution as defined in 2.2.1.4.

2.2.1.8 Subsetting Rules

2.2.1.8.1 General

Note.— 2.2.1.8 specifies conformance requirements which all implementations of the ADS protocol obey.

2.2.1.8.1.1 An implementation of either the ADS ground based service or the ADS air based service claiming conformance to 2.2.1 shall support the ADS protocol features as shown in the tables below.

Note.— The ‘status’ column indicates the level of support required for conformance to the ADS-ASE protocol described in 2.2.1. The values are as follows:

- a) **‘M’** mandatory support is required;
- b) **‘O’** optional support is permitted for conformance to the ADS protocol;
- c) **‘N/A’** the item is not applicable; and
- d) **‘C.n’** the item is conditional where *n* is the number which identifies the condition which is applicable.

Table 2.2.1.8-1. ADS Protocol Versions Implemented

	Status	Associated Predicate
Version 1	M	none

Table 2.2.1.8-2. ADS Protocol Functional Units

	Status	Associated Predicate
The ADS system acts as an airborne system	C.1	ADS/air
The ADS system acts as a ground system	C.1	ADS/ground
The ADS ground system can establish demand contract	If (ADS/ground) C.2 else N/A	G-DC-FU

	Status	Associated Predicate
The ADS ground system can establish event contracts and process emergency reports	If (ADS/ground) C.2 else N/A	G-EC-FU
The ADS ground system can establish periodic contracts and process emergency reports	If (ADS/ground) C.2 else N/A	G-PC-FU
The ADS air system can process demand contracts	If (ADS/air) C.3 else N/A	A-DC-FU
The ADS air system can process event contracts	If (ADS/air) C.3 else N/A	A-EC-FU
The ADS air system can process periodic contracts	If (ADS/air) C.3 else N/A	A-PC-FU
The ADS air system can send emergency reports	If (A-EC-FU or A-PC-FU) O else N/A	A-EM-FU

C.1: a conformant implementation shall support one and only one of these two options.

C.2: a conformant ground implementation shall support at least one of the three options.

C.3: a conformant air implementation shall support at least one of the three options.

Table 2.2.1.8-3. ADS-ground-ASE Conformant Configurations

	List of Predicates	Functionality Description
I	G-DC-FU + ADS/ground	ADS-ground-ASE supporting demand contract only Demand contract only can be established with the aircraft
II	G-EC-FU + ADS/ground	ADS-ground-ASE supporting event and emergency contracts Event and Emergency contracts can be established with the aircraft
III	G-PC-FU + ADS/ground	ADS-ground-ASE supporting periodic and emergency contracts Periodic and Emergency contracts can be established with the aircraft
IV	G-DC-FU + G-EC-FU + ADS/ground	ADS-ground-ASE supporting demand, event and emergency contracts Demand, Event and Emergency contracts can be established with the aircraft

	List of Predicates	Functionality Description
V	G-DC-FU + G-PC-FU + ADS/ground	ADS-ground-ASE supporting demand, periodic and emergency contracts Demand, Periodic and Emergency contracts can be established with the aircraft
VI	G-EC-FU + G-PC-FU + ADS/ground	Event, Periodic and Emergency contracts can be established with the aircraft
VII	G-DC-FU + G-EC-FU + G-PC-FU + ADS/ground	Demand, Event, Periodic and Emergency contracts can be established with the aircraft

Note.— An ADS ground system may or may not support the modify emergency capability.

Table 2.2.1.8-4. ADS-air-ASE Conformant Configurations

	List of Predicates	Functionality Description
I	ADS/air + A-DC-FU	Demand contracts only can be established with the ground system. A Negative Acknowledgement with reason “ADS-service unavailable” is sent when Event or Periodic contracts are requested.
II	ADS/air + A-EC-FU	Event contracts only can be established with the ground system. A Negative Acknowledgement with reason “ADS-service unavailable” is sent when contracts are requested.
III	ADS/air + A-PC-FU	Periodic contracts only can be established with the ground system. A Negative Acknowledgement with reason “ADS-service unavailable” is sent when Demand or Event contracts are requested.
IV	ADS/air + A-DC-FU + A-EC-FU	Demand and Event contracts can be established with the ground system. A Negative Acknowledgement with reason “ADS-service unavailable” is sent when Periodic contracts are requested.
V	ADS/air + A-DC-FU + A-PC-FU	Demand and Periodic contracts can be established with the ground system. A Negative Acknowledgement with reason “ADS-service unavailable” is sent when Event contracts are requested.

	List of Predicates	Functionality Description
VI	ADS/air + A-EC-FU + A-PC-FU	Event and Periodic contracts can be established with the ground system. A Negative Acknowledgement with reason “ADS-service unavailable” is sent when Demand contracts are requested.
VII	ADS/air + A-DC-FU + A-EC-FU + A-PC-FU	Demand, Event and Periodic contracts can be established with the ground system.
VIII	ADS/air + A-EC-FU + A-EM-FU	Event and Emergency contracts can be established with the ground system. A Negative Acknowledgement with reason “ADS-service unavailable” is sent when Demand and Periodic contracts are requested.
IX	ADS/air + A-PC-FU + A-EM-FU	Periodic and Emergency contracts can be established with the ground system. A Negative Acknowledgement with reason “ADS-service unavailable” is sent when Demand and Event contracts are requested.
X	ADS/air + A-DC-FU + A-EC-FU + A-EM-FU	Demand, Event and Emergency contracts can be established with the ground system. A Negative Acknowledgement with reason “ADS-service unavailable” is sent when Periodic contracts are requested.
XI	ADS/air + A-DC-FU + A-PC-FU + A-EM-FU	Demand, Periodic and Emergency contracts can be established with the ground system. A Negative Acknowledgement with reason “ADS-service unavailable” is sent when Event contracts are requested.
XII	ADS/air + A-EC-FU + A-PC-FU + A-EM-FU	Event, Periodic and Emergency contracts can be established with the ground system. A Negative Acknowledgement with reason “ADS-service unavailable” is sent when Demand contracts are requested.
XIII	ADS/air + A-DC-FU + A-EC-FU + A-PC-FU + A-EM-FU	Demand, Event, Periodic and Emergency contracts can be established with the ground system.

2.2.2 AUTOMATIC DEPENDENT SURVEILLANCE REPORT FORWARDING APPLICATION

2.2.2.1 Introduction

2.2.2.1.1 The ADS report forwarding application will allow users to obtain positional and other information from suitably equipped aircraft in a timely manner in accordance with their requirements.

Note 1.— Structure of 2.2.2

2.2.2 defines the ground-ground aspects of ADS only. 2.2.1 defines the air-ground communication aspects of ADS:

- a) *2.2.2.1: INTRODUCTION contains the 2.2.2 's purpose, structure, and a summary of the functions of ADS.*
- b) *2.2.2.2: GENERAL REQUIREMENTS contains backwards compatibility and error processing requirements.*
- c) *2.2.2.3: THE ABSTRACT SERVICE contains the description of the abstract service provided by the application service elements (ASE) defined for ADS Report Forwarding.*
- d) *2.2.2.4: FORMAL DEFINITION OF MESSAGES contains the formal definition of messages exchanged by ADS-RF-ASEs using Abstract Syntax Notation Number One (ASN.1).*
- e) *2.2.2.5: PROTOCOL DEFINITION describes the exchanges of messages allowed by the ADS protocol, as well as time constraints and the exception handling procedures associated with these exchanges. 2.2.2.5 describes also the ADS protocol related to report forwarding in terms of state tables.*
- f) *2.2.2.6: COMMUNICATION REQUIREMENTS contains the requirements that the ADS-RF-ASEs imposes on the underlying communication system.*
- g) *2.2.2.7: ADS USER REQUIREMENTS outlines the requirements that a user of an ADS-RF-ASE must meet.*
- h) *2.2.2.8: SUBSETTING RULES provides rules for subsetting the ADS Report Forwarding SARPs.*

Note 2.— General Functionality

- a) *It will be necessary for an implementation to provide information which is both accurate and timely in the ADS reports; however, quantification of the age and accuracy of the information is beyond the scope of 2.2.2.*

*Note 3.— Establishment and Operation of Forward Contract**a) Functional Description*

- 1) This function provides a method for a ground system to establish a forward contract with another ground system and to forward ADS reports. This function is initiated by a ground system having received ADS reports, which then forwards the received ADS reports to another ground system.*
- 2) The receiving ground system may reject the ADS start forwarding request.*
- 3) When an ADS report is to be sent the ground system will use an ADS forward report message.*

b) Message Descriptions

- 1) The ADS start forwarding request message may contain the first ADS forward report.*
- 2) The ADS start forwarding response contains the result of the establishment of the forward contract.*
- 3) An ADS forward report message contains the aircraft address and flight identification of the aircraft the report is related to, and either a periodic, event, demand, or emergency report.*

*Note 4.— Cancellation of the Forward Contract**a) Functional Description*

- 1) This function allows the sending ground system to cancel the Forward Contract.*
- 2) The sending ground system sends a cancel forward contract message to the receiving ground system.*

b) Message Descriptions

- 1) The cancel forward contract message does not contain any information.*

2.2.2.2 General Requirements**2.2.2.2.1 ADS-RF-ASE Version Number**

2.2.2.2.1.1 The ADS-RF-ASE version number shall be set to one.

2.2.2.2.2 Error Processing Requirements

2.2.2.2.2.1 In the event of information input by the ARF-user being incompatible with that able to be processed by the system, the ARF-user shall be notified.

2.2.2.2.2.2 In the event of a ARF-user invoking an ADS Report Forwarding service primitive, when the ADS-RF-ASE is not in a state specified in 2.2.2.5, the ARF-user shall be notified.

2.2.2.3 The Abstract Service

2.2.2.3.1 Service Description

2.2.2.3.1.1 An implementation of the ADS Report Forwarding service shall exhibit external behaviour consistent with having implemented an ADS-RF-ASE.

Note 1.— 2.2.2.3 defines the abstract service interface for the ADS Report Forwarding service. The ADS-RF-ASE abstract service is described in 2.2.2.3 from the viewpoint of the ADS-RF-user and the ADS-RF service-provider.

Note 2.— 2.2.2.3 defines the static behaviour (i.e. the format) of the ADS Report Forwarding abstract service. Its dynamic behaviour (i.e. how it is used) is described in 2.2.2.7.

Note 3.— Figure 2.2.2.3-1 shows the functional model of the ADS Report Forwarding Function. The functional modules identified in this model are the following :

- a) *the ADS Report Forwarding user,*
- b) *the ADS Report Forwarding Application Entity (ADS-RF-AE) service interface,*
- c) *the ADS-RF-AE,*
- d) *the ADS Report Forwarding Control Function (ADS-RF-CF),*
- e) *the ADS Report Forwarding Application Service Element (ADS-RF-ASE) service interface,*
- f) *the ADS-RF-ASE, and*
- g) *the Dialogue Service (DS) interface.*

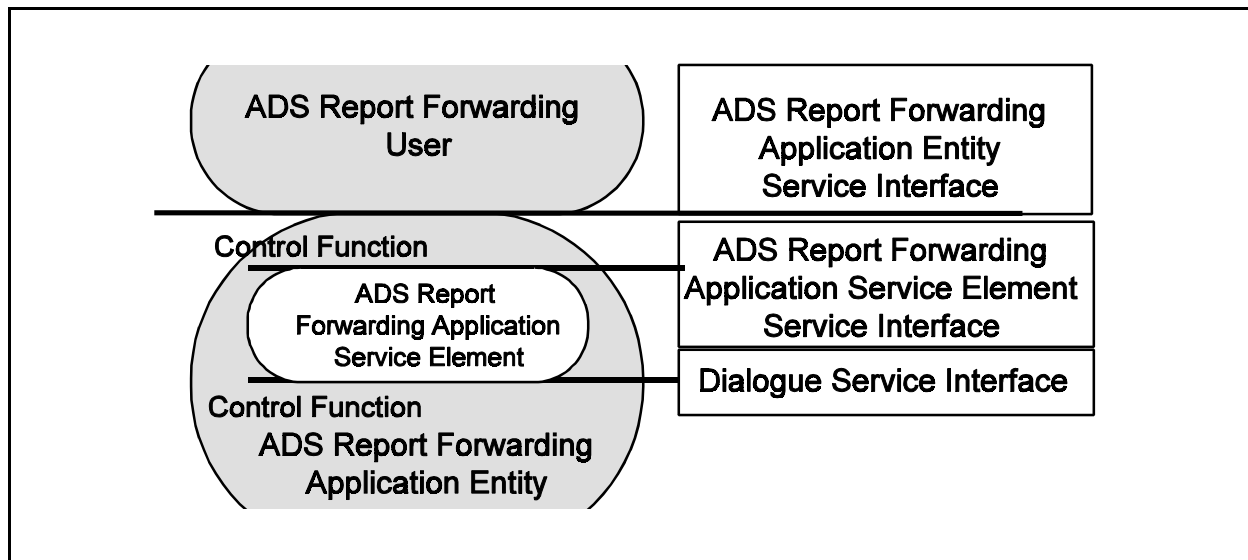


Figure 2.2.2.3-1. Functional Model of the ADS Report Forwarding Function

Note 4.— The ADS-RF-user represents the operational part of the ADS system. This user does not perform the communication functions but relies on a communication service provided to it via the ADS-RF-AE through the ADS-RF-AE service interface. The individual actions at this interface are called ADS-RF-AE service primitives. Similarly, individual actions at other interfaces in the communication system are called service primitives at these interfaces.

Note 5.— The ADS-RF-AE consists of several elements, including the ADS-RF-ASE and the ADS-RF-CF. The DS interface is made available by the ADS-RF-CF to the ADS-RF-ASE for communication with the peer ADS-RF-ASE.

Note 6.— The ADS-RF-ASE is the element in the communication system which executes the ADS-RF specific protocol. In other words, it takes care of the ADS-RF specific service primitive sequencing actions, message creation, timer management, error and exception handling.

Note 7.— The ADS-RF-ASE interfaces only with the ADS-RF-CF. This ADS-RF-CF is responsible for mapping service primitives received from one element (such as the ADS-RF-ASE and the ADS-RF-user) to other elements which interface with it. The part of the ADS-RF-CF which is relevant from the point of view of these SARPs, i.e. the part between the ADS-RF-user and the ADS-RF-ASE, will map ADS-RF-AE service primitives to ADS-RF-ASE service primitives transparently.

Note 8.— The DS interface is the interface between the ADS-RF-ASE and part of ADS-RF-CF underneath, the ADS-RF-ASE and provides the dialogue service.

2.2.2.3.2 The ADS-RF-ASE Abstract Service

Note.— There is no requirement to implement the service in an ADS product; however, it is necessary to implement the ground based system in such a way that it will be impossible to detect (from the peer system) whether or not an interface has been built.

2.2.2.3.2.1 The ADS-RF-ASE abstract service shall consist of a set of the following services as allowed by the subsetting rules defined in 2.2.2.8:

- a) ADS-start-forward service as defined in 2.2.2.3.4;
- b) ADS-forward-report service as defined in 2.2.2.3.5;
- c) ADS-end-forward service as defined in 2.2.2.3.6;
- d) ADS-user-abort service as defined in 2.2.2.3.7;
- e) ADS-provider-abort service as defined in 2.2.2.3.8.

Note.— An abstract syntax is a syntactical description of a parameter which does not imply a specific implementation. Only when the ADS-RF-ASE maps a parameter onto an APDU field, or vice-versa, is the abstract syntax of the parameter described by using the ASN.1 of 2.2.2.4 for this field.

2.2.2.3.3 Conventions

Note 1.— For a given primitive, the presence of each parameter is described by one of the following values in the parameter tables 2.2.2.3:

- a) *blank* not present;
- b) *C* conditional upon some predicate explained in the text;
- c) *C(=)* conditional upon the value of the parameter to the immediate left being present, and equal to that value;
- d) *M* mandatory;
- e) *M(=)* mandatory, and equal to the value of the parameter to the immediate left;
- f) *U* user option.

Note 2.— The following abbreviations are used in this 2.2.2.3:

- a) *Req* request; data is input by an ADS-RF-user initiating the service to its respective ASE;
- b) *Ind* indication; data is indicated by the receiving ASE to its respective ADS-RF-user;

- c) *Rsp response; data is input by receiving ADS-RF-user to its respective ASE;*
- d) *Cnf confirmation; data is confirmed by the initiating ASE to its respective ADS-RF-user.*

Note 3.— An unconfirmed service allows just one message to be transmitted, in one direction,.

Note 4.— A confirmed service provides end-to-end confirmation that a message sent by one user was received by its peer user.

2.2.2.3.4 ADS-start-forward Service

Note.— The ADS-start-forward service allows an ADS-RF-user to request the establishment of a forward contract with another ADS-RF-user. An ADS report may be included within this service. It is a confirmed service, initiated by an ADS-RF-user.

2.2.2.3.4.1 The ADS-start-forward service shall contain primitives and parameters as presented in Table 2.2.2.3-1 where the version numbers of the peer ASEs are compatible.

2.2.2.3.4.2 The ADS-start-forward service shall contain primitives and parameters as presented in Table 2.2.2.3-2 where the version numbers of the peer ASEs are incompatible.

Table 2.2.2.3-1. ADS-start-forward service parameters - compatible version numbers

Parameter Name	Req	Ind	Cnf
ICAO Facility designation	M		
Class of communication service	U		
Forwarded report details	U	C(=)	
Reply			M

Table 2.2.2.3-2. ADS-start-forward service parameters - incompatible version numbers

Parameter Name	Req	Cnf
ICAO Facility designation	M	
Class of communication service	U	
Forwarded report details	U	
Reply		M

2.2.2.3.4.3 ICAO Facility designation

Note.— This parameter contains the receiving ground system's ICAO facility designation.

2.2.2.3.4.3.1 The *ICAO Facility designation* parameter value shall conform to the abstract syntax four-to eight-character ICAO facility designation.

2.2.2.3.4.4 Class of Communication Service

Note.— This parameter contains the value of the required class of communication service, if specified by the ADS-RF-user.

2.2.2.3.4.4.1 Where specified by the ADS-RF-user, the *class of communication service* parameter shall have one of the following abstract values: “A”, “B”, “C”, “D”, “E”, “F”, “G” or “H”.

Note.— Where not specified by the ADS-RF-user, this indicates that there will be no routing preference.

2.2.2.3.4.5 Forwarded Report Details

Note.— This parameter contains the details of the forwarded ADS report.

2.2.2.3.4.5.1 The *forwarded report details* parameter value shall conform to the ASN.1 abstract syntax *ADSForwardedReport*.

2.2.2.3.4.6 Reply

Note.— This parameter indicates whether the ADS-start-forward request has been accepted (abstract value is “accepted”) or rejected (abstract value is “incompatible version”) by the peer ADS-RF-user.

2.2.2.3.4.6.1 The *Reply* parameter value shall have one of the following abstract values:

- a) “accepted”, or
- b) “incompatible version”.

2.2.2.3.5 ADS-forward-report Service

Note.— The ADS-forward-report service allows an ADS-RF-user to forward an ADS report to another ADS-RF-user. This is an unconfirmed service, initiated by the ADS-RF-user which has initiated the ADS-start-forward service.

2.2.2.3.5.1 The ADS-forward-report service shall contain primitives and parameters as contained in Table 2.2.2.3-3.

Table 2.2.2.3-3. ADS-forward-report service parameters

Parameter Name	Req	Ind
Forwarded Report details	M	M(=)

2.2.2.3.5.2 Forwarded Report Details

Note.— This parameter contains the details of the forwarded ADS report.

2.2.2.3.5.2.1 The *forwarded report details* parameter value shall conform to the ASN.1 abstract syntax *ADSForwardedReport*.

2.2.2.3.6 ADS-end-forward Service

Note.— The ADS-end-forward service allows the ADS-RF-user forwarding the ADS reports to end the ADS Report Forwarding service. It is a unconfirmed service, initiated by the sending ADS-RF-user.

2.2.2.3.6.1 The ADS-end-forward service shall contain primitives as contained in Table 2.2.2.3-4.

Table 2.2.2.3-4. ADS-end-forward service parameters

Parameter Name	Req	Ind
none		

2.2.2.3.7 ADS-user-abort Service

Note 1.— The ADS-user-abort service allows the ADS-RF-user to abort a forward contract. It is an unconfirmed service, initiated by an ADS-RF-user. Messages in transit may be lost during this operation. It can be invoked at any time that the ADS-RF-user is aware that any ADS Report Forwarding service is in operation.

Note 2.— If the service is invoked prior to complete establishment of the dialogue, the ADS-user-abort indication may not be provided. An ADS-provider-abort indication may result instead.

2.2.2.3.7.1 The ADS-user-abort service shall contain primitives as contained in Table 2.2.2.3-5.

Table 2.2.2.3-5. ADS-user-abort service parameters

Parameter Name	Req	Ind
none		

2.2.2.3.8 ADS-provider-abort Service

Note.— The ADS-provider-abort service allows the ADS-service-provider to inform the ADS-RF-users that it can no longer provide the ADS Report Forwarding service for a particular ADS-RF-user pairing. It is initiated by the ADS-service-provider. Messages in transit may be lost during this operation.

2.2.2.3.8.1 The ADS-provider-abort service shall contain primitives and parameters as contained in Table 2.2.2.3-6.

Table 2.2.2.3-6. ADS-provider-abort service parameters

Parameter Name	Ind
Reason	M

2.2.2.3.8.2 Reason

Note.— This parameter identifies the reason for the abort.

2.2.2.3.8.2.1 The *reason* parameter shall conform to the ASN.1 abstract syntax *AbortReason*.

2.2.2.4 Formal Definitions of Messages**2.2.2.4.1 Encoding/Decoding Rules**

2.2.2.4.1.1 An ADS-RF-ASE shall be capable of encoding and decoding [ADSRFPDUs] APDUs.

2.2.2.4.2 ADS ASN.1 Abstract Syntax

2.2.2.4.2.1 The abstract syntax of the ADS-RF protocol data units shall comply with the description contained in the ASN.1 module ADSRFMessageSetVersion1 (conforming to ISO/IEC 8824), as defined in 2.2.2.4.

ADSRFMessageSetVersion1 DEFINITIONS ::=

BEGIN

IMPORTS

AbortReason, ADSEmergencyReport, ADSReport, AircraftAddress, EventTypeReported
FROM ADSMessageSetVersion1;

ADSRFPDUs ::= CHOICE

{	aDS-forwarded-report-PDU	[0]	ADSForwardedReport,
	aDS-provider-abort-PDU	[1]	AbortReason,

...

}

ADSForwardedReport ::= SEQUENCE

{	aircraftAddress	AircraftAddress,
	forwardedADSReport	ForwardedReport

}

ForwardedReport ::= CHOICE

{	aDSDemandReport	[0]	ADSReport,
	aDSPeriodicReport	[1]	ADSReport,
	aDSEventReport	[2]	SEQUENCE
		{	
		event-type	EventTypeReported,
		aDSReport	ADSReport
		}	
	aDSEmergencyReport	[3]	ADSEmergencyReport

}

END -- of ADSRFMessageSetVersion1

2.2.2.5 Protocol Definition

2.2.2.5.1 Sequence Rules

2.2.2.5.1.1 Only the sequence of primitives defined illustrated in figures 2.2.2.5-1 to 2.2.2.5-6 shall be permitted.

Note 1.— The following figures define the valid sequences of primitives that are possible to be invoked during the operation of the ADS Report Forwarding function. They show the relationship in time between the service request and the resulting indication, and if applicable, the subsequent response and the resulting confirmation.

Note 2.— Abort primitive may interrupt and terminate any of the normal message sequences outlined below.

Note 3.— Primitives are processed in the order in which they are received (see 4.3.3.1.2.4).

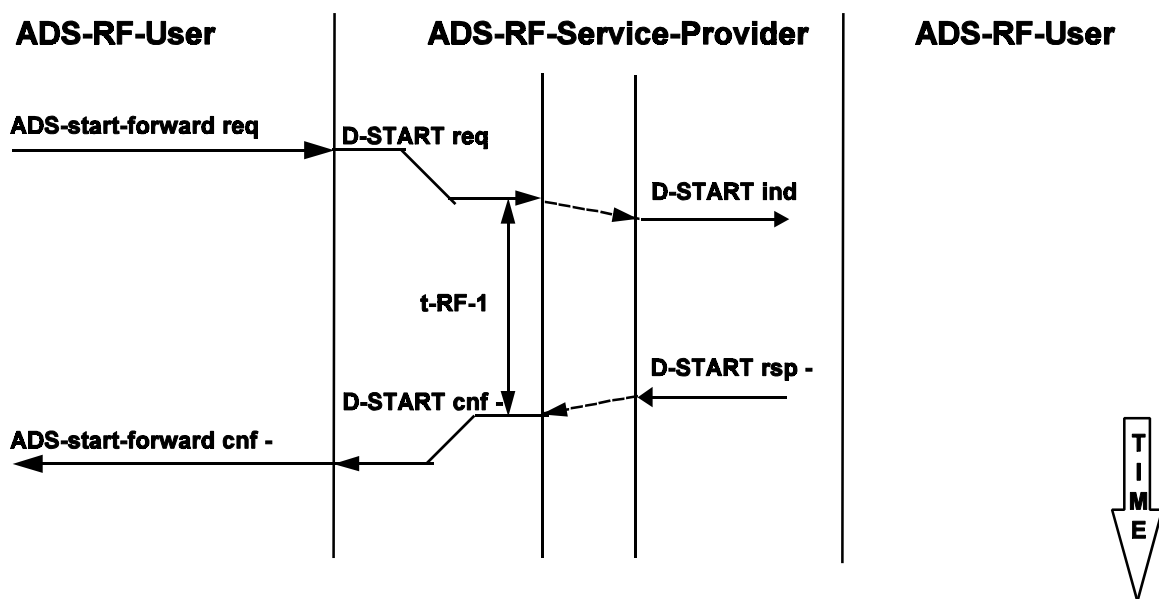


Figure 2.2.2.5-1. Use of forward contract with negative response

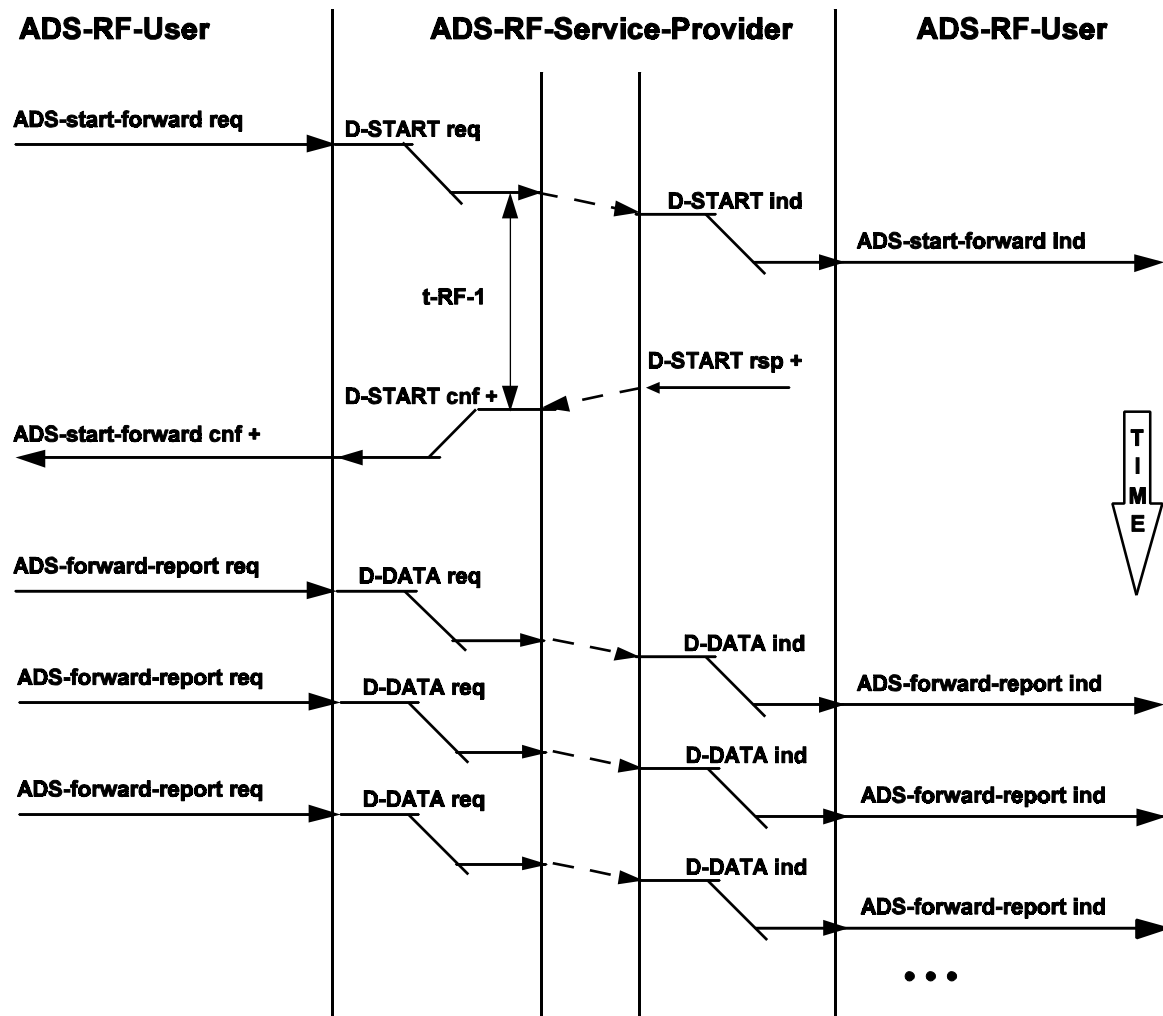


Figure 2.2.2.5-2. Use of forward contract with positive response

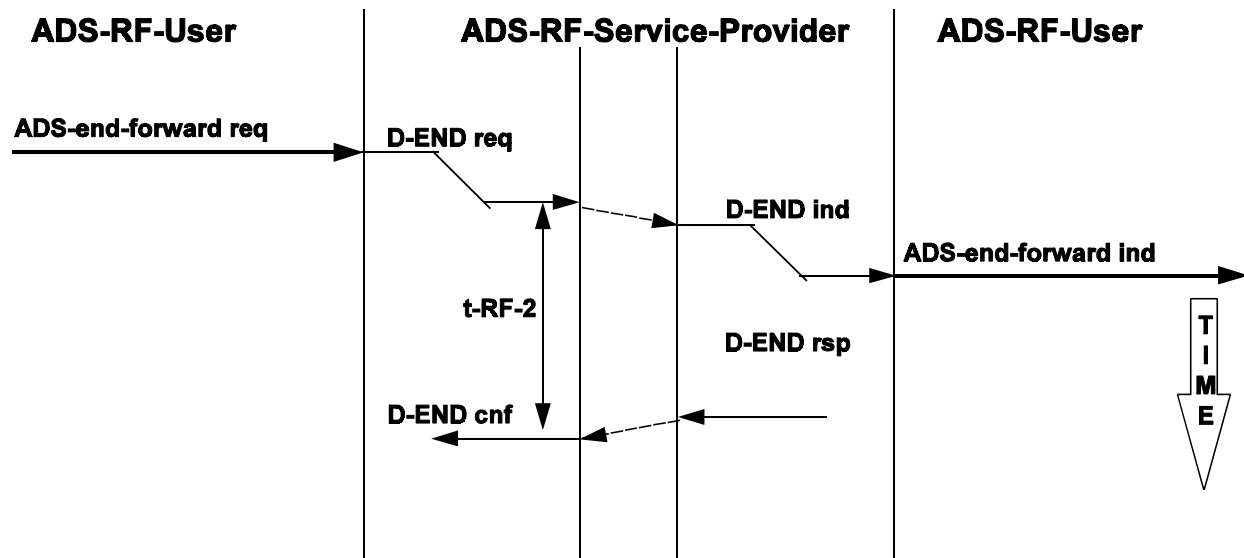


Figure 2.2.2.5-3. Use of end forward service

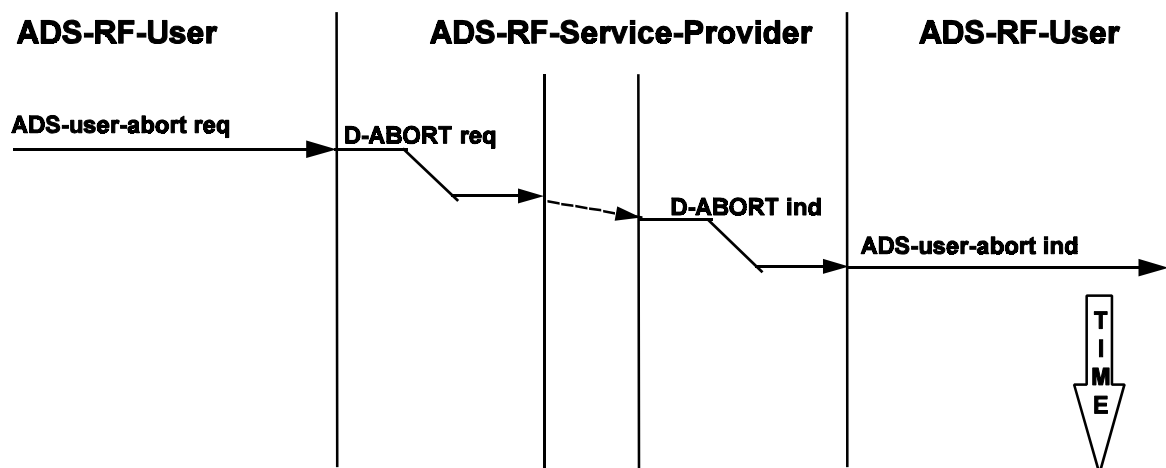


Figure 2.2.2.5-4. ADS-RF-user abort service with a Forward contract in place

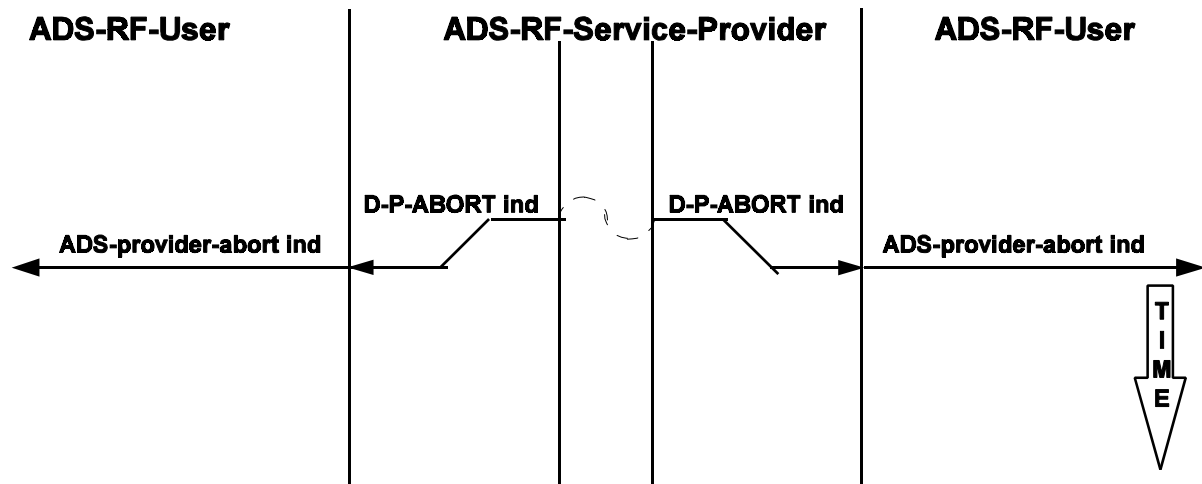


Figure 2.2.2.5-5. Dialogue service provider abort service, with forward contract in place

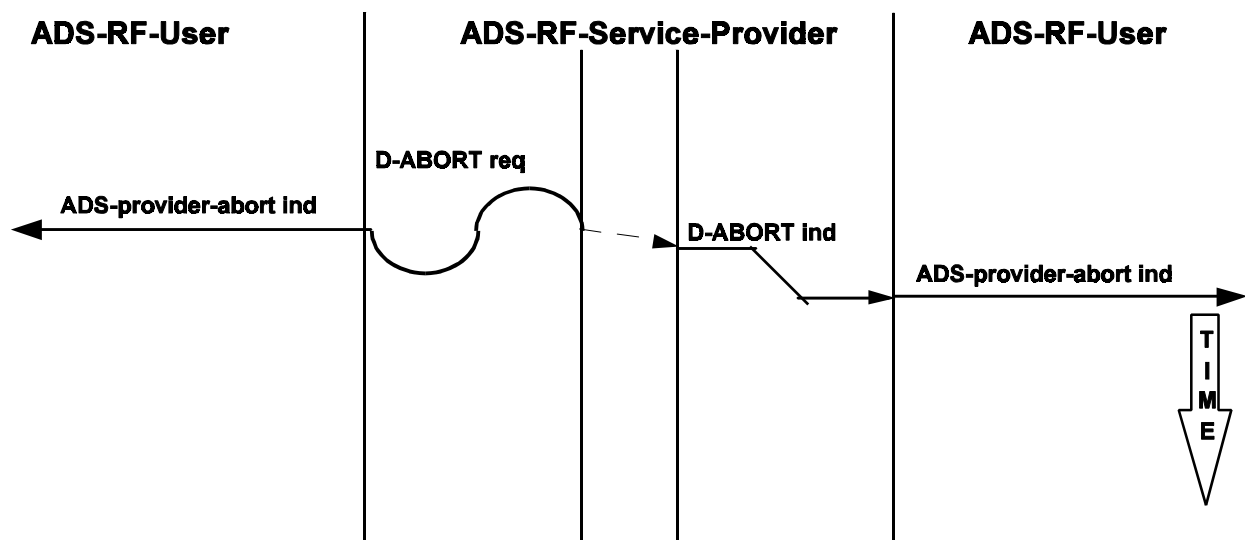


Figure 2.2.2.5-6. ADS-RF-ASE abort with forward contract in place

2.2.2.5.2 ADS RF Service Provider Timers

2.2.2.5.2.1 The ADS-RF-ASE shall be capable of detecting when a timer expires.

Note 1.— Table 2.2.2.5-1 lists the time constraints related to the ADS Report Forwarding function. Each time constraint requires a timer to be set in the ADS protocol machine.

Note 2.— If the timer expires before the final event has occurred, the ADS-RF-ASE takes the appropriate action.

2.2.2.5.2.2 **Recommendation.** — The timer values should be as indicated in Table 2.2.2.5-1.

Table 2.2.2.5-1: ADS RF Service Provider Timers

ADS Service	Timer	Timer Value	Timer Start Event	Timer Stop Event
ADS forward contract	t-RF-1	6 minutes	ADS-start-forward request	ADS-start-forward confirmation
	t-RF-2	6 minutes	D-END request	D-END confirmation

2.2.2.5.3 ADS-ASE Protocol Description

Note.— 2.2.2.5.3 defines the protocol for the ADS-RF-ASE. The protocol for the initiating ADS-RF-ASE and the responding ADS-RF-ASE are given separately.

2.2.2.5.3.1 If an APDU is not received when one is required, or one is received in an inappropriate dialogue service primitive, then the exception handling procedures as described in 2.2.2.5.4.3 shall apply.

2.2.2.5.3.2 Upon receipt of an APDU or dialogue service primitive, if no actions are described for their arrival when in a particular state, then the exception handling procedures as described in 2.2.2.5.4.4 shall apply.

2.2.2.5.3.3 Upon receipt of an APDU that cannot be decoded, then the exception handling procedures as described in 2.2.2.5.4.6 shall apply.

2.2.2.5.3.4 ADS Initiating RF ASE

Note.— The initiating ADS-RF-ASE has the following states:

- a) *RF-I-IDLE*
- b) *RF-I-START*
- c) *RF-I-ACTIVE*
- d) *RF-I-END*

2.2.2.5.3.4.1 On initiation, the initiating ADS-RF-ASE shall be in the RF-I-IDLE state.

2.2.2.5.3.4.2 Upon receipt of an ADS-start-forward request:

2.2.2.5.3.4.2.1 If in the RF-I-IDLE state, the ADS-RF-ASE shall:

- a) invoke D-START request with parameters as defined in Table 2.2.2.5-2;
- b) start the t-RF-1 timer, and
- c) enter the RF-I-START state.

Table 2.2.2.5-2. D-START request parameter values

Parameter name	Derivation of Parameter Value
<i>Called peer Id</i>	<i>ICAO Facility designation</i> parameter value from ADS-start-forward request
<i>Calling peer Id</i>	Not used
<i>DS-user version number</i>	ADS-RF-ASE version number
<i>Security requirements</i>	Not used
<i>Quality of service</i>	Routing class: ATSC, with value from <i>Class of communication service</i> parameter value from ADS-start-forward request Priority: High priority flight safety messages RER: Low
<i>User data</i>	The <i>Forwarded report details</i> parameter value, if provided.

2.2.2.5.3.4.3 Upon receipt of an ADS-forward-report request:

2.2.2.5.3.4.3.1 If in the RF-I-ACTIVE state, the initiating ADS-RF-ASE shall:

- a) invoke D-DATA request with parameters as defined in Table 2.2.2.5-3, and
- b) remain in the RF-I-ACTIVE state.

Table 2.2.2.5-3

Parameter Name	Derivation of Parameter Value
<i>User data</i>	The <i>Forwarded report details</i> parameter value

2.2.2.5.3.4.4 Upon receipt of an ADS-end-forward request:

2.2.2.5.3.4.4.1 If in the RF-I-ACTIVE state, the initiating ADS-RF-ASE shall:

- a) invoke D-END request with parameters as defined in Table 2.2.2.5-4,
- b) start the t-RF-2 timer, and
- c) enter the RF-I-END state.

Table 2.2.2.5-4

Parameter Name	Derivation of Parameter Value
<i>User data</i>	Not provided

2.2.2.5.3.4.5 Upon receipt of an ADS-user-abort request:

2.2.2.5.3.4.5.1 If not in the RF-I-IDLE state, the initiating ADS-RF-ASE shall:

- a) stop any timers,
- b) invoke D-ABORT request with parameters as defined in Table 2.2.2.5-5,
- c) enter the RF-I-IDLE state.

Table 2.2.2.5-5

Parameter Name	Derivation of Parameter Value
<i>Originator</i>	“user”
<i>User data</i>	Not provided

2.2.2.5.3.4.6 Upon receipt of a D-START confirmation with a *Result* parameter value of “accepted”:

2.2.2.5.3.4.6.1 If in the RF-I-START state, the initiating ADS-RF-ASE shall:

- a) stop the t-RF-1 timer,
- b) invoke ADS-start-forward confirmation with parameters as defined in Table 2.2.2.5-6, and
- c) enter the RF-I-ACTIVE state.

Table 2.2.2.5-6

Parameter Name	Derivation of Parameter Value
<i>Reply</i>	“accepted”

2.2.2.5.3.4.7 Upon receipt of a D-START confirmation with a *Result* parameter value of “rejected (permanent)”:

2.2.2.5.3.4.7.1 If in the RF-I-START state, the initiating ADS-RF-ASE shall:

- a) stop the t-RF-1 timer,
- b) invoke ADS-start-forward confirmation with parameters as defined in Table 2.2.2.5-7, and
- c) enter the RF-I-IDLE state.

Table 2.2.2.5-7

Parameter Name	Derivation of Parameter Value
<i>Reply</i>	“Incompatible Version”

2.2.2.5.3.4.8 Upon receipt of a D-END confirmation with a *Result* parameter value of “accepted”:

2.2.2.5.3.4.8.1 If in the RF-I-END state, the initiating ADS-RF-ASE shall:

- a) stop the t-RF-2 timer, and
- b) enter the RF-I-IDLE state.

2.2.2.5.3.4.9 Upon receipt of a D-END confirmation with a *Result* parameter value of “rejected”:

2.2.2.5.3.4.9.1 If in the RF-I-END state, the initiating ADS-RF-ASE shall:

- a) stop the t-RF-2 timer,
- b) invoke D-ABORT request with parameters as defined in Table 2.2.2.5-8, and
- c) enter the RF-I-IDLE state.

Table 2.2.2.5-8

Parameter Name	Derivation of Parameter Value
<i>Originator</i>	“provider”
<i>User data</i>	aDS-provider-abort-PDU with value dialogue-end-not-accepted

2.2.2.5.3.4.10 Upon receipt of a D-ABORT indication with the *Originator* parameter value set to “user”:

2.2.2.5.3.4.10.1 If in the RF-I-START state, the RF-I-ACTIVE state or the RF-I-END, the initiating ADS-RF-ASE shall:

- a) stop any timers,
- b) if not in the RF-I-END state, invoke ADS-user-abort indication, and
- c) enter the RF-I-IDLE state.

2.2.2.5.3.4.11 Upon receipt of a D-ABORT indication with the *Originator* parameter value set to “provider”:

2.2.2.5.3.4.11.1 If in the RF-I-START state, the RF-I-ACTIVE state or the RF-I-END state, the initiating ADS-RF-ASE shall:

- a) stop any timers,
- b) if not in the RF-I-END state, invoke ADS-provider-abort indication with parameter values as defined in Table 2.2.2.5-9, and
- c) enter the RF-I-IDLE state.

Table 2.2.2.5-9

Parameter Name	Derivation of Parameter Value
<i>Reason</i>	D-ABORT <i>user data</i> parameter

2.2.2.5.3.4.12 Upon receipt of a D-P-ABORT indication:

2.2.2.5.3.4.12.1 If in the RF-I-START state or the RF-I-ACTIVE state, the initiating ADS-RF-ASE shall:

- a) stop any timers,
- b) invoke ADS-provider-abort indication with parameter values as defined in Table 2.2.2.5-10, and
- c) enter the RF-I-IDLE state.

Table 2.2.2.5-10

Parameter Name	Derivation of Parameter Value
<i>Reason</i>	communications-service-failure

2.2.2.5.3.4.12.2 If in the RF-I-END state, the initiating ADS-RF-ASE shall:

- a) stop any timers, and
- b) enter the RF-I-IDLE state.

2.2.2.5.3.5 Responding ADS-RF-ASE

Note.— The responding ADS-RF-ASE has the following states:

- a) *RF-R-IDLE*
- b) *RF-R-ACTIVE*

2.2.2.5.3.5.1 On initiation, the responding ADS-RF-ASE shall be in the RF-R-IDLE state.

2.2.2.5.3.5.2 Upon receipt of an ADS-user-abort request:

2.2.2.5.3.5.2.1 If in the RF-I-ACTIVE state, the responding ADS-RF-ASE shall:

- a) invoke D-ABORT request with parameters as defined in Table 2.2.2.5-11,
- b) enter the RF-I-IDLE state.

Table 2.2.2.5-11

Parameter Name	Derivation of Parameter Value
<i>Originator</i>	“user”
<i>User data</i>	Not provided

2.2.2.5.3.5.3 Upon receipt of a D-START indication:

2.2.2.5.3.5.3.1 If in the RF-R-IDLE state, and if the D-START *DS-User Version Number* parameter is not compatible with the version number of the responding ADS-RF-ASE, and the *application service priority* parameter value is “high priority flight safety messages”, and the *RER quality of service* parameter is the abstract value “low”, and the *Routing Class quality of service* parameter identifies the traffic category “Air Traffic Service Communications (ATSC)”, and the *Calling Peer ID* parameter is a valid four to eight character facility designation, the responding ADS-RF-ASE shall:

- a) invoke D-START response with parameter values as defined in Table 2.2.2.5-12, and

- b) remain in the RF-R-IDLE state.

Note.— By “compatible” is meant that the receiving ASE and user is able to react to the receiving protocol in the correct way. If the version numbers are equal, they will always be compatible. If the receiving ASE and user has a version number greater than the initiating version number, then they will be compatible only if the receiving system is able to downgrade itself to the lower version number. If the receiving ASE and user has a version number less than the initiating version number, then the receiving system has no way of knowing whether or not the systems are compatible. It would, therefore, have to assume that it is not compatible..

Table 2.2.2.5-12

Parameter Name	Derivation of Parameter Value
<i>DS-user version number</i>	The version number of the ADS-RF-ASE
<i>Security requirements</i>	Not provided
<i>Quality of service</i>	Not provided
<i>Result</i>	rejected (permanent)
<i>User Data</i>	Not provided

2.2.2.5.3.5.3.2 If in the RF-R-IDLE state, and if the D-START *DS-User Version Number* parameter is compatible with the version number of the responding ADS-RF-ASE, and the *application service priority* parameter value is “high priority flight safety messages”, and the *RER quality of service* parameter is the abstract value “low”, and the *Routing Class quality of service* parameter identifies the traffic category “Air Traffic Service Communications (ATSC)”, and the *Calling Peer ID* parameter is a valid four to eight character facility designation the responding ADS-RF-ASE shall:

- a) invoke ADS-start-forward indication with parameter values as defined in Table 2.2.2.5-14,
- b) invoke D-START response with parameter values as defined in Table 2.2.2.5-13, and
- c) enter the RF-R-ACTIVE state.

Table 2.2.2.5-13

Parameter Name	Derivation of Parameter Value
<i>DS-user version number</i>	Not provided
<i>Security requirements</i>	Not provided
<i>Quality of service</i>	Not provided
<i>Result</i>	accepted
<i>User Data</i>	Not provided

Table 2.2.2.5-14

Parameter Name	Derivation of Parameter Value
<i>Forwarded report details</i>	D-START <i>user data</i> parameter, if provided

2.2.2.5.3.5.4 Upon receipt of a D-DATA indication containing an ADS-forwarded-report-PDU in the user data parameter:

2.2.2.5.3.5.4.1 If in the RF-R-ACTIVE state, the responding ADS-RF-ASE shall:

- a) invoke ADS-forward-report indication with parameters as defined in Table 2.2.2.5-15, and
- b) remain in the RF-R-ACTIVE state.

Table 2.2.2.5-15

Parameter Name	Derivation of Parameter Value
<i>User data</i>	D-DATA <i>user data</i> parameter

2.2.2.5.3.5.5 Upon receipt of a D-END indication:

2.2.2.5.3.5.5.1 If in the RF-R-ACTIVE state, the responding ADS-RF-ASE shall:

- a) invoke ADS-end-forward indication,
- b) invoke D-END response with parameters as defined in Table 2.2.2.5-16, and
- c) enter the RF-R-IDLE state.

Table 2.2.2.5-16

Parameter Name	Derivation of Parameter Value
<i>Result</i>	“accepted”
<i>User data</i>	Not provided

2.2.2.5.3.5.6 Upon receipt of a D-ABORT indication with the *Originator* parameter value set to “user”:

2.2.2.5.3.5.6.1 If in the RF-R-ACTIVE state, the responding ADS-RF-ASE shall:

- a) invoke ADS-user-abort indication, and
- b) enter the RF-R-IDLE state.

2.2.2.5.3.5.7 Upon receipt of a D-ABORT indication with the *Originator* parameter value set to “provider”:

2.2.2.5.3.5.7.1 If in the RF-R-ACTIVE state, the responding ADS-RF-ASE shall:

- a) invoke ADS-provider-abort indication with parameter values as defined in Table 2.2.2.5-17, and
- b) enter the RF-R-IDLE state.

Table 2.2.2.5-17

Parameter Name	Derivation of Parameter Value
<i>Reason</i>	D-ABORT <i>user data</i> parameter

2.2.2.5.3.5.8 Upon receipt of a D-P-ABORT indication:

2.2.2.5.3.5.8.1 If in the RF-R-ACTIVE state, the responding ADS-RF-ASE shall:

- a) invoke ADS-provider-abort indication with parameter values as defined in Table 2.2.2.5-18, and
- b) enter the RF-R-IDLE state.

Table 2.2.2.5-18

Parameter Name	Derivation of Parameter Value
<i>Reason</i>	“communications-service-failure”

2.2.2.5.4 Exception Handling

2.2.2.5.4.1 Timer Expires

2.2.2.5.4.1.1 When either t-RF-1 or t-RF-2 timer expires, the ADS-RF-ASE shall :

- a) invoke D-ABORT request with *Originator* parameter value *provider* and *user data* parameter value aDS-provider-abort-PDU with value *timer-expiry*,
- b) if not in the ADS-I-IDLE state, invoke ADS-provider-abort indication with reason *timer-expiry*, and
- c) enter the RF-I-IDLE state.

2.2.2.5.4.2 Unrecoverable System Error

2.2.2.5.4.2.1 **Recommendation.**— *When the ADS-RF-ASE has an unrecoverable system error, it should:*

- a) *invoke D-ABORT request with Originator parameter value provider and user data parameter value aDS-provider-abort-PDU with value unrecoverable-system-error,*
- b) *if not in the ADS-R-IDLE state or ADS-I-IDLE state, invoke ADS-provider-abort indication with reason unrecoverable-system-error, and*
- c) *if the initiator, enter the RF-I-IDLE state; if the responder, enter the RF-R-IDLE state.*

2.2.2.5.4.3 Invalid PDU

2.2.2.5.4.3.1 When the user data parameter value of a D-START indication is a valid APDU and is not an aDS-forwarded-report-PDU, or the user data parameter value of a D-START confirmation is present, or the user data parameter value of a D-DATA indication is not an aDS-forwarded-report-PDU, or the user data parameter of a D-END indication is present, or the user data parameter of a D-END confirmation is present, the ADS-RF-ASE shall:

- a) *invoke D-ABORT request with Originator parameter value provider and user data parameter value aDS-provider-abort-PDU with value invalid-PDU,*
- b) *if not in the ADS-R-IDLE state or ADS-I-IDLE state, invoke ADS-provider-abort indication with reason invalid-PDU, and*
- c) *if the initiator, enter the RF-I-IDLE state; if the responder, enter the RF-R-IDLE state.*

2.2.2.5.4.4 Sequence Error

2.2.2.5.4.4.1 When a PDU is delivered to the ADS-RF-ASE for which instructions are not stated in 2.2.2.5, it shall:

- a) *invoke D-ABORT request with Originator parameter value provider and user data parameter value aDS-provider-abort-PDU with sequence-error,*
- b) *if not in the ADS-R-IDLE state or ADS-I-IDLE state, invoke ADS-provider-abort indication with reason sequence-error, and*
- c) *if the initiator, enter the RF-I-IDLE state; if the responder, enter the RF-R-IDLE state.*

2.2.2.5.4.4.2 When a Dialogue service primitive is delivered to the ADS-RF-ASE for which there are no instruction in 2.2.2.5.3 (i.e. the primitive was not expected or was expected under other conditions or with other parameter values), it shall:

- a) *invoke D-ABORT request with Originator parameter value provider and user data parameter value aDS-provider-abort-PDU with value sequence-error,*

- b) if not in the ADS-R-IDLE state or ADS-I-IDLE state, invoke ADS-provider-abort indication with reason *sequence-error*, and
- c) if the initiator, enter the RF-I-IDLE state; if the responder, enter the RF-R-IDLE state.

2.2.2.5.4.5 D-START Rejection

2.2.2.5.4.5.1 Upon receipt of a D-START confirmation with the *result* parameter value containing the abstract value rejected (transient) or rejected (permanent), and the *reject source* parameter value containing the abstract value DS provider, the ADS-RF-ASE shall:

- a) invoke ADS-provider-abort indication with reason *cannot-establish-contact*, and
- b) enter the RF-I-IDLE state.

2.2.2.5.4.6 Decoding Error

2.2.2.5.4.6.1 When the ADS-RF-ASE fails to decode an APDU, it shall

- a) invoke D-ABORT request with *Originator* parameter value *provider* and *user data* parameter value aDS-provider-abort-PDU with value *decoding-error*, and
- b) if not in the ADS-R-IDLE state, invoke ADS-provider-abort indication with reason *decoding-error*, and
- c) enter the RF-I-IDLE state.

2.2.2.5.4.7 Invalid QOS

2.2.2.5.4.7.1 Upon receipt of a D-START indication with the *application service priority* parameter set to a value other than the abstract value “high priority flight safety messages”, or the *RER quality of service* parameter set to a value other than the abstract value “low”, or the *Routing Class quality of service* parameter set to a value not identifying the traffic category “Air Traffic Service Communications (ATSC)”, the ADS-RF-ASE shall:

- a) invoke D-ABORT request with *Originator* parameter value *provider* and *user data* parameter value aDS-provider-abort-PDU with value *invalid-qos-parameter*; and
- b) enter the RF-R-IDLE state.

2.2.2.5.5 ADS-ASE State Tables

2.2.2.5.5.1 Priority

2.2.2.5.5.1.1 If the state tables for the ADS-RF-ASE shown below conflict with textual statements made elsewhere in this document, the textual statements shall take precedence.

Note 1.— In the following state tables, the statement “cannot occur” means that if the implementation conforms to the SARPs, it is impossible for this event to occur. If the event does occur, this implies that there is an error in the implementation. If such a situation is detected, it is suggested that the ASE aborts with the error “internal system error”.

Note 2.— In the following state tables, the statement “not permitted” means that the implementation must prevent this event from occurring through some local means. If the event does occur this implies that there is an error in the implementation. If such a situation is detected, it is suggested that the ASE performs a local rejection of the request rather than aborting the dialogue.

Table 2.2.2.5-19. Initiating RF ASE state table

State →	RF-I-IDLE (Initial State)	RF-I-START	RF-I-ACTIVE	RF-I-END
Event ↓				
<i>Primitive Requests and Responses</i>				
ADS-start-forward req	D-START req start t-RF-1 RF-I-START	Not permitted	Not permitted	Not permitted
ADS-forward-report req	Not permitted	Not permitted	D-DATA req RF-I-ACTIVE	Not permitted
ADS-end-forward req	Not permitted	Not permitted	D-END req start t-RF-2 RF-I-END	Not permitted
ADS-user-abort req	Not permitted	D-ABORT req RF-I-IDLE	D-ABORT req RF-I-IDLE	D-ABORT req RF-I-IDLE
<i>Primitive Indications and Confirmations</i>				
D-START cnf accepted	Cannot occur	stop t-RF-1 ADS-start-forward cnf + RF-I-ACTIVE	Cannot occur	Cannot occur
D-START cnf rejected	Cannot occur	stop t-RF-1 ADS-start-forward cnf - RF-I-IDLE	Cannot occur	Cannot occur
D-END cnf accepted	Cannot occur	Cannot occur	Cannot occur	stop t-RF-2 RF-I-IDLE
D-END cnf rejected	Cannot occur	Cannot occur	Cannot occur	stop t-RF-2 D-ABORT req RF-I-IDLE
D-ABORT ind with originator="user"	Cannot occur	stop any timer ADS-user-abort ind RF-I-IDLE	stop any timer ADS-user-abort ind RF-I-IDLE	stop any timer RF-I-IDLE
D-ABORT ind with originator="provider"	Cannot occur	stop any timer ADS-provider-abort ind RF-I-IDLE	stop any timer ADS-provider-abort ind RF-I-IDLE	stop any timer RF-I-IDLE
<i>Timer Expiry</i>				

State →	<i>RF-I-IDLE</i> (Initial State)	<i>RF-I-START</i>	<i>RF-I-ACTIVE</i>	<i>RF-I-END</i>
Event ↓				
t-RF-1	Cannot occur	D-ABORT req ADS-provider-abort ind <i>RF-I-IDLE</i>	Cannot occur	Cannot occur
t-RF-2	Cannot occur	Cannot occur	Cannot occur	D-ABORT req <i>RF-I-IDLE</i>

Table 2.2.2.5-20. Responding RF ASE state table

State →	RF-R-IDLE (Initial State)	RF-R-ACTIVE
Event ↓		
Primitive Requests and Responses		
ADS-user-abort	Not permitted	D-ABORT req RF-R-IDLE
Primitive Indications and Confirmations		
D-START ind with compatible version number	D-START rsp + ADS-start-forward ind RF-R-ACTIVE	Cannot occur
D-START ind with incompatible version number	D-START rsp - RF-R-IDLE	Cannot occur
D-DATA ind	Cannot occur	ADS-forward-report ind RF-R-ACTIVE
D-END ind	Cannot occur	ADS-end-forward ind D-END rsp + RF-R-IDLE
D-ABORT ind with originator=“user”	Cannot occur	ADS-user-abort ind RF-R-IDLE
D-ABORT ind with originator=“provider” or D-P-ABORT ind	Cannot occur	ADS-provider-abort ind RF-R-IDLE

2.2.2.6 Communication Requirements**2.2.2.6.1 Encoding Rules**

2.2.2.6.1.1 The ADS application shall use PER as defined in ISO/IEC 8825-2, using the Basic Unaligned variant to encode/decode the ASN.1 message structure and content specified in 2.2.2.4.

2.2.2.6.2 Dialogue Service Requirements

2.2.2.6.2.1 Primitive Requirements

2.2.2.6.2.1.1 Where dialogue service primitives, that is D-START, D-END, D-ABORT, D-P-ABORT and D-DATA are described as being invoked in 2.2.2.5, the ADS-ground-ASE and the ADS-air-ASE shall exhibit external behavior consistent with the dialogue service, as described in 4.2, having been implemented and its primitives invoked.

2.2.2.6.2.2 Quality of Service Requirements

2.2.2.6.2.2.1 The application service priority for ADS shall have the abstract value of “high priority flight safety messages”.

2.2.2.6.2.2.2 The *RER quality of service* parameter of the D-START request shall be set to the abstract value of “low”.

2.2.2.6.2.2.3 The ADS-ASE shall map the class of communication service abstract values to the ATSC routing class abstract value part of the D-START QOS parameter as presented in Table 2.2.2.6-1.

Table 2.2.2.6-1. Mapping between class of communication and routing class abstract values

Class of Communication Abstract Value	Routing Class Abstract Value
A	Traffic follows Class A ATSC route(s)
B	Traffic follows Class B ATSC route(s)
C	Traffic follows Class C ATSC route(s)
D	Traffic follows Class D ATSC route(s)
E	Traffic follows Class E ATSC route(s)
F	Traffic follows Class F ATSC route(s)
G	Traffic follows Class G ATSC route(s)
H	Traffic follows Class H ATSC route(s)

Note.— ATSC values are defined in 1.3.

2.2.2.7 ADS Report Forwarding User Requirements

2.2.2.7.1 Establishment and operation of a Forward Contract

Note 1.— 2.2.2.7.1 details the actions taken by an ADS-RF-user and a second ADS-RF-user when the first forwards ADS reports to the second.

Note 2.— When an ADS-RF-user requires to initiate forwarding of ADS-reports, it initiates ADS-start-forward request.

Note 3.— When the initiating ADS-RF-user receives an ADS-start-forward confirmation with the Reply parameter value set to “accepted”, it invokes ADS-forward-report request when it requires to forward ADS-reports.

Note 4.— When the initiating ADS-RF-user requires to stop forwarding ADS-reports, it invokes ADS-end-forward request.

2.2.2.7.2 Operation of Aborts

Note 1.— 2.2.2.7.2 details the actions taken by an ADS-RF-user when aborts occur.

Note 2.— When an ADS-RF-user requires to abort the current contract, it invokes ADS-user-abort request.

2.2.2.7.3 Parameter Value Unit, Range and Resolution

2.2.2.7.3.1 An ADS Report Forwarding user shall interpret ADS Report Forwarding parameter value unit, range and resolution as defined in 2.2.2.4.

2.2.2.8 Subsetting Rules

2.2.2.8.1 General

Note.— 2.2.2.8 specifies conformance requirements which all implementations of the ADS Report Forwarding protocol obey.

2.2.2.8.1.1 An implementation of the ADS Report Forwarding service claiming conformance to 2.2.2 shall support the ADS Report Forwarding protocol features as shown in the tables below.

Note.— The ‘status’ column indicates the level of support required for conformance to the ARF-ASE protocol described in this 2.2.2. The values are as follows:

- a) ‘M’ mandatory support is required,*
- b) ‘O’ optional support is permitted for conformance to the ADS Report Forwarding protocol,*
- c) ‘N/A’ the item is not applicable, and*
- d) ‘C.n’ the item is conditional where n is the number which identifies the condition which is applicable.*

Table 2.2.2.8-1. ADS Report Forwarding Protocol Versions Implemented

	Status	Associated Predicate
Version 1	M	none

Table 2.2.2.8-2. ARF Protocol Options

	Status	Associated Predicate
ARF initiator	O	INIT

Table 2.2.2.8-3. ADS-ground-ARF Conformant Configurations

	List of Predicates	Functionality Description
I	INIT	ASE operating as ARF initiator or ARF receiver
II	none	ASE operating as ARF receiver only

2.3 CONTROLLER PILOT DATA LINK COMMUNICATION APPLICATION

2.3.1 INTRODUCTION

2.3.1.1 Overview

2.3.1.1.1 The CPDLC application allows data link communication between controllers and pilots.

2.3.1.1.2 The CPDLC application provides the capability to establish, manage, and terminate CPDLC dialogues between ATS ground and aircraft system peers. Once a dialogue is established, CPDLC provides for controller/pilot message exchange.

2.3.1.1.3 The CPDLC application also provides the capability to establish, manage, and terminate CPDLC dialogues between two ATC ground system peers for the purpose of ground/ground forwarding of a CPDLC message.

Note 1.— Structure:

- a) 2.3.1: *INTRODUCTION* contains the structure of 2.3 and a summary of the functional capabilities of CPDLC.
- b) 2.3.2: *GENERAL REQUIREMENTS* contains the CPDLC version number, and error processing requirements.
- c) 2.3.3: *ABSTRACT SERVICE DEFINITION* contains the description of the abstract service provided by the CPDLC Application Service Element (CPDLC-ASE).
- d) 2.3.4: *FORMAL DEFINITION OF MESSAGES* contains the formal definition of messages exchanged by CPDLC ASEs using Abstract Syntax Notation Number One (ASN.1).
- e) 2.3.5: *PROTOCOL DEFINITION* describes the exchanges of messages allowed by the CPDLC protocol, as well as time constraints and CPDLC-ASE protocol descriptions and state tables.
- f) 2.3.6: *COMMUNICATION REQUIREMENTS* contains the requirements that the CPDLC application imposes on the underlying communication system.
- g) 2.3.7: *CPDLC USER REQUIREMENTS* contains requirements imposed on the user of the CPDLC ASE service and message description tables.
- h) 2.3.8: *SUBSETTING RULES* defines the conformant subsets for the CPDLC-ASE.

Note 2.— Functional Descriptions

- a) The **Controller-Pilot Message Exchange Function** defines a method for a controller and pilot to exchange messages via data link. This function provides messages for the following :
 - 1) general information exchange;
 - 2) clearance
 - i) delivery,
 - ii) request, and
 - iii) response;
 - 3) altitude/identity surveillance;
 - 4) monitoring of current/planned position;
 - 5) advisories
 - i) request and
 - ii) delivery;
 - 6) system management functions; and
 - 7) emergency situations.
- b) The **Transfer of Data Authority Function** provides the capability for the current data authority to designate another ground system as the next data authority. A CPDLC dialogue can be opened with or by the next data authority at a time before becoming the current data authority. This capability is intended to prevent a loss of communication that would occur if the next data authority were prevented from actually setting up a dialogue with an aircraft until it became the current data authority. The designation of a next data authority is accomplished using a CPDLC message.
- c) The **Down Stream Clearance Function** provides the capability for an aircraft to contact an air traffic service unit which is not the current data authority for the purpose of receiving a down stream clearance. This information is exchanged using CPDLC message(s).
- d) The **Ground Forward Function** provides the capability for a ground system to forward information received in a CPDLC message to another ground system. The ground forwarding function can be used by the controlling data authority to forward an aircraft request to the next data authority, so that an aircraft does not need to issue the same request again. This function can also be used by a

downstream data authority to pass a message to a current data authority for transmission by the current data authority to an aircraft. This information is exchanged using CPDLC message(s). It is a one-way forwarding of information with an indication of success, failure or non-support from the receiving ground system.

Note 3.— See 2.3.7 for detailed CPDLC message intent/use descriptions.

2.3.2 GENERAL REQUIREMENTS

2.3.2.1 CPDLC ASE Version Number

2.3.2.1.1 The CPDLC-air-ASE and CPDLC-ground-ASE version numbers shall both be set to one.

2.3.2.2 Error Processing Requirements

2.3.2.2.1 In the event of information input by the CPDLC-user being incompatible with that able to be processed by the system, the CPDLC-user shall be notified.

2.3.2.2.2 In the event of a CPDLC-user invoking a CPDLC service primitive when the CPDLC-ASE is not in a state specified in 2.3.5, the CPDLC-user shall be notified.

2.3.3 THE ABSTRACT SERVICE

2.3.3.1 Service Description

2.3.3.1.1 An implementation of either the CPDLC ground based service or the CPDLC air based service shall exhibit external behavior consistent with having implemented a CPDLC-ground-ASE, or CPDLC-air-ASE respectively, with the following abstract service interface primitives, making them available to the CPDLC-ground-user or CPDLC-air-user respectively.

Note 1.— There is no requirement to implement the service in a CPDLC product; however, it is necessary to implement the ground based and air based system in such a way that it will be impossible to detect (from the peer system) whether or not the interface has been built.

Note 2.— This chapter defines the abstract service interface for the CPDLC service. The CPDLC-ASE abstract service is described in this chapter from the viewpoint of the CPDLC-air-user, the CPDLC-ground-user and the CPDLC-service-provider.

Note 3.— This chapter defines the static behavior (i.e., the format) of the CPDLC abstract service. Its dynamic behavior (i.e., how it is used) is described in 2.3.7.

Note 4.— Figure 2.3.3-1 shows the functional model of the CPDLC Application. The functional modules identified in this model are the following :

- a) the CPDLC-user,
- b) the CPDLC Application Entity (CPDLC-AE) service,
- c) the CPDLC-AE,
- d) the CPDLC Control Function (CPDLC-CF),
- e) the CPDLC Application Service Element (CPDLC-ASE) service,
- f) the CPDLC-ASE, and
- g) the Dialogue Service (DS).

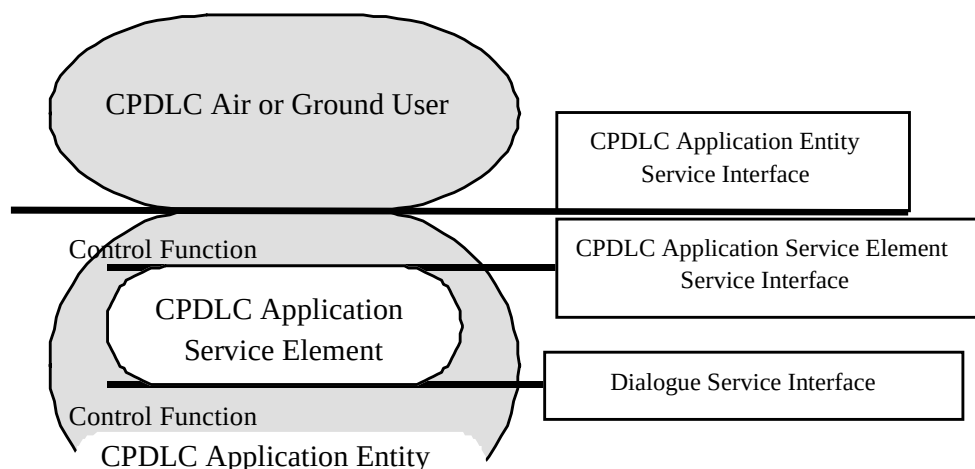


Figure 2.3.3-1. Functional Model of the CPDLC Application

Note 5.— The CPDLC-user represents the operational part of the CPDLC system. This user does not perform the communication functions but relies on a communication service provided to it via the CPDLC-AE through the CPDLC-AE service. The individual actions possible through the CPDLC-AE service are called service primitives.

Note 6.— The CPDLC-AE consists of several elements including the CPDLC-ASE and the CPDLC-CF.

Note 7.— The CPDLC-ASE is the element in the communication system which executes the CPDLC specific protocol. In other words, it takes care of the CPDLC specific service primitive sequencing, message creation, timer management, error and exception handling.

Note 8.— This CPDLC-CF is responsible for mapping service primitives received from one element (such as the CPDLC-ASE and the CPDLC-user) to service primitives of other abstract elements.

Note 9.— The CPDLC-ASE has two abstract boundaries with the CPDLC-CF: the CPDLC-ASE service and the dialogue service. The CPDLC-CF maps CPDLC-AE service primitives to other abstract elements in the CPDLC-AE and the underlying communication service, and vice versa.

2.3.3.2 The CPDLC-ASE Abstract Service

2.3.3.2.1 The CPDLC-ASE abstract service shall consist of a subset of the following services as allowed in 2.3.8:

- a) CPDLC-start service as defined in 2.3.3.3,
- b) DSC-start service as defined in 2.3.3.4,
- c) CPDLC-message service as defined in 2.3.3.5,
- d) CPDLC-end service as defined in 2.3.3.6,

- e) DSC-end service as defined in 2.3.3.7,
- f) CPDLC-forward service as defined in 2.3.3.8,
- g) CPDLC-user-abort service as defined in 2.3.3.9, and
- h) CPDLC-provider-abort service as defined in 2.3.3.10.

Note 1.— For a given primitive, the presence of each parameter is described by one of the following values in the parameter tables in 2.3.3.

- a) **blank** not present;
- b) **C** conditional upon some predicate explained in the text;
- c) **C(=)** conditional upon the value of the parameter to the left being present, and equal to that value;
- d) **M** mandatory;
- e) **M(=)** mandatory, and equal to the value of the parameter to the left;
- f) **U** user option.

Note 2.— The following abbreviations are used in this document:

- a) **Req** - request; data is input by CPDLC-user initiating the service to its respective ASE,
- b) **Ind** - indication; data is indicated by the receiving ASE to its respective CPDLC-user,
- c) **Rsp** - response; data is input by receiving CPDLC user to its respective ASE, and
- d) **Cnf** - confirmation; data is confirmed by the initiating ASE to its respective CPDLC-user.

Note 3.— An unconfirmed service allows a message to be transmitted in one direction without providing a corresponding response.

Note 4.— A confirmed service provides end-to-end confirmation that a message sent by one user was received by its peer user.

Note 5.— An abstract syntax is a syntactical description of a parameter which does not imply a specific implementation. Only when the CPDLC-ASE maps a parameter into an APDU field, or vice versa, the abstract syntax of the parameter is described by using ASN.1 of 2.3.4 for this field.

2.3.3.3 CPDLC-start Service

Note 1.— The CPDLC-start service is used by the CPDLC-air-user or CPDLC-ground-user to establish a CPDLC dialogue. It is a confirmed service.

Note 2.— Once a CPDLC dialogue is established it remains open until explicitly closed. (See CPDLC-end and CPDLC-abort services.)

2.3.3.3.1 The CPDLC-start service shall contain the primitives and parameters as presented in Table 2.3.3-1.

Table 2.3.3-1. CPDLC-start Service Parameters

Parameter Name	Req	Ind	Rsp	Cnf
Called Peer Identifier	M			
Calling Peer Identifier	M	M(=)		
CPDLC Message	U	C(=)		
Reject Reason			C	C(=)
Result			M	M(=)
Class of Communication Service	U	M		

2.3.3.3.2 Called Peer Identifier

Note 1.— If the service is ground initiated, this parameter contains the addressed aircraft's 24-bit aircraft address.

Note 2.— If the service is air initiated, this parameter contains the addressed ground system's facility designation.

2.3.3.3.2.1 If the service is ground initiated, the *Called Peer Identifier* parameter value shall conform to the abstract syntax 24-bit aircraft address.

2.3.3.3.2.2 If the service is air initiated, the *Called Peer Identifier* parameter value shall conform to the abstract syntax four to eight-character facility designation.

2.3.3.3.3 Calling Peer Identifier

Note 1.— If the service is ground initiated, this parameter contains the sending ground system's facility designation.

Note 2.— If the service is air initiated, this parameter contains the sending aircraft's 24-bit aircraft address.

2.3.3.3.3.1 If the service is ground initiated, the *Calling Peer Identifier* parameter value shall conform to the abstract syntax four to eight-character facility designation.

2.3.3.3.3.2 If the service is air initiated, the *Calling Peer Identifier* parameter value shall conform to the abstract syntax 24-bit aircraft address.

2.3.3.3.4 CPDLC Message

Note.— The CPDLC-user can use this parameter to send a CPDLC message to its peer user.

2.3.3.3.4.1 The *CPDLC Message* parameter value shall conform to the ASN.1 abstract syntax *ATCUplinkMessage*, if supplied by the CPDLC-ground-user.

2.3.3.3.4.2 The *CPDLC Message* parameter value shall conform to the ASN.1 abstract syntax *ATCDownlinkMessage*, if supplied by the CPDLC-air-user.

2.3.3.3.5 Reject Reason

Note.— This parameter is used to provide a reason for rejecting a CPDLC dialogue.

2.3.3.3.5.1 If the CPDLC-user accepts the request to open a CPDLC dialogue, the CPDLC user shall be prohibited from providing a CPDLC message for the *Reject Reason* parameter.

2.3.3.3.5.2 The *Reject Reason* parameter value shall conform to the ASN.1 abstract syntax *ATCUplinkMessage* if supplied by the CPDLC-ground-user.

2.3.3.3.5.3 The *Reject Reason* parameter value shall conform to the ASN.1 abstract syntax *ATCDownlinkMessage* if supplied by the CPDLC-air-user.

2.3.3.3.6 Result

Note.— This parameter is used to indicate whether or not a requested CPDLC dialogue is accepted.

2.3.3.3.6.1 This parameter shall have one of two abstract values: “accepted” or “rejected”.

2.3.3.3.7 Class of Communication Service

Note 1.— This parameter contains the value of the required class of communication service. If not specified by the CPDLC-user, this indicates that there is no routing preference.

Note 2.— This parameter is used by the CPDLC-ground-user to determine if the Class of Communication value is acceptable for the establishment of a CPDLC dialogue.

Note 3.— The parameter indicated to the peer user is that provided by the user if specified by the user, else it indicates that no routing preference was requested by the CPDLC dialogue initiator.

2.3.3.3.7.1 Where specified by the CPDLC-user, the *Class of Communication Service* parameter shall have one of the following abstract values: “A”, “B”, “C”, “D”, “E”, “F”, “G”, or “H”.

2.3.3.3.7.2 When this parameter is provided by the user, the same value shall be indicated to the peer user, else the abstract value “ATSC - No Traffic Type Policy Preference” is indicated.

2.3.3.4 DSC-start Service

Note 1.— The DSC-start service is used to establish a DSC dialogue for the purpose of providing down stream clearances. It is a confirmed service.

Note 2.— Once a DSC dialogue is established it remains open until explicitly closed. (See DSC-end and CPDLC-abort services.)

2.3.3.4.1 The DSC-start service shall contain the primitives and parameters as presented in Table 2.3.3-2.

Table 2.3.3-2. DSC-start Service Parameters

Parameter Name	Req	Ind	Rsp	Cnf
Facility Designation	M			
Aircraft Address	M	M(=)		
CPDLC Message	U	C(=)		
Reject Reason			C	C(=)
Result			M	M(=)
Class of Communication Service	U	M		

2.3.3.4.2 Facility Designation

Note.— This parameter contains the addressed ground system 's facility designation.

2.3.3.4.2.1 The *Facility Designation* parameter value shall conform to the abstract syntax four to eight-character facility designation.

2.3.3.4.3 Aircraft Address

2.3.3.4.3.1 The *Aircraft Address* parameter value shall conform to the abstract syntax 24-bit aircraft address.

Note.— This parameter contains the aircraft 's 24 bit aircraft address.

2.3.3.4.4 CPDLC Message

Note.— The CPDLC-air-user can use this parameter to send a CPDLC message to a CPDLC-ground-user.

2.3.3.4.4.1 The *CPDLC Message* parameter value shall conform to the ASN.1 abstract syntax ATCDownlinkMessage.

2.3.3.4.5 Reject Reason

Note.— The parameter is used to provide a reason for rejecting a DSC dialogue.

2.3.3.4.5.1 If, the CPDLC-ground-user accepts the request to open a DSC dialogue, the CPDLC-ground-user shall be prohibited from providing a CPDLC message for the *Reject Reason* parameter.

2.3.3.4.5.2 The *Reject Reason* parameter shall conform to the ASN.1 abstract syntax ATCUplinkMessage.

2.3.3.4.6 Result

Note.— This parameter is used to indicate whether or not a requested DSC dialogue is accepted.

2.3.3.4.6.1 The *Result* parameter value shall have one of two abstract values: “accepted” or “rejected”.

2.3.3.4.7 Class of Communication Service

Note 1.— This parameter contains the value of the required class of communication service. If not specified by the CPDLC-air-user, this indicates that there is no routing preference.

Note 2.— This parameter is used by the CPDLC-ground-user to determine if the Class of Communication value is acceptable for the establishment of a DSC dialogue.

Note 3.— If provided by the user, the parameter indicated to the peer user is that provided by the user, else it indicates that no routing preference was requested by the CPDLC dialogue initiator .

2.3.3.4.7.1 Where specified by the CPDLC-air-user, the *Class of Communication Service* parameter shall have one of the following abstract values: “A”, “B”, “C”, “D”, “E”, “F”, “G”, or “H”.

2.3.3.4.7.2 When this parameter is provided by the user, the same value shall be indicated to the peer user, else the abstract value “ATSC - No Traffic Type Policy Preference” is indicated.

2.3.3.5 CPDLC-message Service

Note.— The CPDLC-message service can be used for pilot/controller message exchange, once a dialogue is established. It is an unconfirmed service.

2.3.3.5.1 The CPDLC-message service shall contain the primitives and parameters as presented in Table 2.3.3-3.

Table 2.3.3-3. CPDLC-message Service Parameters

Parameter Name	Req	Ind
CPDLC Message	M	M(=)

2.3.3.5.2 CPDLC Message

Note.— This parameter contains a CPDLC message.

2.3.3.5.2.1 If the CPDLC-message service is invoked by the CPDLC-ground-user, the *CPDLC Message* parameter value shall conform to the ASN.1 abstract syntax ATCUplinkMessage.

2.3.3.5.2.2 If the CPDLC-message service is invoked by the CPDLC-air-user, the *CPDLC Message* parameter value shall conform to the ASN.1 abstract syntax ATCDownlinkMessage.

2.3.3.6 CPDLC-end Service

Note.— The CPDLC-end service is used by the CPDLC-ground-user to end a CPDLC dialogue with a CPDLC-air-user. It is a confirmed service.

2.3.3.6.1 The CPDLC-end service shall contain the primitives and parameters as presented in Table 2.3.3-4.

Table 2.3.3-4. CPDLC-end Service Parameters

Parameter Name	Req	Ind	Rsp	Cnf
CPDLC Message	U	C(=)	U	C(=)
Result			M	M(=)

2.3.3.6.2 CPDLC Message

Note.— This parameter contains a CPDLC message.

2.3.3.6.2.1 The *CPDLC Message* parameter value shall conform to the ASN.1 abstract syntax ATCUplinkMessage, if provided by the CPDLC-ground-user.

2.3.3.6.2.2 The *CPDLC Message* parameter value shall conform to the ASN.1 abstract syntax ATCDownlinkMessage, if provided by the CPDLC-air-user.

2.3.3.6.3 Result

Note.— This parameter is used to indicate whether or not a request to terminate a CPDLC dialogue is accepted.

2.3.3.6.3.1 The *Result* parameter shall have one of two abstract values: “accepted” or “rejected”.

2.3.3.7 DSC-end Service

Note.— The DSC-end service is used by the DSC-air-user to end a DSC dialogue with a CPDLC-ground-user. It is a confirmed service.

2.3.3.7.1 The DSC-end service shall contain the primitives and parameters as presented in Table 2.3.3-5.

Table 2.3.3-5. DSC-end Service Parameters

Parameter Name	Req	Ind	Rsp	Cnf
CPDLC Message	U	C(=)	U	C(=)
Result			M	M(=)

2.3.3.7.2 CPDLC Message

Note.— This parameter contains a CPDLC message.

2.3.3.7.2.1 The *CPDLC Message* parameter value shall conform to the ASN.1 abstract syntax *ATCUplinkMessage*, if provided by the CPDLC-ground-user.

2.3.3.7.2.2 The *CPDLC Message* parameter value shall conform to the ASN.1 abstract syntax *ATCDownlinkMessage* if provided by the CPDLC-air-user.

2.3.3.7.3 Result

Note.— This parameter is used to indicate whether or not a request to terminate a DSC dialogue is accepted.

2.3.3.7.3.1 The *Result* parameter shall have one of two abstract values: “accepted” or “rejected”.

2.3.3.8 CPDLC-forward Service

Note.— The CPDLC-forward service is used by a CPDLC-ground-user to send a CPDLC message to another CPDLC-ground-user. Its primary use is for the forwarding of aircraft requests.

2.3.3.8.1 If the CPDLC-forward service is supported by the receiving ground system and the sending CPDLC-ground-ASE and receiving CPDLC-ground-ASE version numbers are equal, the CPDLC-forward service shall contain the primitives and parameters as presented in Table 2.3.3-6.

Table 2.3.3-6. CPDLC-forward Service Parameters (Service Supported, Versions Equal)

Parameter Name	Req	Ind	Cnf
Called Facility Designation	M		
Calling Facility Designation	M	M(=)	
CPDLC Message	M	M(=)	
Class of Communication Service	U		
Result			M

2.3.3.8.2 If the CPDLC-forward service is not supported by the receiving ground system, or if the CPDLC-forward service is supported by the receiving ground system but the sending CPDLC-ground-ASE and receiving CPDLC-ground-ASE version numbers are not equal, the CPDLC-forward service shall contain the primitives and parameters as presented in Table 2.3.3-7.

Table 2.3.3-7. CPDLC-forward Service Parameters (Service Not Supported or Versions Not Equal)

Parameter Name	Req	Cnf
Called Facility Designation	M	
Calling Facility Designation	M	
ASE Version Number		C
CPDLC Message	M	
Class of Communication Service	U	
Result		M

2.3.3.8.3 Called Facility Designation

Note.— This parameter contains the addressed ground system 's facility designation.

2.3.3.8.3.1 The *Called Facility Designation* parameter value shall conform to the abstract syntax four to eight-character facility designation.

2.3.3.8.4 Calling Facility Designation

Note.— This parameter contains the sending ground system 's facility designation.

2.3.3.8.4.1 The *Calling Facility Designation* parameter value shall conform to the abstract syntax four to eight-character facility designation.

2.3.3.8.5 ASE Version Number

Note.— This parameter contains the version number of the CPDLC-ASE.

2.3.3.8.5.1 When provided by the CPDLC-ground-ASE, the *ASE Version Number* parameter shall conform to the abstract integer value in the range 1-255.

2.3.3.8.5.2 Only if the sending CPDLC-ground-ASE version number is not equal to the receiving CPDLC-ground-ASE version number shall the receiving CPDLC-ground-ASE version number be confirmed to the sending CPDLC-ground-user.

Note.— If the sending CPDLC-ground-ASE version number is the same as the receiving CPDLC-ground-ASE version number, the Version Number parameter is not present in the indication given to the receiving CPDLC-ground-user, nor in the confirmation to the sending CPDLC-ground-user.

2.3.3.8.6 CPDLC Message

Note.— The sending CPDLC-ground-user uses this parameter to forward a CPDLC message to another CPDLC-ground-user.

2.3.3.8.6.1 The *CPDLC Message* parameter value shall conform to the ASN.1 abstract syntax *ATCForwardMessage*, when supplied by the CPDLC-ground-user.

2.3.3.8.7 Class of Communication Service

Note.— This parameter contains the value of the required class of communication service. If not specified by the CPDLC-ground-user, this indicates that there is no routing preference.

2.3.3.8.7.1 Where specified by the CPDLC-ground-user, the *Class of Communication Service* parameter shall have one of the following abstract values: “A”, “B”, “C”, “D”, “E”, “F”, “G”, or “H”.

2.3.3.8.8 Result

Note.— This parameter contains the result of the CPDLC-forward service. It will indicate success (service supported and matching versions), service unsupported, or version number incompatibility.

2.3.3.8.8.1 The *Result* parameter value shall conform to the ASN.1 abstract syntax *ATCForwardResponse*.

2.3.3.9 CPDLC-user-abort Service

Note 1.— This service provides the capability for either the CPDLC-air-user or a CPDLC-ground-user to abort communication with its peer. It can be invoked at any time the user is aware that the CPDLC service is in operation. The CPDLC-user-abort service can be used for operational or technical reasons. It is an unconfirmed service. Messages in transit may be lost during this operation.

Note 2.— If the service is invoked prior to complete establishment of the dialogue, the CPDLC-user-abort indication may not be provided. A CPDLC-provider-abort indication may result instead.

2.3.3.9.1 The CPDLC-user-abort service shall contain the primitives and parameters as presented in Table 2.3.3-8.

Table 2.3.3-8. CPDLC-user-abort Service Parameters

Parameter Name	Req	Ind
Reason	U	M

2.3.3.9.2 Reason

Note 1.— This parameter is used to indicate a reason for aborting the CPDLC or DSC dialogue.

Note 2.— If provided by the user, the parameter indicated to the peer user is that provided by the user, else it is what the ASE supplies.

2.3.3.9.2.1 The *Reason* parameter value shall conform to the ASN.1 abstract syntax CPDLCUserAbortReason.

2.3.3.9.2.2 When this parameter is provided by the user, the same value shall be indicated to the peer user.

2.3.3.10 CPDLC-provider-abort Service

Note.— This service provides the capability for the CPDLC-service provider to inform its active users that it can no longer provide the CPDLC service. Messages in transit may be lost during this operation.

2.3.3.10.1 The CPDLC-provider-abort service shall contain the primitives and parameters as presented in Table 2.3.3-9.

Table 2.3.3-9. CPDLC-provider-abort Service Parameters

Parameter Name	Ind
Reason	M

2.3.3.10.2 Reason

Note.— This parameter identifies the reason for the abort.

2.3.3.10.2.1 The *Reason* parameter shall conform to the ASN.1 abstract syntax CPDLCProviderAbortReason.

2.3.4 FORMAL DEFINITIONS OF MESSAGES

2.3.4.1 Encoding/Decoding Rules

2.3.4.1.1 A CPDLC-air-ASE shall be capable of encoding AircraftPDUs APDUs and decoding GroundPDUs APDUs.

2.3.4.1.2 A CPDLC-ground-ASE shall be capable of encoding GroundPDUs APDUs and decoding AircraftPDUs APDUs.

2.3.4.2 CPDLC ASN.1 Abstract Syntax

2.3.4.2.1 The abstract syntax of the CPDLC protocol data units shall comply with the description contained in the ASN.1 module CPDLCMessageSetVersion1 conforming to ISO/IEC 8824, as defined in this section.

CPDLCMessageSetVersion1 DEFINITIONS::=

BEGIN

-- -----
-- Ground Generated Messages - Top level
-- -----

GroundPDUs ::= CHOICE

abortUser	[0]	CPDLCUserAbortReason,
abortProvider	[1]	CPDLCProviderAbortReason,
startup	[2]	UplinkMessage,
send	[3]	ATCUplinkMessage,
forward	[4]	ATCForwardMessage,
forwardresponse	[5]	ATCForwardResponse,
...		

UplinkMessage ::= CHOICE

noMessage	[0]	NULL,
aTCUplinkMessage	[1]	ATCUplinkMessage

ATCUplinkMessage ::= SEQUENCE

```
{  
  header          ATCMessageHeader,  
  messageData     ATCUplinkMessageData  
}
```

ATCUplinkMessageData ::= SEQUENCE

```
{  
  elementIds      SEQUENCE SIZE (1..5) OF ATCUplinkMsgElementId,  
  constrainedData SEQUENCE  
    {  
      routeClearanceData SEQUENCE SIZE (1..2) OF RouteClearance OPTIONAL,  
      ...  
    }  
  OPTIONAL  
}
```

ATCForwardMessage ::= SEQUENCE

```
{  
  forwardHeader    ForwardHeader,  
  forwardMessage   ForwardMessage  
}
```

ForwardHeader ::= SEQUENCE

```
{  
  dateTime        DateTimeGroup,  
  aircraftID      AircraftFlightIdentification,  
  aircraftAddress AircraftAddress  
}
```

ForwardMessage ::= CHOICE

```
{  
  upElementIDs    [0] ATCUplinkMessageData,  
  downElementIDs  [1] ATCDownlinkMessageData  
}
```

ATCForwardResponse ::= ENUMERATED

```
{  
  success              (0),  
  service-not-supported (1),  
  version-not-equal    (2),  
  ...  
}
```

-- Aircraft Generated Messages - Top level

AircraftPDUs ::= CHOICE

```

{
  abortUser      [0]    CPDLCUserAbortReason,
  abortProvider  [1]    CPDLCProviderAbortReason,
  startdown      [2]    StartDownMessage,
  send           [3]    ATCDownlinkMessage,
  ...
}

```

StartDownMessage ::= SEQUENCE

```

{
  mode                Mode DEFAULT cpdlc,
  startDownlinkMessage DownlinkMessage
}

```

Mode ::= ENUMERATED

```

{
  cpdlc      (0),
  dsc        (1)
}

```

DownlinkMessage ::= CHOICE

```

{
  noMessage                [0]    NULL,
  aTCDownlinkMessage       [1]    ATCDownlinkMessage
}

```

ATCDownlinkMessage ::= SEQUENCE

```

{
  header                ATCMessageHeader,
  messageData           ATCDownlinkMessageData
}

```

ATCDownlinkMessageData ::= SEQUENCE

```

{
  elementIds            SEQUENCE SIZE (1..5) OF ATCDownlinkMsgElementId,
  constrainedData       SEQUENCE
    {
      routeClearanceData SEQUENCE SIZE (1..2) OF RouteClearance OPTIONAL,
      ...
    }
  OPTIONAL
}

```

 -- Uplink and Downlink messages - Common Elements

ATCMessageHeader ::= SEQUENCE

{			
messageIdNumber	[0]	MsgIdentificationNumber,	
messageRefNumber	[1]	MsgReferenceNumber	OPTIONAL,
dateTime	[2]	DateTimeGroup,	
logicalAck	[3]	LogicalAck	DEFAULT notRequired
}			

MsgIdentificationNumber ::= INTEGER (0..63)

MsgReferenceNumber ::= INTEGER (0..63)

CPDLCUserAbortReason ::= ENUMERATED

{	
undefined	(0),
no-message-identification-numbers-available	(1),
duplicate-message-identification-numbers	(2),
no-longer-next-data-authority	(3),
current-data-authority-abort	(4),
commanded-termination	(5),
invalid-response	(6),
...	
}	

CPDLCProviderAbortReason ::= ENUMERATED

{	
timer-expired	(0),
undefined-error	(1),
invalid-PDU	(2),
protocol-error	(3),
communication-service-error	(4),
communication-service-failure	(5),
invalid-QOS-parameter	(6),
expected-PDU-missing	(7),
...	
}	

LogicalAck ::= ENUMERATED

{	
required	(0),
notRequired	(1)
}	

 -- Uplink message element

ATCUplinkMsgElementId ::= CHOICE

```

    {
--   UNABLE                               Urg(N)/Alr(M)/Resp(N)
      uM0NULL                             [0] NULL,

--   STANDBY                             Urg(N)/Alr(L)/Resp(N)
      uM1NULL                             [1] NULL,

--   REQUEST DEFERRED                    Urg(N)/Alr(L)/Resp(N)
      uM2NULL                             [2] NULL,

--   ROGER                               Urg(N)/Alr(L)/Resp(N)
      uM3NULL                             [3] NULL,

--   AFFIRM                              Urg(N)/Alr(L)/Resp(N)
      uM4NULL                             [4] NULL,

--   NEGATIVE                            Urg(N)/Alr(L)/Resp(N)
      uM5NULL                             [5] NULL,

--   EXPECT [level]                      Urg(L)/Alr(L)/Resp(R)
      uM6Level                            [6] Level,

--   EXPECT CLIMB AT [time]              Urg(L)/Alr(L)/Resp(R)
      uM7Time                             [7] Time,

--   EXPECT CLIMB AT [position]          Urg(L)/Alr(L)/Resp(R)
      uM8Position                         [8] Position,

--   EXPECT DESCENT AT [time]            Urg(L)/Alr(L)/Resp(R)
      uM9Time                             [9] Time,

--   EXPECT DESCENT AT [position]        Urg(L)/Alr(L)/Resp(R)
      uM10Position                        [10] Position,

--   EXPECT CRUISE CLIMB AT [time]       Urg(L)/Alr(L)/Resp(R)
      uM11Time                           [11] Time,

--   EXPECT CRUISE CLIMB AT [position]   Urg(L)/Alr(L)/Resp(R)
      uM12Position                        [12] Position,

--   AT [time] EXPECT CLIMB TO [level]   Urg(L)/Alr(L)/Resp(R)
      uM13TimeLevel                       [13] TimeLevel,

```

--	AT [position] EXPECT CLIMB TO [level] uM14PositionLevel	Urg(L)/Alr(L)/Resp(R) [14] PositionLevel,
--	AT [time] EXPECT DESCENT TO [level] uM15TimeLevel	Urg(L)/Alr(L)/Resp(R) [15] TimeLevel,
--	AT [position] EXPECT DESCENT TO [level] uM16PositionLevel	Urg(L)/Alr(L)/Resp(R) [16] PositionLevel,
--	AT [time] EXPECT CRUISE CLIMB TO [level] uM17TimeLevel	Urg(L)/Alr(L)/Resp(R) [17] TimeLevel,
--	AT [position] EXPECT CRUISE CLIMB TO [level] uM18PositionLevel	Urg(L)/Alr(L)/Resp(R) [18] PositionLevel,
--	MAINTAIN [level] uM19Level	Urg(N)/Alr(M)/Resp(W/U) [19] Level,
--	CLIMB TO [level] uM20Level	Urg(N)/Alr(M)/Resp(W/U) [20] Level,
--	AT [time] CLIMB TO [level] uM21TimeLevel	Urg(N)/Alr(M)/Resp(W/U) [21] TimeLevel,
--	AT [position] CLIMB TO [level] uM22PositionLevel	Urg(N)/Alr(M)/Resp(W/U) [22] PositionLevel,
--	DESCEND TO [level] uM23Level	Urg(N)/Alr(M)/Resp(W/U) [23] Level,
--	AT [time] DESCEND TO [level] uM24TimeLevel	Urg(N)/Alr(M)/Resp(W/U) [24] TimeLevel,
--	AT [position] DESCEND TO [level] uM25PositionLevel	Urg(N)/Alr(M)/Resp(W/U) [25] PositionLevel,
--	CLIMB TO REACH [level] BY [time] uM26LevelTime	Urg(N)/Alr(M)/Resp(W/U) [26] LevelTime,
--	CLIMB TO REACH [level] BY [position] uM27LevelPosition	Urg(N)/Alr(M)/Resp(W/U) [27] LevelPosition,
--	DESCEND TO REACH [level] BY [time] uM28LevelTime	Urg(N)/Alr(M)/Resp(W/U) [28] LevelTime,
--	DESCEND TO REACH [level] BY [position] uM29LevelPosition	Urg(N)/Alr(M)/Resp(W/U) [29] LevelPosition,

--	MAINTAIN BLOCK [level] TO [level] uM30LevelLevel	Urg(N)/Alr(M)/Resp(W/U) [30] LevelLevel,
--	CLIMB TO AND MAINTAIN BLOCK [level] TO [level] uM31LevelLevel	Urg(N)/Alr(M)/Resp(W/U) [31] LevelLevel,
--	DESCEND TO AND MAINTAIN BLOCK [level] TO [level] uM32LevelLevel	Urg(N)/Alr(M)/Resp(W/U) [32] LevelLevel,
--	<i>Reserved</i> uM33NULL	Urg(L)/Alr(L)/Resp(Y) [33] NULL,
--	CRUISE CLIMB TO [level] uM34Level	Urg(N)/Alr(M)/Resp(W/U) [34] Level,
--	CRUISE CLIMB ABOVE [level] uM35Level	Urg(N)/Alr(M)/Resp(W/U) [35] Level,
--	EXPEDITE CLIMB TO [level] uM36Level	Urg(U)/Alr(M)/Resp(W/U) [36] Level,
--	EXPEDITE DESCENT TO [level] uM37Level	Urg(U)/Alr(M)/Resp(W/U) [37] Level,
--	IMMEDIATELY CLIMB TO [level] uM38Level	Urg(D)/Alr(H)/Resp(W/U) [38] Level,
--	IMMEDIATELY DESCEND TO [level] uM39Level	Urg(D)/Alr(H)/Resp(W/U) [39] Level,
--	<i>Reserved</i> uM40NULL	Urg(L)/Alr(L)/Resp(Y) [40] NULL,
--	<i>Reserved</i> uM41NULL	Urg(L)/Alr(L)/Resp(Y) [41] NULL,
--	EXPECT TO CROSS [position] AT [level] uM42PositionLevel	Urg(L)/Alr(L)/Resp(R) [42] PositionLevel,
--	EXPECT TO CROSS [position] AT OR ABOVE [level] uM43PositionLevel	Urg(L)/Alr(L)/Resp(R) [43] PositionLevel,
--	EXPECT TO CROSS [position] AT OR BELOW [level] uM44PositionLevel	Urg(L)/Alr(L)/Resp(R) [44] PositionLevel,

--	EXPECT TO CROSS [position] AT AND MAINTAIN [level]	
--	uM45PositionLevel	Urg(L)/Alr(L)/Resp(R) [45] PositionLevel,
--	CROSS [position] AT [level]	
--	uM46PositionLevel	Urg(N)/Alr(M)/Resp(W/U) [46] PositionLevel,
--	CROSS [position] AT OR ABOVE [level]	
--	uM47PositionLevel	Urg(N)/Alr(M)/Resp(W/U) [47] PositionLevel,
--	CROSS [position] AT OR BELOW [level]	
--	uM48PositionLevel	Urg(N)/Alr(M)/Resp(W/U) [48] PositionLevel,
--	CROSS [position] AT AND MAINTAIN [level]	
--	uM49PositionLevel	Urg(N)/Alr(M)/Resp(W/U) [49] PositionLevel,
--	CROSS [position] BETWEEN [level] AND [level]	
--	uM50PositionLevelLevel	Urg(N)/Alr(M)/Resp(W/U) [50] PositionLevelLevel,
--	CROSS [position] AT [time]	
--	uM51PositionTime	Urg(N)/Alr(M)/Resp(W/U) [51] PositionTime,
--	CROSS [position] AT OR BEFORE [time]	
--	uM52PositionTime	Urg(N)/Alr(M)/Resp(W/U) [52] PositionTime,
--	CROSS [position] AT OR AFTER [time]	
--	uM53PositionTime	Urg(N)/Alr(M)/Resp(W/U) [53] PositionTime,
--	CROSS [position] BETWEEN [time] AND [time]	
--	uM54PositionTimeTime	Urg(N)/Alr(M)/Resp(W/U) [54] PositionTimeTime,
--	CROSS [position] AT [speed]	
--	uM55PositionSpeed	Urg(N)/Alr(M)/Resp(W/U) [55] PositionSpeed,
--	CROSS [position] AT OR LESS THAN [speed]	
--	uM56PositionSpeed	Urg(N)/Alr(M)/Resp(W/U) [56] PositionSpeed,
--	CROSS [position] AT OR GREATER THAN [speed]	
--	uM57PositionSpeed	Urg(N)/Alr(M)/Resp(W/U) [57] PositionSpeed,
--	CROSS [position] AT [time] AT [level]	
--	uM58PositionTimeLevel	Urg(N)/Alr(M)/Resp(W/U) [58] PositionTimeLevel,
--	CROSS [position] AT OR BEFORE [time] AT [level]	
--	uM59PositionTimeLevel	Urg(N)/Alr(M)/Resp(W/U) [59] PositionTimeLevel,
--	CROSS [position] AT OR AFTER [time] AT [level]	
--	uM60PositionTimeLevel	Urg(N)/Alr(M)/Resp(W/U) [60] PositionTimeLevel,

--	CROSS [position] AT AND MAINTAIN [level] AT [speed]	
--	uM61PositionLevelSpeed	Urg(N)/Alr(M)/Resp(W/U) [61] PositionLevelSpeed,
--	AT [time] CROSS [position] AT AND MAINTAIN [level]	
--	uM62TimePositionLevel	Urg(N)/Alr(M)/Resp(W/U) [62] TimePositionLevel,
--	AT [time] CROSS [position] AT AND MAINTAIN [level] AT [speed]	
--	uM63TimePositionLevelSpeed	Urg(N)/Alr(M)/Resp(W/U) [63] TimePositionLevelSpeed,
--	OFFSET [specifiedDistance] [direction] OF ROUTE	
--	uM64DistanceSpecifiedDirection	Urg(N)/Alr(M)/Resp(W/U) [64] DistanceSpecifiedDirection,
--	AT [position] OFFSET [specifiedDistance] [direction] OF ROUTE	
--	uM65PositionDistanceSpecifiedDirection	Urg(N)/Alr(M)/Resp(W/U) [65] PositionDistanceSpecifiedDirection,
--	AT [time] OFFSET [specifiedDistance] [direction] OF ROUTE	
--	uM66TimeDistanceSpecifiedDirection	Urg(N)/Alr(M)/Resp(W/U) [66] TimeDistanceSpecifiedDirection,
--	PROCEED BACK ON ROUTE	
--	uM67NULL	Urg(N)/Alr(M)/Resp(W/U) [67] NULL,
--	REJOIN ROUTE BY [position]	
--	uM68Position	Urg(N)/Alr(M)/Resp(W/U) [68] Position,
--	REJOIN ROUTE BY [time]	
--	uM69Time	Urg(N)/Alr(M)/Resp(W/U) [69] Time,
--	EXPECT BACK ON ROUTE BY [position]	
--	uM70Position	Urg(L)/Alr(L)/Resp(R) [70] Position,
--	EXPECT BACK ON ROUTE BY [time]	
--	uM71Time	Urg(L)/Alr(L)/Resp(R) [71] Time,
--	RESUME OWN NAVIGATION	
--	uM72NULL	Urg(N)/Alr(M)/Resp(W/U) [72] NULL,
--	[DepartureClearance]	
--	uM73DepartureClearance	Urg(N)/Alr(M)/Resp(W/U) [73] DepartureClearance,
--	PROCEED DIRECT TO [position]	
--	uM74Position	Urg(N)/Alr(M)/Resp(W/U) [74] Position,

--	WHEN ABLE PROCEED DIRECT TO [position] uM75Position	Urg(N)/Alr(M)/Resp(W/U) [75] Position,
--	AT [time] PROCEED DIRECT TO [position] uM76TimePosition	Urg(N)/Alr(M)/Resp(W/U) [76] TimePosition,
--	AT [position] PROCEED DIRECT TO [position] uM77PositionPosition	Urg(N)/Alr(M)/Resp(W/U) [77] PositionPosition,
--	AT [level] PROCEED DIRECT TO [position] uM78LevelPosition	Urg(N)/Alr(M)/Resp(W/U) [78] LevelPosition,
--	CLEARED TO [position] VIA [routeClearance] uM79PositionRouteClearance	Urg(N)/Alr(M)/Resp(W/U) [79] PositionRouteClearanceIndex,
--	CLEARED [routeClearance] uM80RouteClearance	Urg(N)/Alr(M)/Resp(W/U) [80] RouteClearanceIndex,
--	CLEARED [procedureName] uM81ProcedureName	Urg(N)/Alr(M)/Resp(W/U) [81] ProcedureName,
--	CLEARED TO DEVIATE UP TO [specifiedDistance] [direction] OF ROUTE uM82DistanceSpecifiedDirection	Urg(N)/Alr(M)/Resp(W/U) [82] DistanceSpecifiedDirection,
--	AT [position] CLEARED [routeClearance] uM83PositionRouteClearance	Urg(N)/Alr(M)/Resp(W/U) [83] PositionRouteClearanceIndex,
--	AT [position] CLEARED [procedureName] uM84PositionProcedureName	Urg(N)/Alr(M)/Resp(W/U) [84] PositionProcedureName,
--	EXPECT [routeClearance] uM85RouteClearance	Urg(L)/Alr(L)/Resp(R) [85] RouteClearanceIndex,
--	AT [position] EXPECT [routeClearance] uM86PositionRouteClearance	Urg(L)/Alr(L)/Resp(R) [86] PositionRouteClearanceIndex,
--	EXPECT DIRECT TO [position] uM87Position	Urg(L)/Alr(L)/Resp(R) [87] Position,
--	AT [position] EXPECT DIRECT TO [position] uM88PositionPosition	Urg(L)/Alr(L)/Resp(R) [88] PositionPosition,
--	AT [time] EXPECT DIRECT TO [position] uM89TimePosition	Urg(L)/Alr(L)/Resp(R) [89] TimePosition,
--	AT [level] EXPECT DIRECT TO [position] uM90LevelPosition	Urg(L)/Alr(L)/Resp(R) [90] LevelPosition,

--	HOLD AT [position] MAINTAIN [level] INBOUND TRACK [degrees][direction]	
--	TURNS [legtype]	Urg(N)/Alr(M)/Resp(W/U)
	uM91HoldClearance	[91] HoldClearance,
--	HOLD AT [position] AS PUBLISHED MAINTAIN [level]	
--	uM92PositionLevel	Urg(N)/Alr(M)/Resp(W/U)
		[92] PositionLevel,
--	EXPECT FURTHER CLEARANCE AT [time]	Urg(L)/Alr(L)/Resp(R)
	uM93Time	[93] Time,
--	TURN [direction] HEADING [degrees]	Urg(N)/Alr(M)/Resp(W/U)
	uM94DirectionDegrees	[94] DirectionDegrees,
--	TURN [direction] GROUND TRACK [degrees]	Urg(N)/Alr(M)/Resp(W/U)
	uM95DirectionDegrees	[95] DirectionDegrees,
--	CONTINUE PRESENT HEADING	Urg(N)/Alr(M)/Resp(W/U)
	uM96NULL	[96] NULL,
--	AT [position] FLY HEADING [degrees]	Urg(N)/Alr(M)/Resp(W/U)
	uM97PositionDegrees	[97] PositionDegrees,
--	IMMEDIATELY TURN [direction] HEADING [degrees]	
--	uM98DirectionDegrees	Urg(D)/Alr(H)/Resp(W/U)
		[98] DirectionDegrees,
--	EXPECT [procedureName]	Urg(L)/Alr(L)/Resp(R)
	uM99ProcedureName	[99] ProcedureName,
--	AT [time] EXPECT [speed]	Urg(L)/Alr(L)/Resp(R)
	uM100TimeSpeed	[100] TimeSpeed,
--	AT [position] EXPECT [speed]	Urg(L)/Alr(L)/Resp(R)
	uM101PositionSpeed	[101] PositionSpeed,
--	AT [level] EXPECT [speed]	Urg(L)/Alr(L)/Resp(R)
	uM102LevelSpeed	[102] LevelSpeed,
--	AT [time] EXPECT [speed] TO [speed]	Urg(L)/Alr(L)/Resp(R)
	uM103TimeSpeedSpeed	[103] TimeSpeedSpeed,
--	AT [position] EXPECT [speed] TO [speed]	Urg(L)/Alr(L)/Resp(R)
	uM104PositionSpeedSpeed	[104] PositionSpeedSpeed,
--	AT [level] EXPECT [speed] TO [speed]	Urg(L)/Alr(L)/Resp(R)
	uM105LevelSpeedSpeed	[105] LevelSpeedSpeed,

--	MAINTAIN [speed] uM106Speed	Urg(N)/Alr(M)/Resp(W/U) [106] Speed,
--	MAINTAIN PRESENT SPEED uM107NULL	Urg(N)/Alr(M)/Resp(W/U) [107] NULL,
--	MAINTAIN [speed] OR GREATER uM108Speed	Urg(N)/Alr(M)/Resp(W/U) [108] Speed,
--	MAINTAIN [speed] OR LESS uM109Speed	Urg(N)/Alr(M)/Resp(W/U) [109] Speed,
--	MAINTAIN [speed] TO [speed] uM110SpeedSpeed	Urg(N)/Alr(M)/Resp(W/U) [110] SpeedSpeed,
--	INCREASE SPEED TO [speed] uM111Speed	Urg(N)/Alr(M)/Resp(W/U) [111] Speed,
--	INCREASE SPEED TO [speed] OR GREATER uM112Speed	Urg(N)/Alr(M)/Resp(W/U) [112] Speed,
--	REDUCE SPEED TO [speed] uM113Speed	Urg(N)/Alr(M)/Resp(W/U) [113] Speed,
--	REDUCE SPEED TO [speed] OR LESS uM114Speed	Urg(N)/Alr(M)/Resp(W/U) [114] Speed,
--	DO NOT EXCEED [speed] uM115Speed	Urg(N)/Alr(M)/Resp(W/U) [115] Speed,
--	RESUME NORMAL SPEED uM116NULL	Urg(N)/Alr(M)/Resp(W/U) [116] NULL,
--	CONTACT [unitname] [frequency] uM117UnitNameFrequency	Urg(N)/Alr(M)/Resp(W/U) [117] UnitNameFrequency,
--	AT [position] CONTACT [unitname] [frequency] uM118PositionUnitNameFrequency	Urg(N)/Alr(M)/Resp(W/U) [118] PositionUnitNameFrequency,
--	AT [time] CONTACT [unitname] [frequency] uM119TimeUnitNameFrequency	Urg(N)/Alr(M)/Resp(W/U) [119] TimeUnitNameFrequency,
--	MONITOR [unitname] [frequency] uM120UnitNameFrequency	Urg(N)/Alr(M)/Resp(W/U) [120] UnitNameFrequency,
--	AT [position] MONITOR [unitname] [frequency] uM121PositionUnitNameFrequency	Urg(N)/Alr(M)/Resp(W/U) [121] PositionUnitNameFrequency,

--	AT [time] MONITOR [unitname] [frequency] uM122TimeUnitNameFrequency	Urg(N)/Alr(M)/Resp(W/U) [122] TimeUnitNameFrequency,
--	SQUAWK [code] uM123Code	Urg(N)/Alr(M)/Resp(W/U) [123] Code,
--	STOP SQUAWK uM124NULL	Urg(N)/Alr(M)/Resp(W/U) [124] NULL,
--	SQUAWK MODE CHARLIE uM125NULL	Urg(N)/Alr(M)/Resp(W/U) [125] NULL,
--	STOP SQUAWK MODE CHARLIE uM126NULL	Urg(N)/Alr(M)/Resp(W/U) [126] NULL,
--	REPORT BACK ON ROUTE uM127NULL	Urg(N)/Alr(L)/Resp(W/U) [127] NULL,
--	REPORT LEAVING [level] uM128Level	Urg(N)/Alr(L)/Resp(W/U) [128] Level,
--	REPORT MAINTAINING [level] uM129Level	Urg(N)/Alr(L)/Resp(W/U) [129] Level,
--	REPORT PASSING [position] uM130Position	Urg(N)/Alr(L)/Resp(W/U) [130] Position,
--	REPORT REMAINING FUEL AND PERSONS ON BOARD uM131NULL	Urg(U)/Alr(M)/Resp(Y) [131] NULL,
--	REPORT POSITION uM132NULL	Urg(N)/Alr(M)/Resp(Y) [132] NULL,
--	REPORT PRESENT LEVEL uM133NULL	Urg(N)/Alr(M)/Resp(Y) [133] NULL,
--	REPORT [speedtype] [speedtype] [speedtype]SPEED uM134SpeedTypeSpeedTypeSpeedType	Urg(N)/Alr(M)/Resp(Y) [134] SpeedTypeSpeedTypeSpeedType,
--	CONFIRM ASSIGNED LEVEL uM135NULL	Urg(N)/Alr(L)/Resp(Y) [135] NULL,
--	CONFIRM ASSIGNED SPEED uM136NULL	Urg(N)/Alr(L)/Resp(Y) [136] NULL,
--	CONFIRM ASSIGNED ROUTE uM137NULL	Urg(N)/Alr(L)/Resp(Y) [137] NULL,

--	CONFIRM TIME OVER REPORTED WAYPOINT uM138NULL	Urg(N)/Alr(L)/Resp(Y) [138] NULL,
--	CONFIRM REPORTED WAYPOINT uM139NULL	Urg(N)/Alr(L)/Resp(Y) [139] NULL,
--	CONFIRM NEXT WAYPOINT uM140NULL	Urg(N)/Alr(L)/Resp(Y) [140] NULL,
--	CONFIRM NEXT WAYPOINT ETA uM141NULL	Urg(N)/Alr(L)/Resp(Y) [141] NULL,
--	CONFIRM ENSUING WAYPOINT uM142NULL	Urg(N)/Alr(L)/Resp(Y) [142] NULL,
--	CONFIRM REQUEST uM143NULL	Urg(N)/Alr(L)/Resp(Y) [143] NULL,
--	CONFIRM SQUAWK uM144NULL	Urg(N)/Alr(L)/Resp(Y) [144] NULL,
--	REPORT HEADING uM145NULL	Urg(N)/Alr(M)/Resp(Y) [145] NULL,
--	REPORT GROUND TRACK uM146NULL	Urg(N)/Alr(M)/Resp(Y) [146] NULL,
--	REQUEST POSITION REPORT uM147NULL	Urg(N)/Alr(M)/Resp(Y) [147] NULL,
--	WHEN CAN YOU ACCEPT [level] uM148Level	Urg(N)/Alr(L)/Resp(Y) [148] Level,
--	CAN YOU ACCEPT [level] AT [position] uM149LevelPosition	Urg(N)/Alr(L)/Resp(A/N) [149] LevelPosition,
--	CAN YOU ACCEPT [level] AT [time] uM150LevelTime	Urg(N)/Alr(L)/Resp(A/N) [150] LevelTime,
--	WHEN CAN YOU ACCEPT [speed] uM151Speed	Urg(N)/Alr(L)/Resp(Y) [151] Speed,
--	WHEN CAN YOU ACCEPT [specifiedDistance] [direction] OFFSET	
--	uM152DistanceSpecifiedOffsetDirection	Urg(N)/Alr(L)/Resp(Y) [152] DistanceSpecifiedDirection,
--	ALTIMETER [altimeter] uM153Altimeter	Urg(N)/Alr(L)/Resp(R) [153] Altimeter,

--	RADAR SERVICE TERMINATED uM154NULL	Urg(N)/Alr(L)/Resp(R) [154] NULL,
--	RADAR CONTACT [position] uM155Position	Urg(N)/Alr(M)/Resp(R) [155] Position,
--	RADAR CONTACT LOST uM156NULL	Urg(N)/Alr(M)/Resp(R) [156] NULL,
--	CHECK STUCK MICROPHONE [frequency] uM157Frequency	Urg(U)/Alr(M)/Resp(N) [157] Frequency,
--	ATIS [atiscode] uM158AtisCode	Urg(N)/Alr(L)/Resp(R) [158] ATISCode,
--	ERROR [errorInformation] uM159ErrorInformation	Urg(U)/Alr(M)/Resp(N) [159] ErrorInformation,
--	NEXT DATA AUTHORITY [facility] uM160Facility	Urg(L)/Alr(N)/Resp(N) [160] Facility,
--	END SERVICE uM161NULL	Urg(L)/Alr(N)/Resp(N) [161] NULL,
--	SERVICE UNAVAILABLE uM162NULL	Urg(L)/Alr(L)/Resp(N) [162] NULL,
--	[facilitydesignation] uM163FacilityDesignation	Urg(L)/Alr(N)/Resp(N) [163] FacilityDesignation,
--	WHEN READY uM164NULL	Urg(L)/Alr(N)/Resp(N) [164] NULL,
--	THEN uM165NULL	Urg(L)/Alr(N)/Resp(N) [165] NULL,
--	DUE TO [traffictype]TRAFFIC uM166TrafficType	Urg(L)/Alr(N)/Resp(N) [166] TrafficType,
--	DUE TO AIRSPACE RESTRICTION uM167NULL	Urg(L)/Alr(N)/Resp(N) [167] NULL,
--	DISREGARD uM168NULL	Urg(U)/Alr(M)/Resp(R) [168] NULL,
--	[freetext] uM169FreeText	Urg(N)/Alr(L)/Resp(R) [169] FreeText,

--	[freetext] uM170FreeText	Urg(D)/Alr(H)/Resp(R) [170] FreeText,
--	CLIMB AT [verticalRate] MINIMUM uM171VerticalRate	Urg(N)/Alr(M)/Resp(W/U) [171] VerticalRate,
--	CLIMB AT [verticalRate] MAXIMUM uM172VerticalRate	Urg(N)/Alr(M)/Resp(W/U) [172] VerticalRate,
--	DESCEND AT [verticalRate] MINIMUM uM173VerticalRate	Urg(N)/Alr(M)/Resp(W/U) [173] VerticalRate,
--	DESCEND AT [verticalRate] MAXIMUM uM174VerticalRate	Urg(N)/Alr(M)/Resp(W/U) [174] VerticalRate,
--	REPORT REACHING [level] uM175Level	Urg(N)/Alr(L)/Resp(W/U) [175] Level,
--	MAINTAIN OWN SEPARATION AND VMC uM176NULL	Urg(N)/Alr(M)/Resp(W/U) [176] NULL,
--	AT PILOTS DISCRETION uM177NULL	Urg(L)/Alr(L)/Resp(N) [177] NULL,
--	<i>Reserved</i> uM178NULL	Urg(L)/Alr(L)/Resp(Y) [178] NULL,
--	SQUAWK IDENT uM179NULL	Urg(N)/Alr(M)/Resp(W/U) [179] NULL,
--	REPORT REACHING BLOCK [level] TO [level] uM180LevelLevel	Urg(N)/Alr(L)/Resp(W/U) [180] LevelLevel,
--	REPORT DISTANCE [tofrom] [position] uM181ToFromPosition	Urg(N)/Alr(M)/Resp(Y) [181] ToFromPosition,
--	CONFIRM ATIS CODE uM182NULL	Urg(N)/Alr(L)/Resp(Y) [182] NULL,
--	[freetext] uM183FreeText	Urg(N)/Alr(M)/Resp(N) [183] FreeText,
--	AT [time] REPORT DISTANCE [tofrom] [position] uM184TimeToFromPosition	Urg(N)/Alr(L)/Resp(Y) [184] TimeToFromPosition,
--	AFTER PASSING [position] CLIMB TO [level] uM185PositionLevel	Urg(N)/Alr(M)/Resp(W/U) [185] PositionLevel,

--	AFTER PASSING [position] DESCEND TO [level] uM186PositionLevel	Urg(N)/Alr(M)/Resp(W/U) [186] PositionLevel,
--	[freetext] uM187FreeText	Urg(L)/Alr(N)/Resp(N) [187] FreeText,
--	AFTER PASSING [position] MAINTAIN [speed] uM188PositionSpeed	Urg(N)/Alr(M)/Resp(W/U) [188] PositionSpeed,
--	ADJUST SPEED TO [speed] uM189Speed	Urg(N)/Alr(M)/Resp(W/U) [189] Speed,
--	FLY HEADING [degrees] uM190Degrees	Urg(N)/Alr(M)/Resp(W/U) [190] Degrees,
--	ALL ATS TERMINATED uM191NULL	Urg(N)/Alr(M)/Resp(R) [191] NULL,
--	REACH [level] BY [time] uM192LevelTime	Urg(N)/Alr(M)/Resp(W/U) [192] LevelTime,
--	IDENTIFICATION LOST uM193NULL	Urg(N)/Alr(M)/Resp(R) [193] NULL,
--	[freetext] uM194FreeText	Urg(N)/Alr(L)/Resp(Y) [194] FreeText,
--	[freetext] uM195FreeText	Urg(L)/Alr(L)/Resp(R) [195] FreeText,
--	[freetext] uM196FreeText	Urg(N)/Alr(M)/Resp(W/U) [196] FreeText,
--	[freetext] uM197FreeText	Urg(U)/Alr(M)/Resp(W/U) [197] FreeText,
--	[freetext] uM198FreeText	Urg(D)/Alr(H)/Resp(W/U) [198] FreeText,
--	[freetext] uM199FreeText	Urg(N)/Alr(L)/Resp(N) [199] FreeText,
--	REPORT REACHING uM200NULL	Urg(N)/Alr(L)/Resp(W/U) [200] NULL,
--	<i>Not Used</i> uM201NULL	Urg(L)/Alr(L)/Resp(N) [201] NULL,

--	<i>Not Used</i> uM202NULL	Urg(L)/Alr(L)/Resp(N) [202] NULL,
--	[freetext] uM203FreeText	Urg(N)/Alr(M)/Resp(R) [203] FreeText,
--	[freetext] uM204FreeText	Urg(N)/Alr(M)/Resp(Y) [204] FreeText,
--	[freetext] uM205FreeText	Urg(N)/Alr(M)/Resp(A/N) [205] FreeText,
--	[freetext] uM206FreeText	Urg(L)/Alr(N)/Resp(Y) [206] FreeText,
--	[freetext] uM207FreeText	Urg(L)/Alr(L)/Resp(Y) [207] FreeText,
--	[freetext] uM208FreeText	Urg(L)/Alr(L)/Resp(N) [208] FreeText,
--	REACH [level] BY [position] uM209LevelPosition	Urg(N)/Alr(M)/Resp(W/U) [209] LevelPosition,
--	IDENTIFIED [position] uM210Position	Urg(N)/Alr(M)/Resp(R) [210] Position,
--	REQUEST FORWARDED uM211NULL	Urg(N)/Alr(L)/Resp(N) [211] NULL,
--	[facilitydesignation] ATIS [atiscode] CURRENT uM212FacilityDesignationATISCode	Urg(N)/Alr(L)/Resp(R) [212] FacilityDesignationATISCode,
--	[facilitydesignation] ALTIMETER [altimeter] uM213FacilityDesignationAltimeter	Urg(N)/Alr(L)/Resp(R) [213] FacilityDesignationAltimeter,
--	RVR RUNWAY [runway] [rvr] uM214RunwayRVR	Urg(N)/Alr(M)/Resp(R) [214] RunwayRVR,
--	TURN [direction][degrees] uM215DirectionDegrees	Urg(N)/Alr(M)/Resp(W/U) [215] DirectionDegrees,
--	REQUEST FLIGHT PLAN uM216NULL	Urg(N)/Alr(M)/Resp(Y) [216] NULL,
--	REPORT ARRIVAL uM217NULL	Urg(N)/Alr(M)/Resp(Y) [217] NULL,

--	REQUEST ALREADY RECEIVED uM218NULL	Urg(L)/Alr(N)/Resp(N) [218] NULL,
--	STOP CLIMB AT [level] uM219Level	Urg(U)/Alr(M)/Resp(W/U) [219] Level,
--	STOP DESCENT AT [level] uM220Level	Urg(U)/Alr(M)/Resp(W/U) [220] Level,
--	STOP TURN HEADING [degrees] uM221Degrees	Urg(U)/Alr(M)/Resp(W/U) [221] Degrees,
--	NO SPEED RESTRICTION uM222NULL	Urg(L)/Alr(L)/Resp(R) [222] NULL,
--	REDUCE TO MINIMUM APPROACH SPEED uM223NULL	Urg(N)/Alr(M)/Resp(W/U) [223] NULL,
--	NO DELAY EXPECTED uM224NULL	Urg(N)/Alr(L)/Resp(R) [224] NULL,
--	DELAY NOT DETERMINED uM225NULL	Urg(N)/Alr(L)/Resp(R) [225] NULL,
--	EXPECTED APPROACH TIME [time] uM226Time	Urg(N)/Alr(L)/Resp(R) [226] Time,
--	LOGICAL ACKNOWLEDGMENT uM227NULL	Urg(N)/Alr(M)/Resp(N) [227] NULL,
--	REPORT ETA [position] uM228Position	Urg(L)/Alr(L)/Resp(Y) [228] Position,
--	REPORT ALTERNATE AERODROME uM229NULL	Urg(L)/Alr(L)/Resp(Y) [229] NULL,
--	IMMEDIATELY uM230NULL	Urg(D)/Alr(H)/Resp(N) [230] NULL,
--	STATE PREFERRED LEVEL uM231NULL	Urg(L)/Alr(L)/Resp(Y) [231] NULL,
--	STATE TOP OF DESCENT uM232NULL	Urg(L)/Alr(L)/Resp(Y) [232] NULL,
--	USE OF LOGICAL ACKNOWLEDGMENT PROHIBITED uM233NULL	Urg(N)/Alr(M)/Resp(N) [233] NULL,

--	FLIGHT PLAN NOT HELD uM234NULL	Urg(L)/Alr(L)/Resp(N) [234] NULL,
--	ROGER 7500 uM235NULL	Urg(U)/Alr(H)/Resp(N) [235] NULL,
--	LEAVE CONTROLLED AIRSPACE uM236NULL	Urg(N)/Alr(M)/Resp(W/U) [236] NULL,
	...	
	}	

 -- Downlink message element

ATCDownlinkMsgElementId ::= CHOICE

	{	
--	WILCO dM0NULL	Urg(N)/Alr(M)/Resp(N) [0] NULL,
--	UNABLE dM1NULL	Urg(N)/Alr(M)/Resp(N) [1] NULL,
--	STANDBY dM2NULL	Urg(N)/Alr(M)/Resp(N) [2] NULL,
--	ROGER dM3NULL	Urg(N)/Alr(M)/Resp(N) [3] NULL,
--	AFFIRM dM4NULL	Urg(N)/Alr(M)/Resp(N) [4] NULL,
--	NEGATIVE dM5NULL	Urg(N)/Alr(M)/Resp(N) [5] NULL,
--	REQUEST [level] dM6Level	Urg(N)/Alr(L)/Resp(Y) [6] Level,
--	REQUEST BLOCK [level] TO [level] dM7LevelLevel	Urg(N)/Alr(L)/Resp(Y) [7] LevelLevel,
--	REQUEST CRUISE CLIMB TO [level] dM8Level	Urg(N)/Alr(L)/Resp(Y) [8] Level,
--	REQUEST CLIMB TO [level] dM9Level	Urg(N)/Alr(L)/Resp(Y) [9] Level,

--	REQUEST DESCENT TO [level] dM10Level	Urg(N)/Alr(L)/Resp(Y) [10] Level,
--	AT [position] REQUEST CLIMB TO [level] dM11PositionLevel	Urg(N)/Alr(L)/Resp(Y) [11] PositionLevel,
--	AT [position] REQUEST DESCENT TO [level] dM12PositionLevel	Urg(N)/Alr(L)/Resp(Y) [12] PositionLevel,
--	AT [time] REQUEST CLIMB TO [level] dM13TimeLevel	Urg(N)/Alr(L)/Resp(Y) [13] TimeLevel,
--	AT [time] REQUEST DESCENT TO [level] dM14TimeLevel	Urg(N)/Alr(L)/Resp(Y) [14] TimeLevel,
--	REQUEST OFFSET [specifiedDistance] [direction] OF ROUTE	
--	dM15DistanceSpecifiedDirection	Urg(N)/Alr(L)/Resp(Y) [15] DistanceSpecifiedDirection,
--	AT [position] REQUEST OFFSET [specifiedDistance] [direction] OF ROUTE	
--	dM16PositionDistanceSpecifiedDirection	Urg(N)/Alr(L)/Resp(Y) [16] PositionDistanceSpecifiedDirection,
--	AT [time] REQUEST OFFSET [specifiedDistance] [direction] OF ROUTE	
--	dM17TimeDistanceSpecifiedDirection	Urg(N)/Alr(L)/Resp(Y) [17] TimeDistanceSpecifiedDirection,
--	REQUEST [speed] dM18Speed	Urg(N)/Alr(L)/Resp(Y) [18] Speed,
--	REQUEST [speed] TO [speed] dM19SpeedSpeed	Urg(N)/Alr(L)/Resp(Y) [19] SpeedSpeed,
--	REQUEST VOICE CONTACT dM20NULL	Urg(N)/Alr(L)/Resp(Y) [20] NULL,
--	REQUEST VOICE CONTACT [frequency] dM21Frequency	Urg(N)/Alr(L)/Resp(Y) [21] Frequency,
--	REQUEST DIRECT TO [position] dM22Position	Urg(N)/Alr(L)/Resp(Y) [22] Position,
--	REQUEST [procedureName] dM23ProcedureName	Urg(N)/Alr(L)/Resp(Y) [23] ProcedureName,
--	REQUEST CLEARANCE [routeClearance] dM24RouteClearance	Urg(N)/Alr(L)/Resp(Y) [24] RouteClearanceIndex,

--	REQUEST [clearanceType] CLEARANCE dM25ClearanceType	Urg(N)/Alr(L)/Resp(Y) [25] ClearanceType,
--	REQUEST WEATHER DEVIATION TO [position] VIA [routeClearance] dM26PositionRouteClearance	Urg(N)/Alr(M)/Resp(Y) [26] PositionRouteClearanceIndex,
--	REQUEST WEATHER DEVIATION UP TO [specifiedDistance] [direction] OF ROUTE dM27DistanceSpecifiedDirection	Urg(N)/Alr(M)/Resp(Y) [27] DistanceSpecifiedDirection,
--	LEAVING [level] dM28Level	Urg(N)/Alr(L)/Resp(N) [28] Level,
--	CLIMBING TO [level] dM29Level	Urg(N)/Alr(L)/Resp(N) [29] Level,
--	DESCENDING TO [level] dM30Level	Urg(N)/Alr(L)/Resp(N) [30] Level,
--	PASSING [position] dM31Position	Urg(N)/Alr(L)/Resp(N) [31] Position,
--	PRESENT LEVEL [level] dM32Level	Urg(N)/Alr(L)/Resp(N) [32] Level,
--	PRESENT POSITION [position] dM33Position	Urg(N)/Alr(L)/Resp(N) [33] Position,
--	PRESENT SPEED [speed] dM34Speed	Urg(N)/Alr(L)/Resp(N) [34] Speed,
--	PRESENT HEADING [degrees] dM35Degrees	Urg(N)/Alr(L)/Resp(N) [35] Degrees,
--	PRESENT GROUND TRACK [degrees] dM36Degrees	Urg(N)/Alr(L)/Resp(N) [36] Degrees,
--	MAINTAINING [level] dM37Level	Urg(N)/Alr(L)/Resp(N) [37] Level,
--	ASSIGNED LEVEL [level] dM38Level	Urg(N)/Alr(M)/Resp(N) [38] Level,
--	ASSIGNED SPEED [speed] dM39Speed	Urg(N)/Alr(M)/Resp(N) [39] Speed,

--	ASSIGNED ROUTE [routeClearance] dM40RouteClearance	Urg(N)/Alr(M)/Resp(N) [40] RouteClearanceIndex,
--	BACK ON ROUTE dM41NULL	Urg(N)/Alr(M)/Resp(N) [41] NULL,
--	NEXT WAYPOINT [position] dM42Position	Urg(N)/Alr(L)/Resp(N) [42] Position,
--	NEXT WAYPOINT ETA [time] dM43Time	Urg(N)/Alr(L)/Resp(N) [43] Time,
--	ENSUING WAYPOINT [position] dM44Position	Urg(N)/Alr(L)/Resp(N) [44] Position,
--	REPORTED WAYPOINT [position] dM45Position	Urg(N)/Alr(L)/Resp(N) [45] Position,
--	REPORTED WAYPOINT [time] dM46Time	Urg(N)/Alr(L)/Resp(N) [46] Time,
--	SQUAWKING [code] dM47Code	Urg(N)/Alr(L)/Resp(N) [47] Code,
--	POSITION REPORT [positionreport] dM48PositionReport	Urg(N)/Alr(M)/Resp(N) [48] PositionReport,
--	WHEN CAN WE EXPECT [speed] dM49Speed	Urg(L)/Alr(L)/Resp(Y) [49] Speed,
--	WHEN CAN WE EXPECT [speed] TO [speed] dM50SpeedSpeed	Urg(L)/Alr(L)/Resp(Y) [50] SpeedSpeed,
--	WHEN CAN WE EXPECT BACK ON ROUTE dM51NULL	Urg(L)/Alr(L)/Resp(Y) [51] NULL,
--	WHEN CAN WE EXPECT LOWER LEVEL dM52NULL	Urg(L)/Alr(L)/Resp(Y) [52] NULL,
--	WHEN CAN WE EXPECT HIGHER LEVEL dM53NULL	Urg(L)/Alr(L)/Resp(Y) [53] NULL,
--	WHEN CAN WE EXPECT CRUISE CLIMB TO [level] dM54Level	Urg(L)/Alr(L)/Resp(Y) [54] Level,
--	PAN PAN PAN dM55NULL	Urg(U)/Alr(H)/Resp(Y) [55] NULL,

--	MAYDAY MAYDAY MAYDAY dM56NULL	Urg(D)/Alr(H)/Resp(Y) [56] NULL,
--	[remainingFuel] OF FUEL REMAINING AND [personsonboard] PERSONS ON BOARD	
--	dM57RemainingFuelPersonsOnBoard	Urg(U)/Alr(H)/Resp(Y) [57] RemainingFuelPersonsOnBoard,
--	CANCEL EMERGENCY dM58NULL	Urg(U)/Alr(M)/Resp(Y) [58] NULL,
--	DIVERTING TO [position] VIA [routeClearance] dM59PositionRouteClearance	Urg(U)/Alr(H)/Resp(Y) [59] PositionRouteClearanceIndex,
--	OFFSETTING [specifiedDistance] [direction] OF ROUTE dM60DistanceSpecifiedDirection	Urg(U)/Alr(H)/Resp(Y) [60] DistanceSpecifiedDirection,
--	DESCENDING TO [level] dM61Level	Urg(U)/Alr(H)/Resp(Y) [61] Level,
--	ERROR [errorInformation] dM62ErrorInformation	Urg(U)/Alr(L)/Resp(N) [62] ErrorInformation,
--	NOT CURRENT DATA AUTHORITY dM63NULL	Urg(L)/Alr(L)/Resp(N) [63] NULL,
--	[facilitydesignation] dM64FacilityDesignation	Urg(L)/Alr(L)/Resp(N) [64] FacilityDesignation,
--	DUE TO WEATHER dM65NULL	Urg(L)/Alr(L)/Resp(N) [65] NULL,
--	DUE TO AIRCRAFT PERFORMANCE dM66NULL	Urg(L)/Alr(L)/Resp(N) [66] NULL,
--	[freetext] dM67FreeText	Urg(N)/Alr(L)/Resp(N) [67] FreeText,
--	[freetext] dM68FreeText	Urg(D)/Alr(H)/Resp(Y) [68] FreeText,
--	REQUEST VMC DESCENT dM69NULL	Urg(N)/Alr(L)/Resp(Y) [69] NULL,
--	REQUEST HEADING [degrees] dM70Degrees	Urg(N)/Alr(L)/Resp(Y) [70] Degrees,
--	REQUEST GROUND TRACK [degrees] dM71Degrees	Urg(N)/Alr(L)/Resp(Y) [71] Degrees,

--	REACHING [level] dM72Level	Urg(N)/Alr(L)/Resp(N) [72] Level,
--	[versionnumber] dM73Versionnumber	Urg(L)/Alr(L)/Resp(N) [73] VersionNumber,
--	REQUEST TO MAINTAIN OWN SEPARATION AND VMC	
--	dM74NULL	Urg(L)/Alr(L)/Resp(Y) [74] NULL,
--	AT PILOTS DISCRETION dM75NULL	Urg(L)/Alr(L)/Resp(N) [75] NULL,
--	REACHING BLOCK [level] TO [level] dM76LevelLevel	Urg(N)/Alr(L)/Resp(N) [76] LevelLevel,
--	ASSIGNED BLOCK [level] TO [level] dM77LevelLevel	Urg(N)/Alr(M)/Resp(N) [77] LevelLevel,
--	AT [time] [distance] [tofrom] [position] dM78TimeDistanceToFromPosition	Urg(N)/Alr(L)/Resp(N) [78] TimeDistanceToFromPosition,
--	ATIS [atiscode] dM79AtisCode	Urg(N)/Alr(L)/Resp(N) [79] ATISCode,
--	DEVIATING UP TO [specifiedDistance] [direction] OF ROUTE	
--	dM80DistanceSpecifiedDirection	Urg(U)/Alr(H)/Resp(Y) [80] DistanceSpecifiedDirection,
--	WE CAN ACCEPT [level] AT [time] dM81LevelTime	Urg(L)/Alr(L)/Resp(N) [81] LevelTime,
--	WE CANNOT ACCEPT [level] dM82Level	Urg(L)/Alr(L)/Resp(N) [82] Level,
--	WE CAN ACCEPT [speed] AT [time] dM83SpeedTime	Urg(L)/Alr(L)/Resp(N) [83] SpeedTime,
--	WE CANNOT ACCEPT [speed] dM84Speed	Urg(L)/Alr(L)/Resp(N) [84] Speed,
--	WE CAN ACCEPT [specifiedDistance] [direction] AT [time]	
--	dM85DistanceSpecifiedDirectionTime	Urg(L)/Alr(L)/Resp(N) [85] DistanceSpecifiedDirectionTime,
--	WE CANNOT ACCEPT [specifiedDistance] [direction] dM86DistanceSpecifiedDirection	Urg(L)/Alr(L)/Resp(N) [86] DistanceSpecifiedDirection,

--	WHEN CAN WE EXPECT CLIMB TO [level] dM87Level	Urg(L)/Alr(L)/Resp(Y) [87] Level,
--	WHEN CAN WE EXPECT DESCENT TO [level] dM88Level	Urg(L)/Alr(L)/Resp(Y) [88] Level,
--	MONITORING [unitname] [frequency] dM89UnitnameFrequency	Urg(U)/Alr(M)/Resp(N) [89] UnitNameFrequency,
--	[freetext] dM90FreeText	Urg(N)/Alr(M)/Resp(N) [90] FreeText,
--	[freetext] dM91FreeText	Urg(N)/Alr(L)/Resp(Y) [91] FreeText,
--	[freetext] dM92FreeText	Urg(L)/Alr(L)/Resp(Y) [92] FreeText,
--	[freetext] dM93FreeText	Urg(U)/Alr(H)/Resp(N) [93] FreeText,
--	[freetext] dM94FreeText	Urg(D)/Alr(H)/Resp(N) [94] FreeText,
--	[freetext] dM95FreeText	Urg(U)/Alr(M)/Resp(N) [95] FreeText,
--	[freetext] dM96FreeText	Urg(U)/Alr(L)/Resp(N) [96] FreeText,
--	[freetext] dM97FreeText	Urg(L)/Alr(L)/Resp(N) [97] FreeText,
--	[freetext] dM98FreeText	Urg(N)/Alr(N)/Resp(N) [98] FreeText,
--	CURRENT DATA AUTHORITY dM99NULL	Urg(L)/Alr(L)/Resp(N) [99] NULL,
--	LOGICAL ACKNOWLEDGMENT dM100NULL	Urg(N)/Alr(M)/Resp(N) [100] NULL,
--	REQUEST END OF SERVICE dM101NULL	Urg(L)/Alr(L)/Resp(Y) [101] NULL,
--	LANDING REPORT dM102NULL	Urg(N)/Alr(N)/Resp(N) [102] NULL,

--	CANCELLING IFR dM103NULL	Urg(N)/Alr(L)/Resp(Y) [103] NULL,
--	ETA[position][time] dM104PositionTime	Urg(L)/Alr(L)/Resp(N) [104] PositionTime,
--	ALTERNATE AERODROME[airport] dM105Airport	Urg(L)/Alr(L)/Resp(N) [105] Airport,
--	PREFERRED LEVEL[level] dM106Level	Urg(L)/Alr(L)/Resp(N) [106] Level,
--	NOT AUTHORIZED NEXT DATA AUTHORITY dM107NULL	Urg(L)/Alr(L)/Resp(N) [107] NULL,
--	DE-ICING COMPLETE dM108NULL	Urg(L)/Alr(L)/Resp(N) [108] NULL,
--	TOP OF DESCENT [time] dM109Time	Urg(L)/Alr(L)/Resp(N) [109] Time,
--	TOP OF DESCENT [position] dM110Position	Urg(L)/Alr(L)/Resp(N) [110] Position,
--	TOP OF DESCENT [time] [position] dM111TimePosition	Urg(L)/Alr(L)/Resp(N) [111] TimePosition,
--	SQUAWKING 7500 dM112NULL	Urg(U)/Alr(H)/Resp(N) [112] NULL,
--	[speedType] [speedType] [speedType] SPEED [speed] dM113SpeedTypeSpeedTypeSpeedTypeSpeed	Urg(N)/Alr(L)/Resp(N) [113]SpeedTypeSpeedTypeSpeedTypeSpeed,
	...	
	}	

AircraftAddress ::= BIT STRING (SIZE(24))

AircraftFlightIdentification ::= IA5String (SIZE (2..8))

Airport ::= IA5String (SIZE (4))

Altimeter ::= CHOICE

{		
altimeterEnglish	[0]	AltimeterEnglish,
altimeterMetric	[1]	AltimeterMetric
}		

AltimeterEnglish ::= INTEGER (2200..3200)

-- unit = Inches Mercury, Range (22.00 .. 32.00), resolution = 0.01

AltimeterMetric ::= INTEGER (7500..12500)

-- unit = Hectopascal, Range (750.0..1250.0), resolution = 0.1

ATISCode ::= IA5String (SIZE (1))

ATSRouteDesignator ::= IA5String (SIZE (2..7))

ATWAlongTrackWaypoint ::= SEQUENCE

{			
position	[0]	Position,	
aTWDistance	[1]	ATWDistance,	
speed	[2]	Speed	OPTIONAL,
aTWLevels	[3]	ATWLevelSequence	OPTIONAL
}			

ATWLevel ::= SEQUENCE

{	
atw	ATWLevelTolerance,
level	Level
}	

ATWLevelSequence ::= SEQUENCE SIZE (1..2) OF ATWLevel

ATWLevelTolerance ::= ENUMERATED

{	
at	(0),
atorabove	(1),
atorbelow	(2)
}	

ATWDistance ::= SEQUENCE

{	
atwDistanceTolerance	ATWDistanceTolerance,
distance	Distance
}	

ATWDistanceTolerance ::= ENUMERATED

{	
plus	(0),
minus	(1)
}	

ClearanceType ::= ENUMERATED

```
{
  noneSpecified    (0),
  approach          (1),
  departure         (2),
  further           (3),
  start-up          (4),
  pushback          (5),
  taxi              (6),
  take-off          (7),
  landing           (8),
  oceanic           (9),
  en-route          (10),
  downstream        (11),
  ...
}
```

Code ::= SEQUENCE SIZE (4) OF CodeOctalDigit

CodeOctalDigit ::= INTEGER (0..7)

ControlledTime ::= SEQUENCE

```
{
  time              Time,
  timeTolerance     TimeTolerance
}
```

Date ::= SEQUENCE

```
{
  year              Year,
  month             Month,
  day               Day
}
```

DateTimeGroup ::= SEQUENCE

```
{
  date              Date,
  timehhmmss       Timehhmmss
}
```

Day ::= INTEGER (1..31)

--unit = Day, Range (1..31), resolution = 1

DegreeIncrement ::= INTEGER (1..20)

--unit = Degree, Range (1..20), resolution = 1

Degrees ::= CHOICE

```
{
  degreesMagnetic      [0]    DegreesMagnetic,
  degreesTrue          [1]    DegreesTrue
}
```

DegreesMagnetic ::= INTEGER (1..360)

--unit = degree, Range (1..360), resolution = 1

DegreesTrue ::= INTEGER (1..360)

--unit = degree, Range (1..360), resolution = 1

DepartureClearance ::= SEQUENCE

```
{
  aircraftFlightIdentification      [0]    AircraftFlightIdentification,
  clearanceLimit                    [1]    Position,
  flightInformation                  [2]    FlightInformation      OPTIONAL,
  furtherInstructions                [3]    FurtherInstructions    OPTIONAL
}
```

DepartureMinimumInterval ::= INTEGER (1..150)

--unit = Minute, Range (0.1..15.0), resolution = 0.1

Direction ::= ENUMERATED

```
{
  left           (0),
  right          (1),
  eitherSide     (2),
  north          (3),
  south          (4),
  east           (5),
  west           (6),
  northEast      (7),
  northWest      (8),
  southEast      (9),
  southWest      (10)
}
```

DirectionDegrees ::= SEQUENCE

```
{
  direction      Direction,
  degrees        Degrees
}
```

Distance ::= CHOICE

```
{
  distanceNm      [0]    DistanceNm,
  distanceKm      [1]    DistanceKm
}
```


DistanceKm ::= INTEGER (0..8000)

-- unit = Kilometer, Range (0..2000), resolution = 0.25

DistanceNm ::= INTEGER (0..9999)

-- unit = Nautical Mile, Range (0..999.9), resolution = 0.1

DistanceSpecified ::= CHOICE

```
{
  distanceSpecifiedNm    [0]    DistanceSpecifiedNm,
  distanceSpecifiedKm    [1]    DistanceSpecifiedKm
}
```

DistanceSpecifiedDirection ::= SEQUENCE

```
{
  distanceSpecified      DistanceSpecified,
  direction               Direction
}
```

DistanceSpecifiedDirectionTime ::= SEQUENCE

```
{
  distanceSpecifiedDirection      DistanceSpecifiedDirection,
  time                            Time
}
```

DistanceSpecifiedKm ::= INTEGER (1..500)

-- unit = Kilometer, Range (1..500), resolution = 1

DistanceSpecifiedNm ::= INTEGER (1..250)

-- unit = Nautical Mile, Range (1..250), resolution = 1

ErrorInformation ::= ENUMERATED

```
{
  unrecognizedMsgReferenceNumber      (0),
  logicalAcknowledgmentNotAccepted    (1),
  insufficientResources                (2),
  invalidMessageElementCombination    (3),
  invalidMessageElement               (4),
  ...
}
```

Facility ::= CHOICE

```
{
  noFacility                [0]    NULL,
  facilityDesignation       [1]    FacilityDesignation
}
```

FacilityDesignation ::= IA5String (SIZE (4..8))

FacilityFunction ::= ENUMERATED

```

{
  center           (0),
  approach         (1),
  tower            (2),
  final            (3),
  groundControl    (4),
  clearanceDelivery (5),
  departure        (6),
  control          (7),
  radio            (8),
  ...
}

```

FacilityDesignationAltimeter ::= SEQUENCE

```

{
  facilityDesignation FacilityDesignation,
  altimeter           Altimeter
}

```

FacilityDesignationATISCode ::= SEQUENCE

```

{
  facilityDesignation FacilityDesignation,
  aTISCode            ATISCode
}

```

FacilityIdentification ::= CHOICE

```

{
  facilityDesignation [0] FacilityDesignation,
  facilityName        [1] FacilityName
}

```

FacilityName ::= IA5String (SIZE (3..18))**Fix** ::= IA5String (SIZE (1..5))**FixName** ::= SEQUENCE

```

{
  name      [0] Fix,
  latlon    [1] LatitudeLongitude OPTIONAL
}

```

FlightInformation ::= CHOICE

```

{
  routeOfFlight [0] RouteInformation,
  levelsOfFlight [1] LevelsOfFlight,
  routeAndLevels [2] RouteAndLevels
}

```

FreeText ::= IA5String (SIZE (1..256))

Frequency ::= CHOICE

{		
frequencyhf	[0]	Frequencyhf,
frequencyvhf	[1]	Frequencyvhf,
frequencyuhf	[2]	Frequencyuhf,
frequenciesatchannel	[3]	Frequenciesatchannel
}		

Frequencyhf ::= INTEGER (2850..28000)

-- unit = Kilohertz, Range (2850..28000), resolution = 1

Frequenciesatchannel ::= NumericString (SIZE (12))

-- Frequenciesatchannel corresponds to a 12 digit telephone number

Frequencyuhf ::= INTEGER (9000..15999)

-- unit = Megahertz, Range (225.000..399.975), resolution = 0.025

Frequencyvhf ::= INTEGER (23600..27398)

-- unit = Megahertz, Range (118.000..136.990), resolution = 0.005

FurtherInstructions ::= SEQUENCE

{			
code	[0]	Code	OPTIONAL,
frequencyDeparture	[1]	UnitNameFrequency	OPTIONAL,
clearanceExpiryTime	[2]	Time	OPTIONAL,
airportDeparture	[3]	Airport	OPTIONAL,
airportDestination	[4]	Airport	OPTIONAL,
timeDeparture	[5]	TimeDeparture	OPTIONAL,
runwayDeparture	[6]	Runway	OPTIONAL,
revisionNumber	[7]	RevisionNumber	OPTIONAL,
aTISCode	[8]	ATISCode	OPTIONAL
}			

Holdatwaypoint ::= SEQUENCE

{			
position	[0]	Position,	
holdatwaypointspeedlow	[1]	Speed	OPTIONAL,
aTWlevel	[2]	ATWLevel	OPTIONAL,
holdatwaypointspeedhigh	[3]	Speed	OPTIONAL,
direction	[4]	Direction	OPTIONAL,
degrees	[5]	Degrees	OPTIONAL,
eFCtime	[6]	Time	OPTIONAL,
legtype	[7]	LegType	OPTIONAL
}			

HoldClearance ::= SEQUENCE

{			
position	[0]	Position,	
level	[1]	Level,	
degrees	[2]	Degrees,	
direction	[3]	Direction,	
legType	[4]	LegType	OPTIONAL
}			

Humidity ::= INTEGER (0..100)

-- unit = Percent humidity, Range (0..100), resolution = 1

InterceptCourseFrom ::= SEQUENCE

{		
fromSelection	InterceptCourseFromSelection,	
degrees	Degrees	
}		

InterceptCourseFromSelection ::= CHOICE

{		
publishedIdentifier	[0]	PublishedIdentifier,
latitudeLongitude	[1]	LatitudeLongitude,
placeBearingPlaceBearing	[2]	PlaceBearingPlaceBearing,
placeBearingDistance	[3]	PlaceBearingDistance
}		

Icing ::= ENUMERATED

{	
trace	(0),
light	(1),
moderate	(2),
severe	(3)
}	

Latitude ::= SEQUENCE

{	
latitudeType	LatitudeType,
latitudeDirection	LatitudeDirection
}	

LatitudeDegrees ::= INTEGER (0..90000)
 -- unit = Degree, Range (0..90), resolution = 0.001

LatitudeDegreesMinutes ::= SEQUENCE
 {
 latitudeWholeDegrees LatitudeWholeDegrees,
 minutesLatLon MinutesLatLon
 }

LatitudeDegreesMinutesSeconds ::= SEQUENCE
 {
 latitudeWholeDegrees LatitudeWholeDegrees,
 latlonWholeMinutes LatLonWholeMinutes,
 secondsLatLon SecondsLatLon
 }

LatitudeDirection ::= ENUMERATED
 {
 north (0),
 south (1)
 }

LatitudeWholeDegrees ::= INTEGER (0..89)
 -- unit = Degree, Range (0..89), resolution = 1

LatitudeLongitude ::= SEQUENCE
 {
 latitude [0] Latitude OPTIONAL,
 longitude [1] Longitude OPTIONAL
 }

LatitudeReportingPoints ::= SEQUENCE
 {
 latitudeDirection LatitudeDirection,
 latitudeDegrees LatitudeDegrees
 }

LatitudeType ::= CHOICE
 {
 latitudeDegrees [0] LatitudeDegrees,
 latitudeDegreesMinutes [1] LatitudeDegreesMinutes,
 latitudeDMS [2] LatitudeDegreesMinutesSeconds
 }

LatLonWholeMinutes ::= INTEGER (0..59)
 -- unit = Minute, Range (0..59), resolution = 1

LatLonReportingPoints ::= CHOICE

```
{
  latitudeReportingPoints      [0]    LatitudeReportingPoints,
  longitudeReportingPoints     [1]    LongitudeReportingPoints
}
```

LegDistance ::= CHOICE

```
{
  legDistanceEnglish          [0]    LegDistanceEnglish,
  legDistanceMetric           [1]    LegDistanceMetric
}
```

LegDistanceEnglish ::= INTEGER (0..50)

-- unit = Nautical Mile, Range (0..50), resolution = 1

LegDistanceMetric ::= INTEGER (1..128)

-- unit = Kilometer, Range (1..128), resolution = 1

LegTime ::= INTEGER (0..10)

--unit = Minute, Range (0..10), resolution = 1

LegType ::= CHOICE

```
{
  legDistance      [0]    LegDistance,
  legTime          [1]    LegTime
}
```

Level ::= CHOICE

```
{
  singleLevel      [0]    LevelType,
  blockLevel       [1]    SEQUENCE SIZE (2) OF LevelType
}
```

LevelFeet ::= INTEGER (-60..7000)

--unit = Feet, Range (-600..70000), resolution = 10

LevelFlightLevel ::= INTEGER (30..700)

--unit = Level (100 Feet), Range (030..700), resolution = 1

LevelFlightLevelMetric ::= INTEGER (100..2500)

--unit = Level (10 Meters), Range (100..2500), resolution = 1

LevelLevel ::= SEQUENCE SIZE (2) OF Level

LevelMeters ::= INTEGER (-30..25000)

--unit = Meter, Range (-30..25000), resolution = 1

LevelPosition ::= SEQUENCE

{		
level		Level,
position		Position
}		

LevelProcedureName ::= SEQUENCE

{		
level		Level,
procedureName		ProcedureName
}		

LevelsOfFlight ::= CHOICE

{		
level	[0]	Level,
procedureName	[1]	ProcedureName,
levelProcedureName	[2]	LevelProcedureName
}		

LevelSpeed ::= SEQUENCE

{		
level		Level,
speed		Speed
}		

LevelSpeedSpeed ::= SEQUENCE

{		
level		Level,
speeds		SpeedSpeed
}		

LevelTime ::= SEQUENCE

{		
level		Level,
time		Time
}		

LevelType ::= CHOICE

{		
levelFeet	[0]	LevelFeet,
levelMeters	[1]	LevelMeters,
levelFlightLevel	[2]	LevelFlightLevel,
levelFlightLevelMetric	[3]	LevelFlightLevelMetric
}		

Longitude ::= SEQUENCE

{		
longitudeType		LongitudeType,
longitudeDirection		LongitudeDirection
}		

LongitudeDegrees ::= INTEGER (0..180000)
 --unit = Degree, Range (0..180), resolution = 0.001

LongitudeDegreesMinutes ::= SEQUENCE
 {
 longitudeWholeDegrees LongitudeWholeDegrees,
 minutesLatLon MinutesLatLon
 }

LongitudeDegreesMinutesSeconds ::= SEQUENCE
 {
 longitudeWholeDegrees LongitudeWholeDegrees,
 latonWholeMinutes LatLonWholeMinutes,
 secondsLatLon SecondsLatLon
 }

LongitudeDirection ::= ENUMERATED
 {
 east (0),
 west (1)
 }

LongitudeWholeDegrees ::= INTEGER (0..179)
 -- unit = Degree, Range (0..179), resolution = 1

LongitudeReportingPoints ::= SEQUENCE
 {
 longitudeDirection LongitudeDirection,
 longitudeDegrees LongitudeDegrees
 }

LongitudeType ::= CHOICE
 {
 longitudeDegrees [0] LongitudeDegrees,
 longitudeDegreesMinutes [1] LongitudeDegreesMinutes,
 longitudeDMS [2] LongitudeDegreesMinutesSeconds
 }

MinutesLatLon ::= INTEGER (0..5999)
 --unit = Minute, Range (0..59.99), resolution = 0.01

Month ::= INTEGER (1..12)
 --unit = 1 Month, Range (1..12), resolution = 1

Navaid ::= SEQUENCE
 {
 name [0] NavaidName,
 latlon [1] LatitudeLongitude OPTIONAL
 }

NavaidName ::= IA5String (SIZE (1..4))

PersonsOnBoard ::= INTEGER (1..1024)

PlaceBearing ::= SEQUENCE

{	
publishedIdentifier	PublishedIdentifier,
degrees	Degrees
}	

PlaceBearingDistance ::= SEQUENCE

{	
publishedIdentifier	PublishedIdentifier,
degrees	Degrees,
distance	Distance
}	

PlaceBearingPlaceBearing ::= SEQUENCE SIZE (2) OF PlaceBearing

Position ::= CHOICE

{		
fixName	[0]	FixName,
navaid	[1]	Navaid,
airport	[2]	Airport,
latitudeLongitude	[3]	LatitudeLongitude,
placeBearingDistance	[4]	PlaceBearingDistance
}		

PositionDegrees ::= SEQUENCE

{	
position	Position,
degrees	Degrees
}	

PositionDistanceSpecifiedDirection ::= SEQUENCE

{	
position	Position,
distanceSpecifiedDirection	DistanceSpecifiedDirection
}	

PositionLevel ::= SEQUENCE

{	
position	Position,
level	Level
}	

PositionLevelLevel ::= SEQUENCE

```

{
    position      Position,
    levels        LevelLevel
}

```

PositionLevelSpeed ::= SEQUENCE

```

{
    positionlevel PositionLevel,
    speed         Speed
}

```

PositionPosition ::= SEQUENCE SIZE (2) OF Position**PositionProcedureName ::= SEQUENCE**

```

{
    position      Position,
    procedureName ProcedureName
}

```

PositionReport ::= SEQUENCE

```

{
    positioncurrent      [0]    Position,
    timeatpositioncurrent [1]    Time,
    level                [2]    Level,
    fixnext              [3]    Position OPTIONAL,
    timeetaatfixnext     [4]    Time OPTIONAL,
    fixnextplusone       [5]    Position OPTIONAL,
    timeetaatdestination [6]    Time OPTIONAL,
    remainingFuel        [7]    RemainingFuel OPTIONAL,
    temperature          [8]    Temperature OPTIONAL,
    winds                [9]    Winds OPTIONAL,
    turbulence           [10]   Turbulence OPTIONAL,
    icing                [11]   Icing OPTIONAL,
    speed                [12]   Speed OPTIONAL,
    speedground          [13]   SpeedGround OPTIONAL,
    verticalChange        [14]   VerticalChange OPTIONAL,
    trackAngle           [15]   Degrees OPTIONAL,
    heading              [16]   Degrees OPTIONAL,
    distance             [17]   Distance OPTIONAL,
    humidity              [18]   Humidity OPTIONAL,
    reportedWaypointPosition [19] Position OPTIONAL,
    reportedWaypointTime   [20] Time OPTIONAL,
    reportedWaypointLevel  [21] Level OPTIONAL
}

```

PositionRouteClearanceIndex ::= SEQUENCE

```
{
  position          Position,
  routeClearanceIndex RouteClearanceIndex
}
```

PositionSpeed ::= SEQUENCE

```
{
  position          Position,
  speed             Speed
}
```

PositionSpeedSpeed ::= SEQUENCE

```
{
  position          Position,
  speeds            SpeedSpeed
}
```

PositionTime ::= SEQUENCE

```
{
  position          Position,
  time              Time
}
```

PositionTimeLevel ::= SEQUENCE

```
{
  positionTime      PositionTime,
  level             Level
}
```

PositionTimeTime ::= SEQUENCE

```
{
  position          Position,
  times             TimeTime
}
```

PositionUnitNameFrequency ::= SEQUENCE

```
{
  position          Position,
  unitname          UnitName,
  frequency          Frequency
}
```

Procedure ::= IA5String (SIZE (1..20))

ProcedureName ::= SEQUENCE

{			
type	[0]	ProcedureType,	
procedure	[1]	Procedure,	
transition	[2]	ProcedureTransition	OPTIONAL
}			

ProcedureTransition ::= IA5String (SIZE (1..5))**ProcedureType** ::= ENUMERATED

{	
arrival	(0),
approach	(1),
departure	(2)
}	

PublishedIdentifier ::= CHOICE

{		
fixName	[0]	FixName,
navaid	[1]	Navaid
}		

RemainingFuel ::= Time**RemainingFuelPersonsOnBoard** ::= SEQUENCE

{		
remainingFuel	RemainingFuel,	
personsOnBoard	PersonsOnBoard	
}		

ReportingPoints ::= SEQUENCE

{			
latLonReportingPoints	[0]	LatLonReportingPoints,	
degreeIncrement	[1]	DegreeIncrement	OPTIONAL
}			

RevisionNumber ::= INTEGER (1..16)**RouteAndLevels** ::= SEQUENCE

{		
routeOfFlight	RouteInformation,	
levelsOfFlight	LevelsOfFlight	
}		

RouteClearance ::= SEQUENCE

{			
airportDeparture	[0]	Airport	OPTIONAL,
airportDestination	[1]	Airport	OPTIONAL,
runwayDeparture	[2]	Runway	OPTIONAL,
procedureDeparture	[3]	ProcedureName	OPTIONAL,
runwayArrival	[4]	Runway	OPTIONAL,
procedureApproach	[5]	ProcedureName	OPTIONAL,
procedureArrival	[6]	ProcedureName	OPTIONAL,
routeInformations	[7]	SEQUENCE SIZE (1..128) OF RouteInformation	OPTIONAL,
routeInformationAdditional	[8]	RouteInformationAdditional	OPTIONAL
}			

RouteClearanceIndex ::= INTEGER (1..2)

-- RouteClearanceIndex identifies the position of the RouteClearance data
 -- in the ASN.1 type for
 -- ATC UplinkMessage, constrained Data, routeClearance Data
 -- ATC DownlinkMessage, constrained Data, routeClearance Data

RouteInformation ::= CHOICE

{			
publishedIdentifier	[0]	PublishedIdentifier,	
latitudeLongitude	[1]	LatitudeLongitude,	
placeBearingPlaceBearing	[2]	PlaceBearingPlaceBearing,	
placeBearingDistance	[3]	PlaceBearingDistance,	
aTSRouteDesignator	[4]	ATSRRouteDesignator	
}			

RouteInformationAdditional ::= SEQUENCE

{			
aTWAlongTrackWaypoints	[0]	SEQUENCE SIZE (1..8) OF ATWAlongTrackWaypoint	OPTIONAL,
reportingpoints	[1]	ReportingPoints	OPTIONAL,
interceptCourseFroms	[2]	SEQUENCE SIZE (1..4) OF InterceptCourseFrom	OPTIONAL,
holdAtWaypoints	[3]	SEQUENCE SIZE (1..8) OF Holdatwaypoint	OPTIONAL,
waypointSpeedLevels	[4]	SEQUENCE SIZE (1..32) OF WaypointSpeedLevel	OPTIONAL,
rTARequiredTimeArrivals	[5]	SEQUENCE SIZE (1..32) OF RTARequiredTimeArrival	OPTIONAL
}			

RTARequiredTimeArrival ::= SEQUENCE

```
{
  position          [0]    Position,
  rTATime           [1]    RTATime,
  rTATolerance      [2]    RTATolerance      OPTIONAL
}
```

RTATime ::= SEQUENCE

```
{
  time              Time,
  timeTolerance     TimeTolerance
}
```

RTATolerance ::= INTEGER (1..150)

--unit= Minute, Range (0.1..15.0), resolution = 0.1

Runway ::= SEQUENCE

```
{
  direction          RunwayDirection,
  configuration       RunwayConfiguration
}
```

RunwayDirection ::= INTEGER (1..36)

RunwayConfiguration ::= ENUMERATED

```
{
  left              (0),
  right             (1),
  center            (2),
  none              (3)
}
```

RunwayRVR ::= SEQUENCE

```
{
  runway            Runway,
  rVR               RVR
}
```

RVR ::= CHOICE

```
{
  rVRFeet          [0]    RVRFeet,
  rVRMeters        [1]    RVRMeters
}
```

RVRFeet ::= INTEGER (0..6100)

-- unit = Feet, Range (0..6100), resolution = 1

RVRMeters ::= INTEGER (0..1500)
 -- unit = Meters (0..1500), resolution = 1

SecondsLatLon ::= INTEGER (0..59)
 --unit = Second, Range (0.. 59), resolution = 1

Speed ::= CHOICE
 {
 speedIndicated [0] SpeedIndicated,
 speedIndicatedMetric [1] SpeedIndicatedMetric,
 speedTrue [2] SpeedTrue,
 speedTrueMetric [3] SpeedTrueMetric,
 speedGround [4] SpeedGround,
 speedGroundMetric [5] SpeedGroundMetric,
 speedMach [6] SpeedMach
 }

SpeedIndicated ::= INTEGER (0..400)
 -- unit = Knots, Range (0..400), resolution = 1

SpeedIndicatedMetric ::= INTEGER (0..800)
 -- unit = Kilometers/Hour, Range (0..800), resolution = 1

SpeedGround ::= INTEGER (-50..2000)
 -- unit = Knots, Range (-50..2000), resolution = 1

SpeedGroundMetric ::= INTEGER (-100..4000)
 -- unit = Kilometers/Hour, Range (-100..4000), resolution = 1

SpeedMach ::= INTEGER (500..4000)
 -- unit = Mach Range (0.5 to 4.0), resolution = 0.001

SpeedSpeed ::= SEQUENCE SIZE (2) OF Speed

SpeedTime ::= SEQUENCE
 {
 speed Speed,
 time Time
 }

SpeedTrue ::= INTEGER (0..2000)
 -- unit = Knots, Range (0..2000), resolution = 1

SpeedTrueMetric ::= INTEGER (0..4000)
 -- unit = Kilometers/Hour, Range (0..4000), resolution = 1

SpeedType ::= ENUMERATED

```
{
  noneSpecified    (0),
  indicated         (1),
  true              (2),
  ground           (3),
  mach              (4),
  approach         (5),
  cruise            (6),
  minimum           (7),
  maximum           (8),
  ...
}
```

SpeedTypeSpeedTypeSpeedType ::= SEQUENCE SIZE (3) OF SpeedType

SpeedTypeSpeedTypeSpeedTypeSpeed ::= SEQUENCE

```
{
  speedTypes      SpeedTypeSpeedTypeSpeedType,
  speed            Speed
}
```

Temperature ::= INTEGER (-100..100)

-- unit = Degree Celsius, Range (-100..100), resolution = 1

Time ::= SEQUENCE

```
{
  hours            TimeHours,
  minutes          TimeMinutes
}
```

TimeLevel ::= SEQUENCE

```
{
  time             Time,
  level            Level
}
```

TimeDeparture ::= SEQUENCE

```
{
  timeDepartureAllocated      [0]    Time                OPTIONAL,
  timeDepartureControlled     [1]    ControlledTime       OPTIONAL,
  timeDepartureClearanceExpected [2]    Time                OPTIONAL,
  departureMinimumInterval    [3]    DepartureMinimumInterval OPTIONAL
}
```


TimeDistanceSpecifiedDirection ::= SEQUENCE

```
{
  time                Time,
  distanceSpecifiedDirection DistanceSpecifiedDirection
}
```

TimeDistanceToFromPosition ::= SEQUENCE

```
{
  time                Time,
  distance            Distance,
  tofrom             ToFrom,
  position            Position
}
```

Timehhmmss ::= SEQUENCE

```
{
  hoursminutes        Time,
  seconds             TimeSeconds
}
```

TimeHours ::= INTEGER (0..23)

-- unit = Hour, Range (0..23), resolution = 1

TimeUnitNameFrequency ::= SEQUENCE

```
{
  time                Time,
  unitName            UnitName,
  frequency           Frequency
}
```

TimeMinutes ::= INTEGER (0..59)

-- unit = Minute, Range (0..59), resolution = 1

TimePosition ::= SEQUENCE

```
{
  time                Time,
  position            Position
}
```

TimePositionLevel ::= SEQUENCE

```
{
  timeposition        TimePosition,
  level               Level
}
```

TimePositionLevelSpeed ::= SEQUENCE

```
{
  timeposition      TimePosition,
  levelspeed        LevelSpeed
}
```

TimeSeconds ::= INTEGER (0..59)

-- unit = Second, Range (0..59), resolution = 1

TimeSpeed ::= SEQUENCE

```
{
  time              Time,
  speed             Speed
}
```

TimeSpeedSpeed ::= SEQUENCE

```
{
  time              Time,
  speedspeed        SpeedSpeed
}
```

TimeTime ::= SEQUENCE SIZE (2) OF Time

TimeToFromPosition ::= SEQUENCE

```
{
  time              Time,
  tofrom            ToFrom,
  position          Position
}
```

TimeTolerance ::= ENUMERATED

```
{
  at                (0),
  atorafter         (1),
  atorbefore        (2)
}
```

ToFrom ::= ENUMERATED

```
{
  to                (0),
  from              (1)
}
```

ToFromPosition ::= SEQUENCE

```
{
  toFrom            ToFrom,
  position          Position
}
```

TrafficType ::= ENUMERATED

```
{
  noneSpecified      (0),
  oppositeDirection  (1),
  sameDirection      (2),
  converging         (3),
  crossing            (4),
  diverging          (5),
  ...
}
```

Turbulence ::= ENUMERATED

```
{
  light      (0),
  moderate   (1),
  severe     (2)
}
```

UnitName ::= SEQUENCE

```
{
  facilityDesignation [0] FacilityDesignation,
  facilityName        [1] FacilityName        OPTIONAL,
  facilityFunction    [2] FacilityFunction
}
```

UnitNameFrequency ::= SEQUENCE

```
{
  unitName      UnitName,
  frequency     Frequency
}
```

VersionNumber ::= INTEGER (0..15)

VerticalChange ::= SEQUENCE

```
{
  direction      VerticalDirection,
  rate           VerticalRate
}
```

VerticalDirection ::= ENUMERATED

```
{
  up      (0),
  down    (1)
}
```

VerticalRate ::= CHOICE

```
{
  verticalRateEnglish      [0]    VerticalRateEnglish,
  verticalRateMetric       [1]    VerticalRateMetric
}
```

VerticalRateEnglish ::= INTEGER (0..3000)

-- unit = Feet/Minute, Range (0..30000), resolution = 10

VerticalRateMetric ::= INTEGER (0..1000)

-- unit = Meters/Minute, Range (0..10000), resolution = 10

WaypointSpeedLevel ::= SEQUENCE

```
{
  position      [0]    Position,
  speed         [1]    Speed
  aTWLevels     [2]    ATWLevelSequence
}
OPTIONAL,
OPTIONAL
```

WindDirection ::= INTEGER (1..360)

-- unit = Degree, Range (1..360), resolution = 1

Winds ::= SEQUENCE

```
{
  direction      WindDirection,
  speed          WindSpeed
}
```

WindSpeed ::= CHOICE

```
{
  windSpeedEnglish      [0]    WindSpeedEnglish,
  windSpeedMetric       [1]    WindSpeedMetric
}
```

WindSpeedEnglish ::= INTEGER (0..255)

-- unit = Knot, Range (0..255), resolution = 1

WindSpeedMetric ::= INTEGER (0..511)

-- unit = Kilometer/Hour, Range (0..511), resolution = 1

Year ::= INTEGER (1996..2095)

-- unit = Year, Range (1996..2095), resolution = 1

END

2.3.5 PROTOCOL DEFINITION

2.3.5.1 Sequence Rules

2.3.5.1.1 With the exception of abort primitives, only the sequence of primitives illustrated in figures 2.3.5-1 to 2.3.5-18 shall be permitted.

Note 1.— The following figures define the valid sequences of primitives that are possible to be invoked during the operation of the CPDLC application. It shows the relationship in time between the service request and the resulting indication, and if applicable, the subsequent response and resulting confirmation.

Note 2.— Abort primitives may interrupt and terminate any of the normal message sequences outlined below.

Note 3.— Primitives are processed in the order received. See 4.4.3.

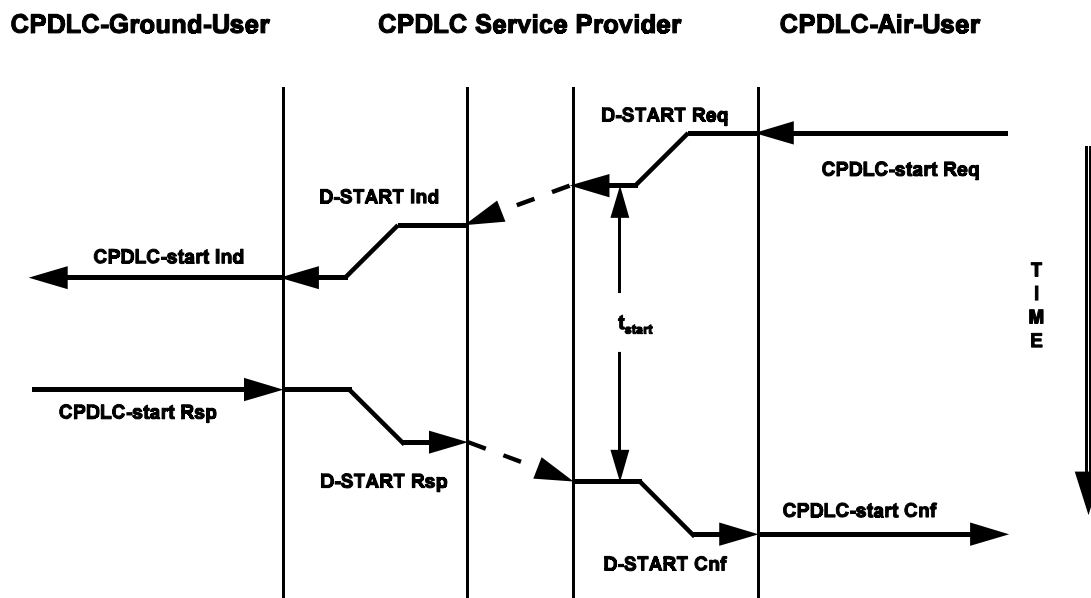


Figure 2.3.5-1. Sequence Diagram for CPDLC-start Service/Air Initiated

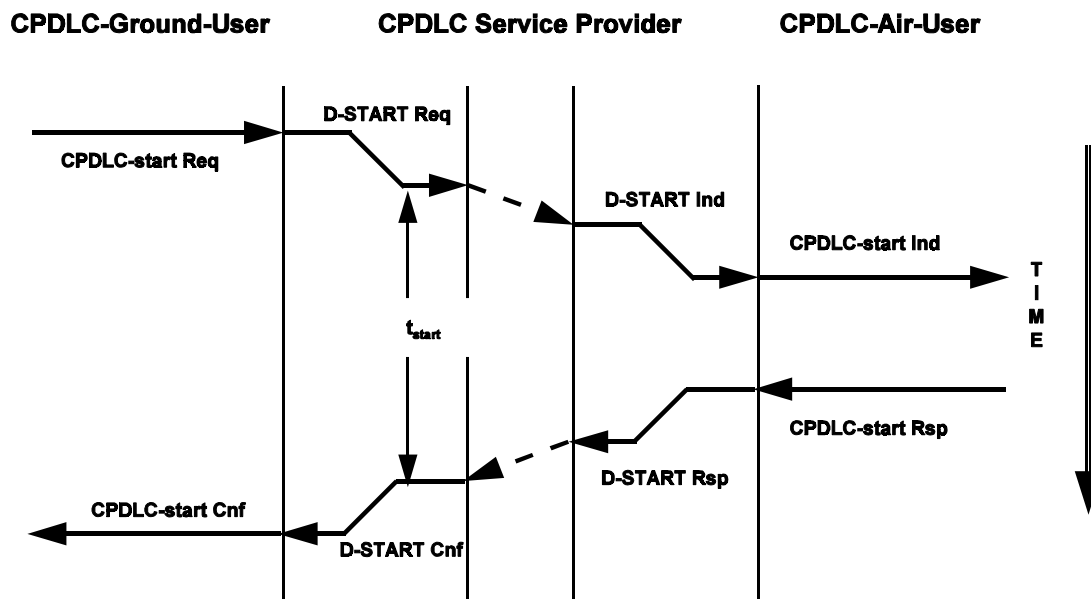


Figure 2.3.5-2. Sequence Diagram for CPDLC-start Service/Ground Initiated

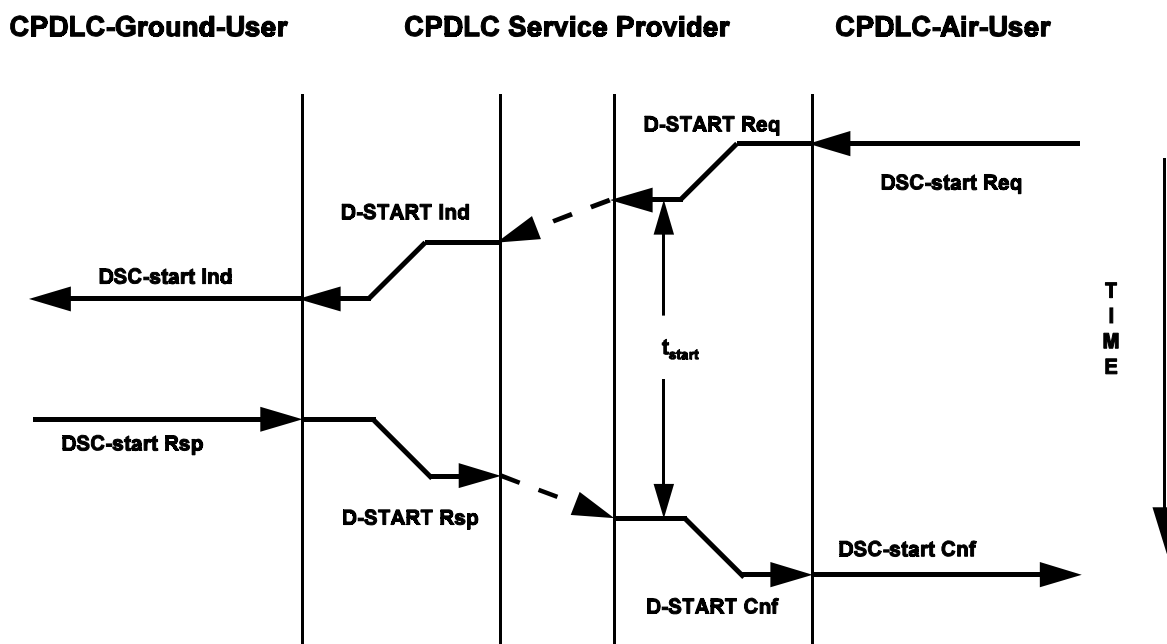


Figure 2.3.5-3. Sequence Diagram for DSC-start Service

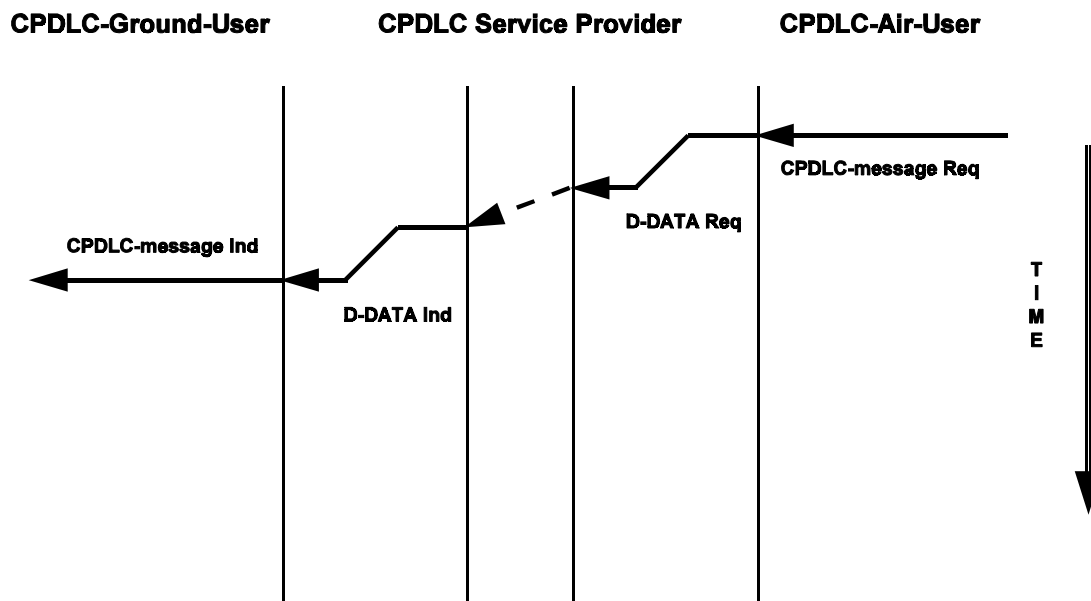


Figure 2.3.5-4. Sequence Diagram for CPDLC-message Service/Air Initiated

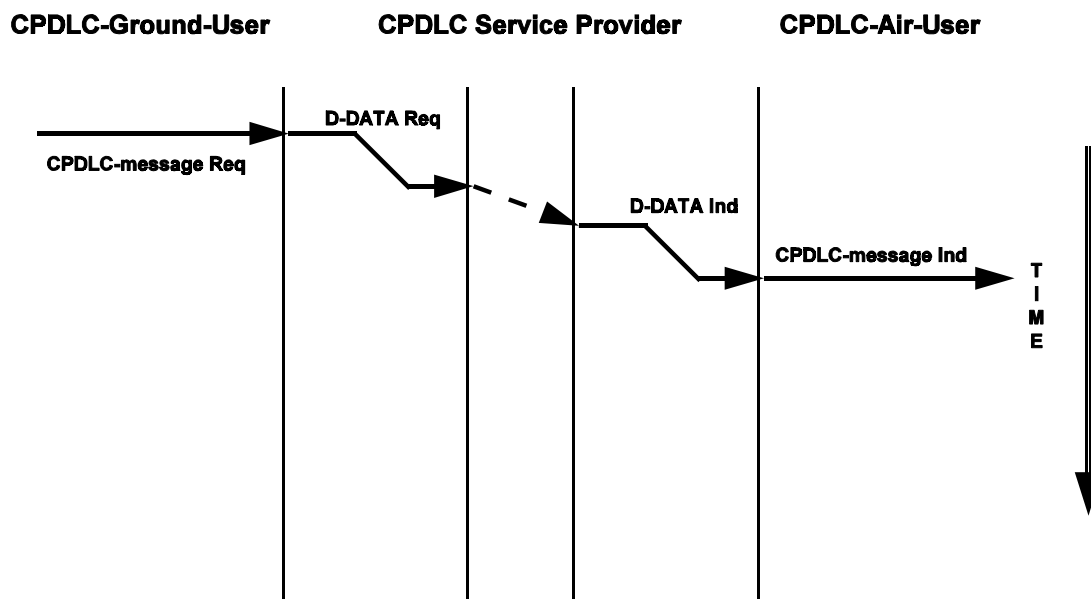


Figure 2.3.5-5. Sequence Diagram for CPDLC-message Service/Ground Initiated

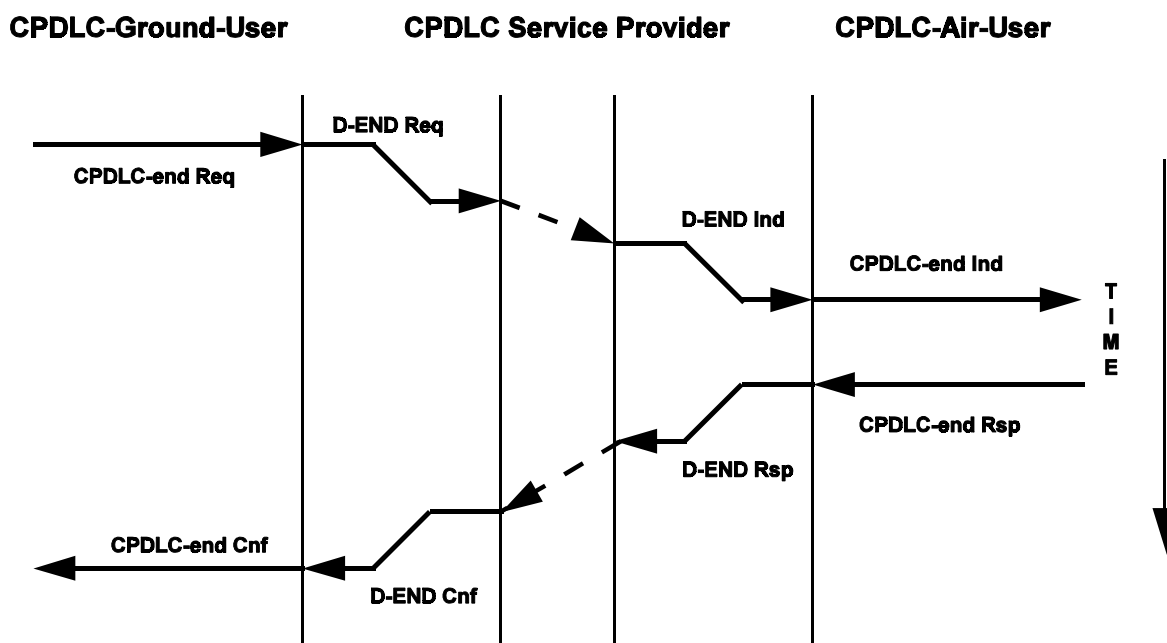


Figure 2.3.5-6. Sequence Diagram for CPDLC-end Service

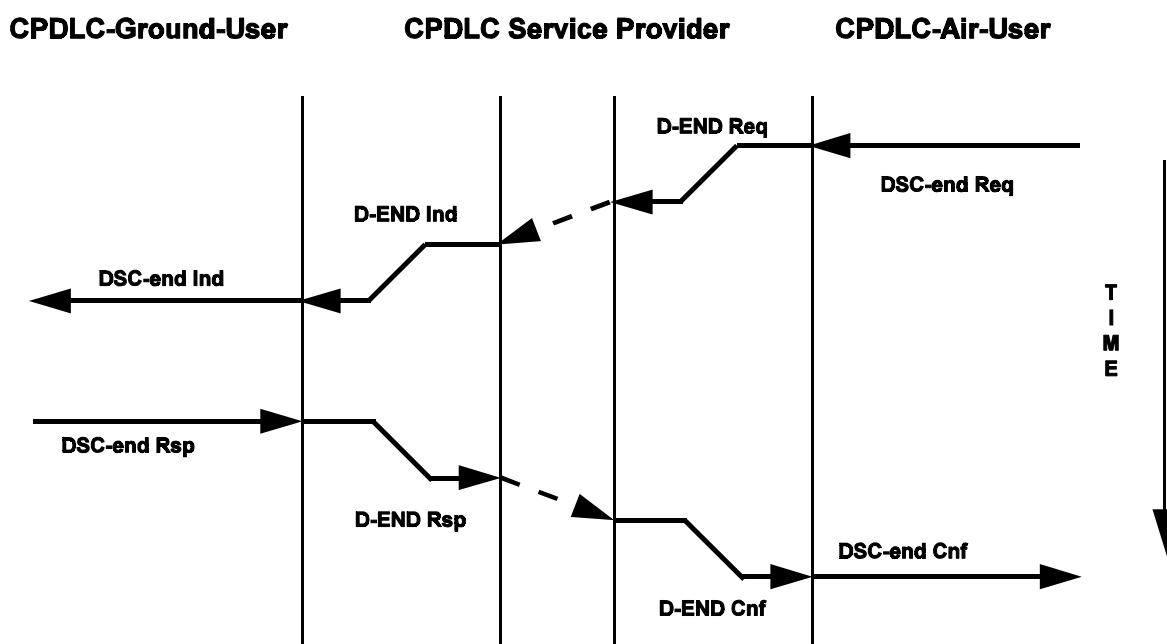


Figure 2.3.5-7. Sequence Diagram for DSC-end Service

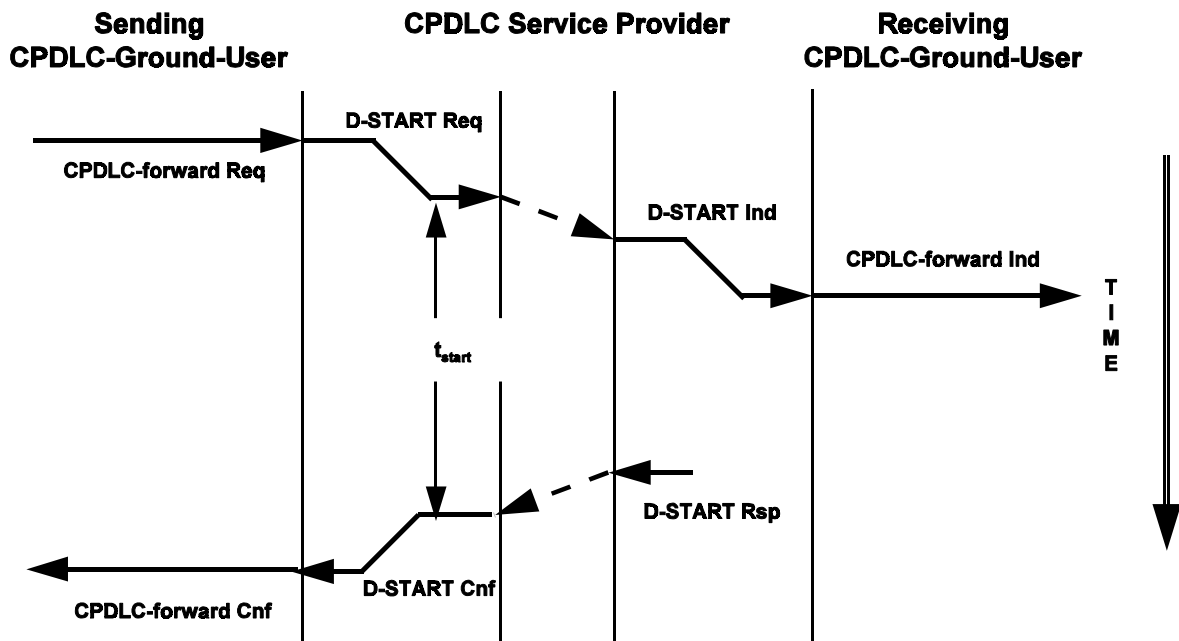


Figure 2.3.5-8. Sequence Diagram for CPDLC-forward Service/Ground Forwarding Supported, ASE Version Numbers the Same

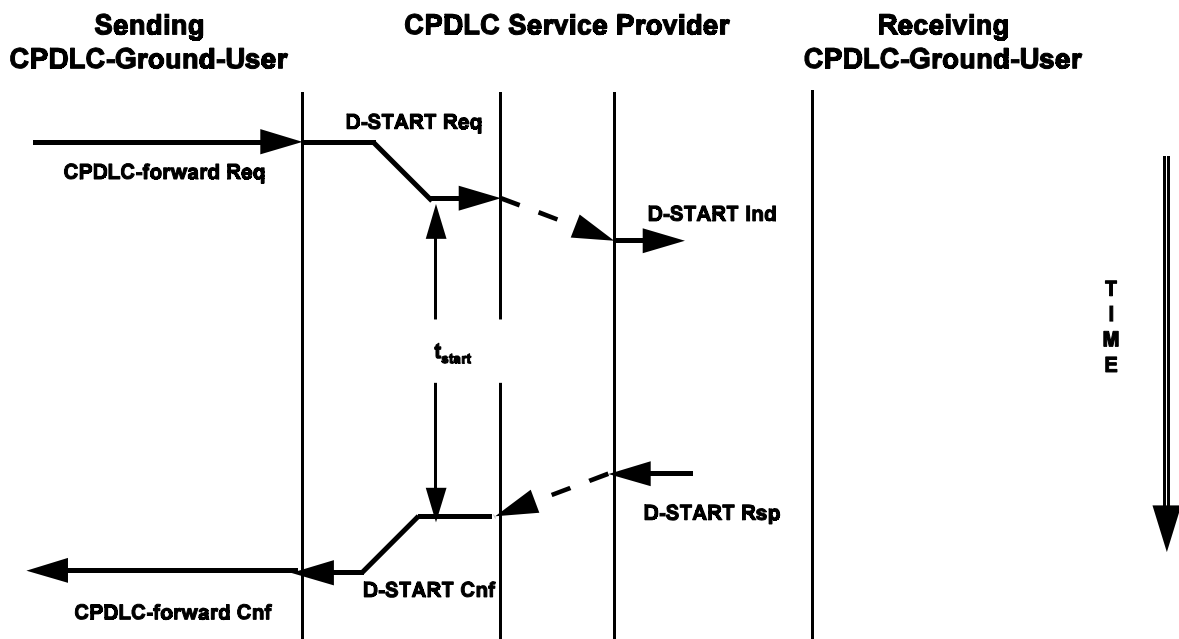


Figure 2.3.5-9. Sequence Diagram for CPDLC-forward Service/Ground Forwarding Not Supported, or Ground Forwarding Supported and ASE Version Numbers Not the Same

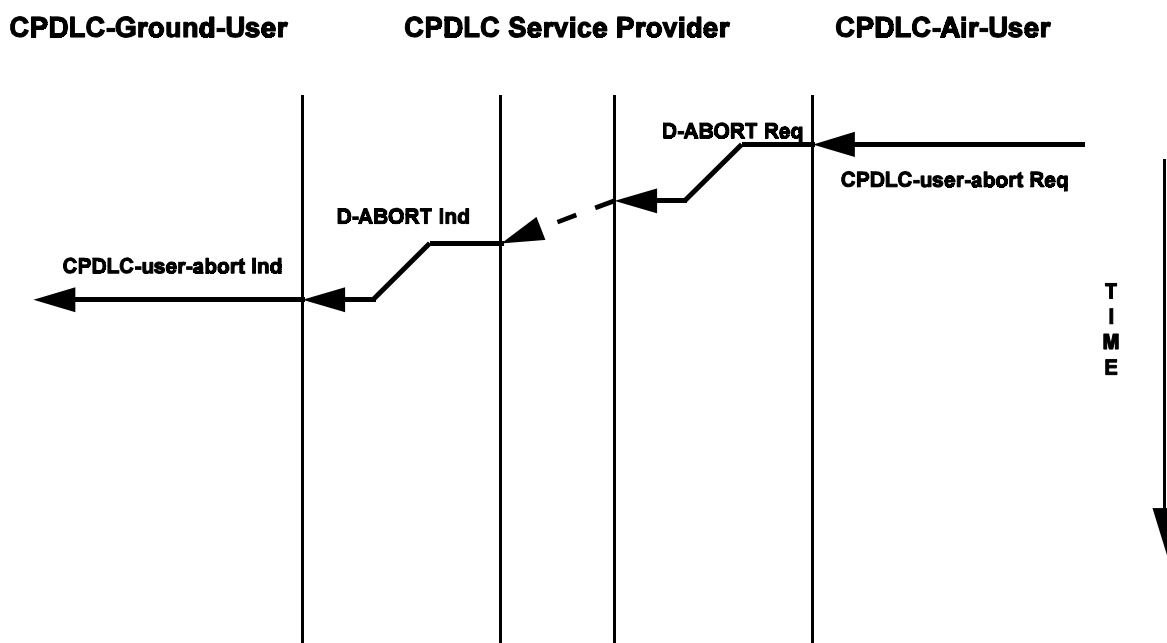


Figure 2.3.5-10. Sequence Diagram for CPDLC-user-abort Service/CPDLC-Air-User Initiated

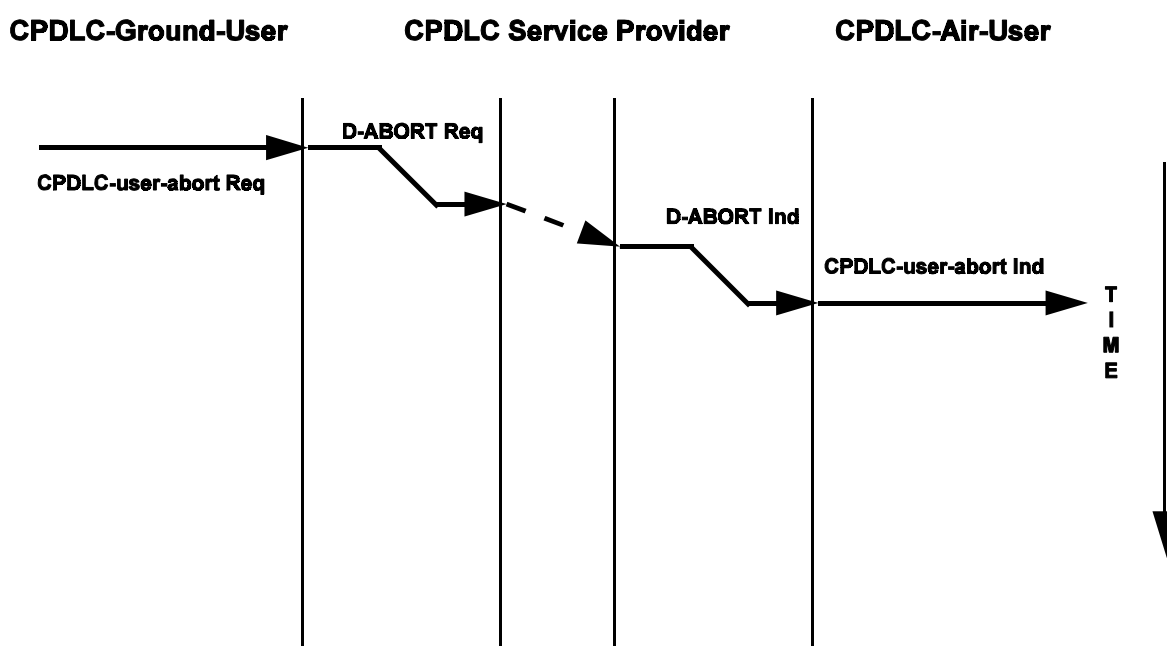


Figure 2.3.5-11. Sequence Diagram for CPDLC-user-abort Service/CPDLC-Ground-User Initiated

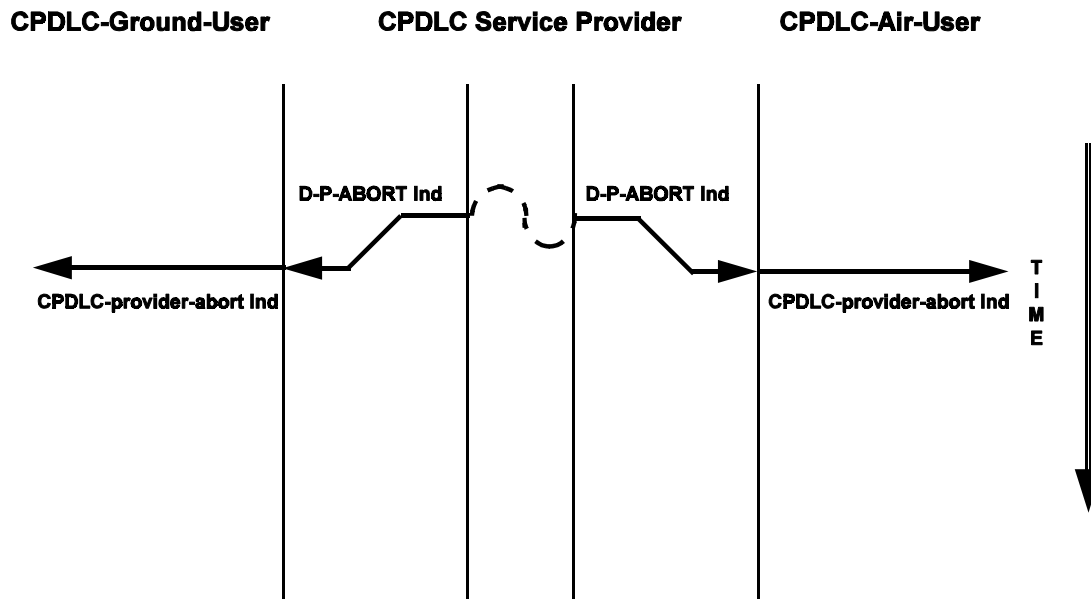


Figure 2.3.5-12. Sequence Diagram for CPDLC-provider-abort Service/Dialogue Service Abort

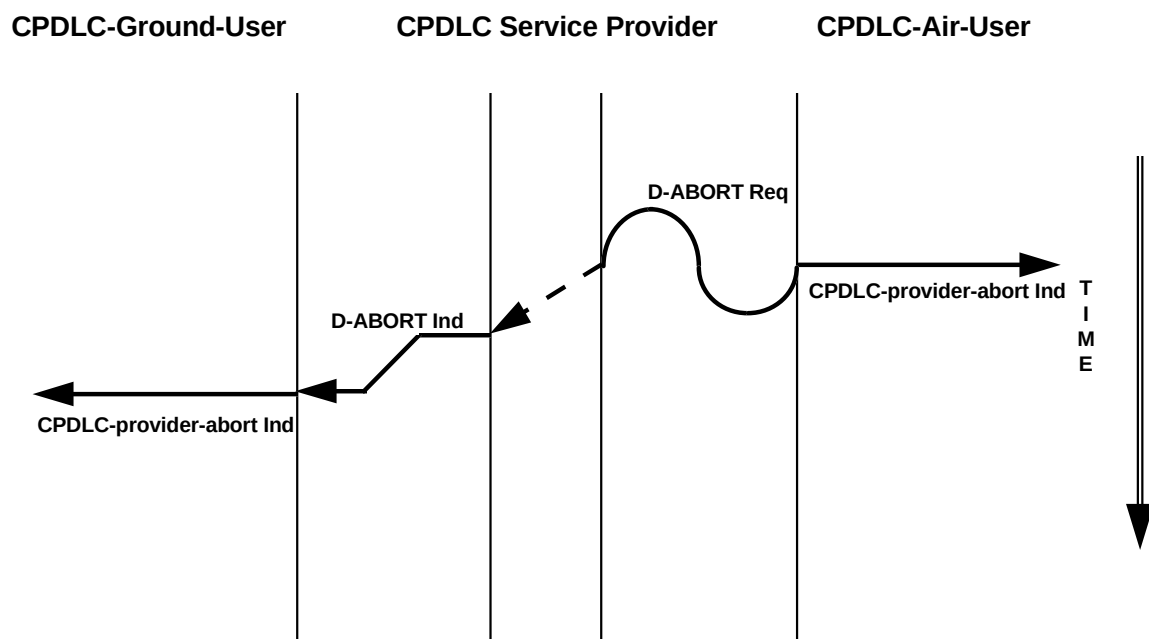


Figure 2.3.5-13. Sequence Diagram for CPDLC-provider-abort Service/CPDLC-Air-ASE Abort

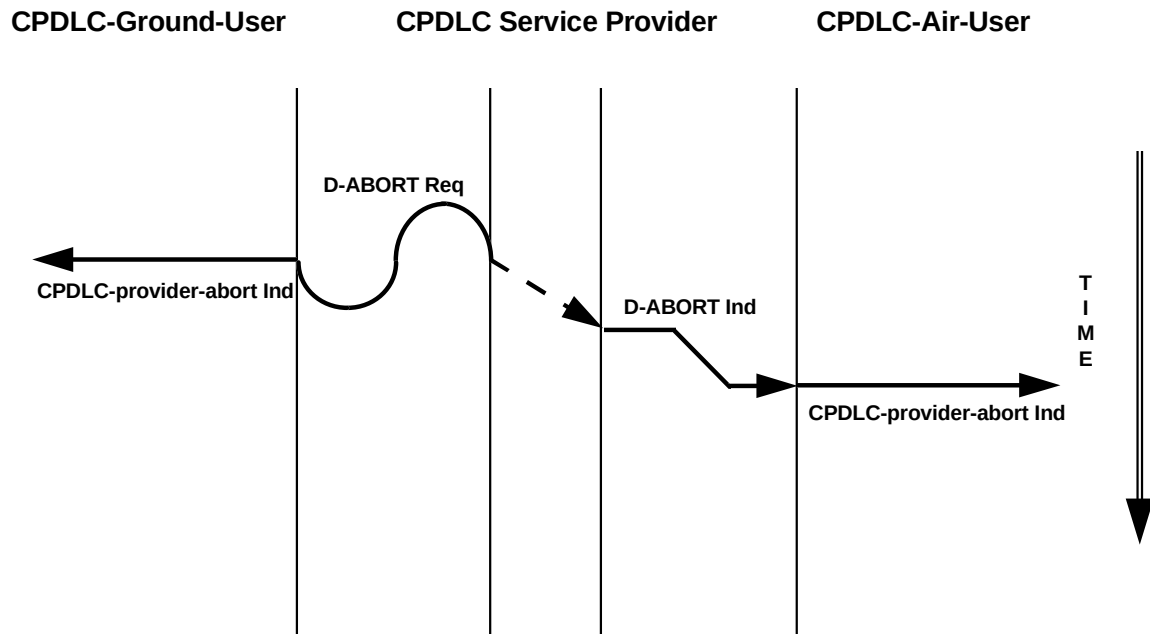


Figure 2.3.5-14. Sequence Diagram for CPDLC-provider-abort Service/
CPDLC-Ground-ASE Abort

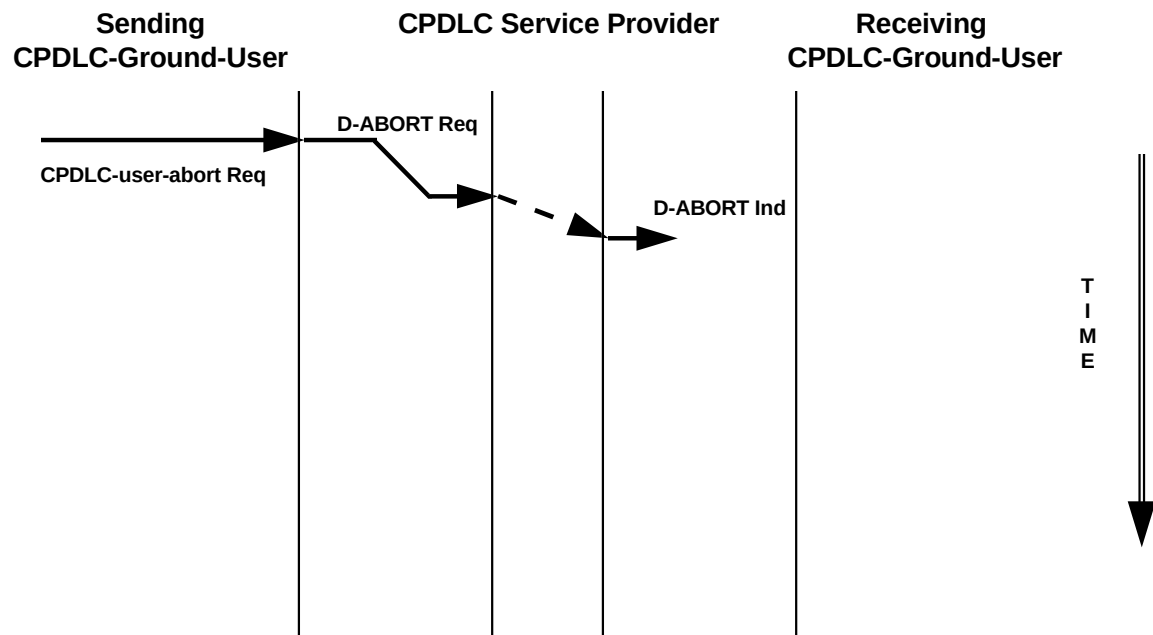


Figure 2.3.5-15. Sequence Diagram for CPDLC-user-abort Service/Sending
CPDLC-Ground-User Initiated

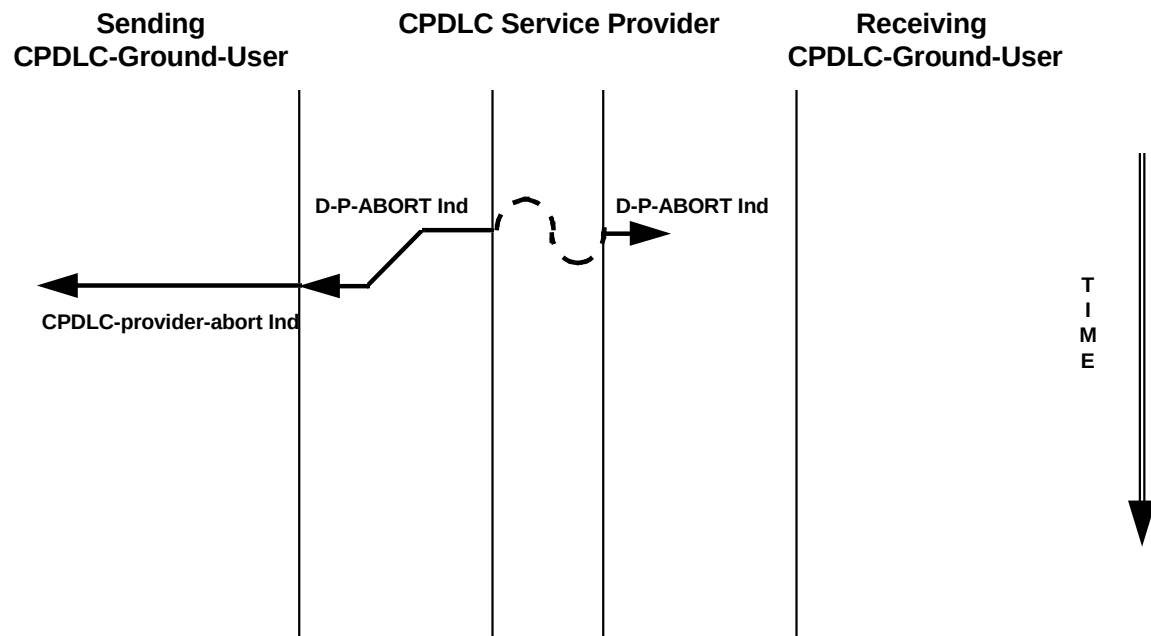


Figure 2.3.5-16. Sequence Diagram for CPDLC-provider-abort Service/Dialogue Service Abort

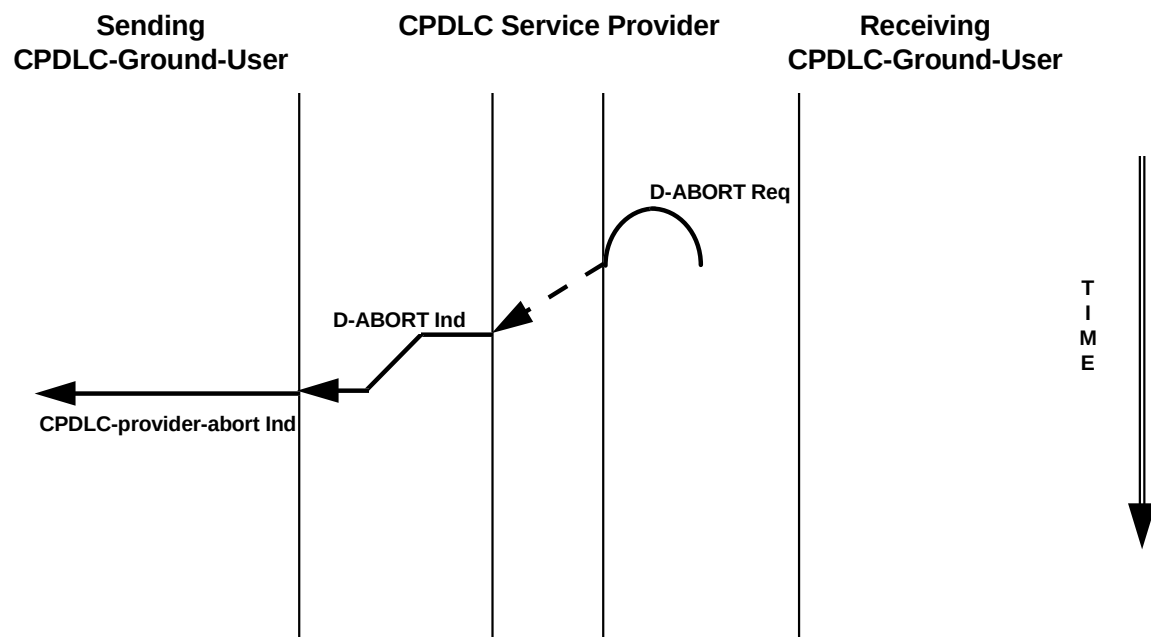


Figure 2.3.5-17. Sequence Diagram for CPDLC-provider-abort Service/Receiving CPDLC-Ground-ASE Abort

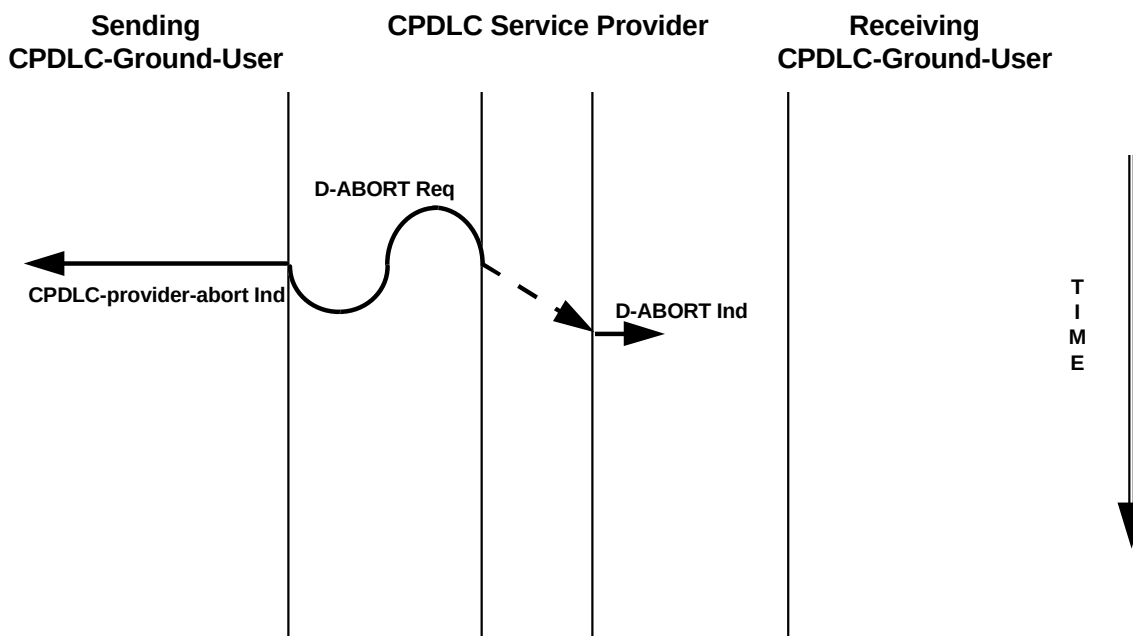


Figure 2.3.5-18. Sequence Diagram for CPDLC-provider-abort Service/Sending CPDLC-Ground-ASE Abort

2.3.5.2 CPDLC Service Provider Timers

2.3.5.2.1 A CPDLC-ASE shall be capable of detecting when a timer expires.

Note 1.— Table 2.3.5-1 lists the time constraints related to the CPDLC application. Each time constraint requires a timer to be set in the CPDLC protocol machine.

Note 2.— If the timer expires before the final event has occurred, a CPDLC-ASE takes appropriate action as defined in 2.3.5.4.1.

2.3.5.2.2 **Recommendation.** — The timer values should be as indicated in Table 2.3.5-1.

Table 2.3.5-1. CPDLC Service Provider Timers

CPDLC Service	Timer	Timer Value	Timer Start Event	Timer Stop Event
CPDLC-start	t_{start}	6 minutes	D-START request	D-START confirmation
DSC-start	t_{start}	6 minutes	D-START request	D-START confirmation
CPDLC-forward	t_{start}	6 minutes	D-START request	D-START confirmation

Note.— The receipt of CPDLC-user-abort requests, D-ABORT Indications, or D-P-ABORT Indications are also timer stop events.

2.3.5.3 CPDLC-Air-ASE Protocol Description

2.3.5.3.1 Introduction

2.3.5.3.1.1 If no actions are described for a CPDLC service primitive when a CPDLC-air-ASE is in a specific state, then the invocation of that service primitive shall be prohibited while the CPDLC-air-ASE is in that state.

2.3.5.3.1.2 Upon receipt of a PDU, if no actions are described for the arrival of that PDU when a CPDLC-air-ASE is in a specific state, then that PDU is considered not permitted, and exception handling procedures as described in 2.3.5.4.4 shall apply.

2.3.5.3.1.3 If a PDU is received that cannot be decoded, then exception handling procedures as described in 2.3.5.4.3 for invalid PDU shall apply.

2.3.5.3.1.4 If a PDU is not received when one is required, then exception handling as described 2.3.5.4.3 shall apply.

Note 1.— The states defined for the CPDLC-air-ASE are the following.

- a) *IDLE*
- b) *START-REQ,*
- c) *START-IND,*
- d) *DIALOGUE, and*
- e) *END.*

Note 2.— The CPDLC-air-user is an active user from:

- a) *the time it has invoked the CPDLC-start service request until:*
 - 1) *receipt of a CPDLC-start service confirmation with Result parameter equal to the abstract value “rejected”, or*
 - 2) *invocation of a CPDLC-end service response with the Result parameter set to the abstract value “accepted”, or*
 - 3) *invocation of a CPDLC-user-abort service request, or*
 - 4) *receipt of CPDLC-user-abort service indication, or*
 - 5) *receipt of a CPDLC-provider-abort service indication; or*

- b) *the time it has received the CPDLC-start service indication until:*
 - 1) *invocation of a CPDLC-start service response with Result parameter equal to the abstract value “rejected”, or*
 - 2) *invocation of a CPDLC-end service response with the Result parameter set to the abstract value “accepted”, or*
 - 3) *invocation of a CPDLC-user-abort service request, or*
 - 4) *receipt of CPDLC-user-abort service indication, or*
 - 5) *receipt of a CPDLC-provider-abort service indication; or*
- c) *the time it has invoked the DSC-start service request until:*
 - 1) *receipt of a DSC-start service confirmation with Result parameter equal to the abstract value “rejected”, or*
 - 2) *receipt of a DSC-end service confirmation with Result parameter equal to the abstract value “accepted”, or*
 - 3) *invocation of a CPDLC-user-abort service request, or*
 - 4) *receipt of CPDLC-user-abort service indication, or*
 - 5) *receipt of a CPDLC-provider-abort service indication.*

2.3.5.3.1.5 On initiation the CPDLC-air-ASE shall be in the *IDLE* state.

Note.— The CPDLC-air-ASE contains a Boolean called DSC. DSC has the abstract value “true” when the dialogue is a DSC dialogue, and has the abstract value “false” otherwise.

2.3.5.3.1.6 On the initiation of a CPDLC-air-ASE, DSC shall be set to the abstract value “false”.

2.3.5.3.2 D-START Indication

2.3.5.3.2.1 Upon receipt of a D-START indication, if the CPDLC-air-ASE is in the *IDLE* state and the D-START *User Data* parameter contains a GroundPDUs [UplinkMessage] APDU, and the D-START QOS *Priority* parameter has the abstract value “high priority flight safety message” and the D-START QOS *Residual Error Rate* parameter has the abstract value “low”, the D-START QOS *Routing Class* parameter has one of the abstract values specified in Table 2.3.6-1, and the D-START *Calling Peer ID* parameter is a valid four to eight character facility designation, the CPDLC-air-ASE shall:

- a) Invoke CPDLC-start service indication containing the following:
 - 1) the D-START *Calling Peer ID* parameter value as the CPDLC-start service *Calling Peer Identifier* parameter value,

- 2) the D-START *QOS Routing Class* parameter value as the CPDLC-start service *Class of Communication* parameter value,
 - 3) if the GroundPDUs [UplinkMessage] APDU contained in the D-START *User Data* parameter is an ATCUplinkMessage, set the GroundPDUs APDU-element as the CPDLC-start service *CPDLC Message* parameter value, and
- b) Enter the *START-IND* state.

2.3.5.3.3 D-START Confirmation

2.3.5.3.3.1 Upon receipt of a D-START confirmation, if the CPDLC-air-ASE is in the *START-REQ* state and the D-START *Result* parameter has the abstract value “accepted” and DSC has the abstract value “false” and D-START *User Data* parameter is not provided, the CPDLC-air-ASE shall:

- a) Stop timer t_{start} ,
- b) Invoke CPDLC-start service confirmation with the abstract value “accepted” as the CPDLC-start service *Result* parameter value,
- c) Enter the *DIALOGUE* state.

2.3.5.3.3.2 Upon receipt of a D-START confirmation, if the CPDLC-air-ASE is in the *START-REQ* state and the D-START *Result* parameter has the abstract value “rejected (permanent)” and the D-START *Reject Source* parameter has the abstract value “DS user” and DSC has the abstract value “false” and if the D-START *User Data* parameter is provided, the *User Data* parameter contains a GroundPDUs [ATCUplinkMessage] APDU, the CPDLC-air-ASE shall:

- a) Stop timer t_{start} ,
- b) Invoke CPDLC-start service confirmation containing the following:
 - 1) if the D-START *User Data* parameter is provided, the APDU contained in the D-START *User Data* parameter as the CPDLC-start service *Reject Reason* parameter value, and
 - 2) the abstract value “rejected” as the CPDLC-start service *Result* parameter value, and
- c) Enter the *IDLE* state.

2.3.5.3.3.3 Upon receipt of a D-START confirmation, if the CPDLC-air-ASE is in the *START-REQ* state and the D-START *Result* parameter has the abstract value “accepted” and DSC has the abstract value “true” and D-START *User Data* parameter is not provided, the CPDLC-air-ASE shall:

- a) Stop timer t_{start} ,

- b) Invoke DSC-start service confirmation with the abstract value “accepted” as the DSC-start service *Result* parameter value,
- c) Enter the *DIALOGUE* state.

2.3.5.3.3.4 Upon receipt of a D-START confirmation, if the CPDLC-air-ASE is in the *START-REQ* state and the D-START *Result* parameter has the abstract value “rejected (permanent)” and the D-START *Reject Source* parameter has the abstract value “DS user”, and DSC has the abstract value “true”, and if the D-START *User Data* parameter is provided, the *User Data* parameter contains a GroundPDUs [ATCUplinkMessage] APDU, the CPDLC-air-ASE shall:

- a) Stop timer t_{start} ,
- b) Invoke DSC-start service confirmation containing the following:
 - 1) if the D-START *User Data* parameter is provided, the APDU contained in the D-START *User Data* parameter as the CPDLC-start service *Reject Reason* parameter value, and
 - 2) the abstract value “rejected” as the DSC-start service *Result* parameter value,
- c) Set DSC to the abstract value “false”, and
- d) Enter the *IDLE* state.

2.3.5.3.4 D-DATA Indication

2.3.5.3.4.1 Upon receipt of a D-DATA indication, if the CPDLC-air-ASE is in the *DIALOGUE* state and the APDU contained in the D-DATA *User Data* parameter is a GroundPDUs [ATCUplinkMessage] APDU, the CPDLC-air-ASE shall:

- a) Invoke CPDLC-message service indication with the APDU contained in the D-DATA *User Data* parameter as the CPDLC-message service *CPDLC Message* parameter value, and
- b) Remain in the *DIALOGUE* state.

2.3.5.3.4.2 Upon receipt of a D-DATA indication, if the CPDLC-air-ASE is in the *END* state and DSC has the abstract value of “true” and the APDU contained in the D-DATA *User Data* parameter is a GroundPDUs [ATCUplinkMessage] APDU, the CPDLC-air-ASE shall:

- a) Invoke CPDLC-message service indication with the APDU contained in the D-DATA *User Data* parameter as the CPDLC-message service *CPDLC Message* parameter value, and
- b) Remain in the *END* state.

2.3.5.3.5 D-END Indication

2.3.5.3.5.1 Upon receipt of a D-END indication, if the CPDLC-air-ASE is in the *DIALOGUE* state, and DSC has the abstract value “false”, and if the D-END *User Data* parameter is provided, the *User Data* parameter contains a GroundPDUs [ATCUplinkMessage] APDU, the CPDLC-air-ASE shall:

- a) Invoke CPDLC-end service indication with the APDU contained in the D-END *User Data* parameter as the CPDLC-message service *CPDLC Message* parameter value, if provided, as the CPDLC-end service *CPDLC Message* parameter value, and
- b) Enter the *END* state.

2.3.5.3.6 D-END Confirmation

2.3.5.3.6.1 Upon receipt of a D-END confirmation, if the CPDLC-air-ASE is in the *END* state and the abstract the D-END *Result* parameter has the abstract value “accepted” and DSC has the abstract value “true” and if the D-END *User Data* parameter is provided, the *User Data* parameter contains a GroundPDUs [ATCUplinkMessage] APDU, the CPDLC-air-ASE shall:

- a) Invoke DSC-end service confirmation with:
 - 1) if the D-END *User Data* parameter is provided, the APDU contained in the D-END *User Data* parameter as the DSC-end service *CPDLC Message* parameter value, and
 - 2) the abstract value “accepted” as the CPDLC-end service *Result* parameter value,
- b) Set DSC to the abstract value “false”, and
- c) Enter the *IDLE* state.

2.3.5.3.6.2 Upon receipt of a D-END confirmation, if the CPDLC-air-ASE is in the *END* state and the D-END *Result* parameter has the abstract value “rejected”, and DSC has the abstract value “true”, and if the D-END *User Data* parameter is provided, the *User Data* parameter contains a GroundPDUs [ATCUplinkMessage] APDU, the CPDLC-air-ASE shall:

- a) Invoke DSC-end service confirmation with:
 - 1) if the D-END *User Data* parameter is provided, the APDU contained in the D-END *User Data* parameter as the DSC-end service *CPDLC Message* parameter value, and
 - 2) the abstract value “rejected” as the CPDLC-end service *Result* parameter value:
- b) Enter the *DIALOGUE* state.

2.3.5.3.7 CPDLC-start Service Request

2.3.5.3.7.1 Upon receipt of a CPDLC-start service request, if the CPDLC-air-ASE is in the *IDLE* state, the CPDLC-air-ASE shall:

- a) Create an AircraftPDUs APDU with a StartDownMessage APDU element containing:
 - 1) the abstract value “cpdlc” as the mode,
 - 2) if provided, the *CPDLC Message* parameter as the DownlinkMessage, or
 - 3) else, NULL as the DownlinkMessage,
- b) Invoke D-START request with the following:
 - 1) the CPDLC-start service *Called Peer Identifier* parameter value as the D-START *Called Peer ID* parameter value,
 - 2) the CPDLC-start service *Calling Peer Identifier* parameter value as the D-START *Calling Peer ID* parameter value,
 - 3) the D-START *Quality of Service* parameters set as follows:
 - i) if provided, the CPDLC-start service *Class of Communication* parameter value as the D-START *QOS Routing Class* parameter value, or
 - ii) The abstract value of “high priority flight safety messages”, as the D-START *QOS Priority* parameter value, and
 - iii) The abstract value of “low” as the D-START *QOS Residual Error Rate* parameter value, and
 - 4) the APDU as the D-START *User Data* parameter value;
- c) Start timer t_{start} and
- d) Enter the *START-REQ* state.

2.3.5.3.8 CPDLC-start Service Response

2.3.5.3.8.1 Upon receipt of a CPDLC-start service response, if the CPDLC-air-ASE is in the *START-IND* state and the CPDLC-start service *Result* parameter has the abstract value “accepted” and the CPDLC-start service *Reject Reason* parameter is not provided, and DSC has the abstract value “false”, the CPDLC-air-ASE shall:

- a) Invoke D-START response with the abstract value “accepted” as the D-START *Result* parameter value, and

- b) Enter the *DIALOGUE* state.

2.3.5.3.8.2 Upon receipt of a CPDLC-start service response, if the CPDLC-air-ASE is in the *START-IND* state, and the CPDLC-start service *Result* parameter has the abstract value “rejected” and DSC has the abstract value “false”, the CPDLC-air-ASE shall:

- a) If the CPDLC-start service *Reject Reason* parameter is provided, create an AircraftPDUs APDU with an ATCDownlinkMessage APDU element based on the *Reject Reason* parameter,
- b) Invoke D-START response with the following:
 - 1) if created, the APDU as the D-START *User Data* parameter, and
 - 2) the abstract value “rejected (permanent)” as the D-START *Result* parameter value, and
- c) Enter the *IDLE* state.

2.3.5.3.9 DSC-start Service Request

2.3.5.3.9.1 Upon receipt of a DSC-start service request, if the CPDLC-air-ASE is in the *IDLE* state, the CPDLC-air-ASE shall:

- a) Create an AircraftPDUs APDU with a StartDownMessage APDU element containing:
 - 1) the abstract value “dsc” as the mode,
 - 2) if provided, the *CPDLC Message* parameter as the DownlinkMessage, or
 - 3) else, NULL as the DownlinkMessage,
- b) Invoke D-START request with the following:
 - 1) the DSC-start service *Facility Designation* parameter value as the D-START *Called Peer ID* parameter value,
 - 2) the DSC-start service *Aircraft Address* parameter value as the D-START *Calling Peer ID* parameter value,
 - 3) Set the D-START *Quality of Service* parameters as follows:
 - i) if provided, the DSC-START service *Class of Communication* parameter value as the D-START *QOS Routing Class* parameter value,
 - ii) The abstract value of “high priority flight safety messages”, as the D-START *QOS Priority* parameter value, and

- iii) The abstract value of “low” as the D-START *QOS Residual Error Rate* parameter value, and
- 4) the APDU as the D-START *User Data* parameter value;
- c) Set DSC to the abstract value “true”,
- d) Start timer t_{start} and
- e) Enter the *START-REQ* state.

2.3.5.3.10 CPDLC-message Service Request

2.3.5.3.10.1 Upon receipt of a CPDLC-message service request, if the CPDLC-air-ASE is in the *DIALOGUE* state, the CPDLC-air-ASE shall:

- a) Create an AircraftPDUs APDU with an ATCDownlinkMessage APDU-element based on the CPDLC-message service *CPDLC Message* parameter,
- b) Invoke D-DATA request with the APDU as the D-DATA *User Data* parameter value, and
- c) Remain in the *DIALOGUE* state.

2.3.5.3.10.2 Upon receipt of a CPDLC-message service request, if the CPDLC-air-ASE is in the *END* state and DSC has the abstract value “false”, the CPDLC-air-ASE shall:

- a) Create an AircraftPDUs APDU with an ATCDownlinkMessage APDU-element based on the CPDLC-message service *CPDLC Message* parameter,
- b) Invoke D-DATA request with the APDU as the D-DATA *User Data* parameter value, and
- c) Remain in the *END* state.

2.3.5.3.11 CPDLC-end Service Response

2.3.5.3.11.1 Upon receipt of a CPDLC-end service response, if the CPDLC-air-ASE is in the *END* state, and the CPDLC-end service *Result* parameter has the abstract value “accepted” and DSC has the abstract value “false”, the CPDLC-air-ASE shall:

- a) Create an AircraftPDUs APDU with an ATCDownlinkMessage APDU-element based on the CPDLC-end service *CPDLC Message* parameter, if provided,
- b) Invoke D-END response with the following:
 - 1) if created, the APDU as the D-END *User Data* parameter value, and
 - 2) the abstract value “accepted”, as the D-END *Result* parameter value, and

- c) Enter the *IDLE* state.

2.3.5.3.11.2 Upon receipt of a CPDLC-end service response, if the CPDLC-air-ASE is in the *END* state, and the CPDLC-end service *Result* parameter has the abstract value “rejected” and DSC has the abstract value “false”, the CPDLC-air-ASE shall:

- a) Create an AircraftPDUs APDU with an ATCDownlinkMessage APDU-element based on the CPDLC-end service *CPDLC Message* parameter, if provided,
- b) Invoke D-END response with the following:
 - 1) if created, the APDU as the D-END *User Data* parameter value, and
 - 2) the abstract value “rejected”, as the D-END *Result* parameter value, and
- c) Enter the *DIALOGUE* state.

2.3.5.3.12 DSC-end Service Request

2.3.5.3.12.1 Upon receipt of a DSC-end service request, if the CPDLC-air-ASE is in the *DIALOGUE* state and DSC has the abstract value “true”, the CPDLC-air-ASE shall:

- a) Create an AircraftPDUs APDU with an ATCDownlinkMessage APDU-element based on the DSC-end service *CPDLC Message* parameter, if provided,
- b) Invoke D-END request with the APDU as the D-END *User Data* parameter value, if provided, and
- c) Enter the *END* state.

2.3.5.3.13 CPDLC-user-abort Service Request

2.3.5.3.13.1 Upon receipt of a CPDLC-user-abort service request, if the CPDLC-air-ASE is not in the *IDLE* state, the CPDLC-air-ASE shall:

- a) Stop any timer,
- b) If the CPDLC-user-abort service *Reason* parameter is provided, create an AircraftPDUs APDU with a CPDLCUserAbortReason APDU-element based on the CPDLC-user-abort service *Reason* parameter,
- c) Else create an AircraftPDUs APDU with a CPDLCUserAbortReason [undefined] APDU-element,
- d) Invoke D-ABORT request with the following:
 - 1) the D-ABORT *Originator* parameter set to the abstract value “user”, and
 - 2) the APDU as the D-ABORT *User Data* parameter value, and

- e) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- f) Enter the *IDLE* state.

2.3.5.3.14 D-ABORT Indication

2.3.5.3.14.1 Upon receipt of a D-ABORT indication, if the CPDLC-air-ASE is not in the *IDLE* state, and the D-ABORT *Originator* parameter is “user” and the D-ABORT *User Data* parameter contains a GroundPDUs [CPDLCUserAbortReason] APDU, the CPDLC-air-ASE shall:

- a) Stop any timer,
- b) If the CPDLC-air-user is an active user, invoke CPDLC-user-abort service indication with the APDU contained in the D-ABORT *User Data* parameter as the CPDLC-user-abort service *Reason* parameter value,
- c) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- d) Enter the *IDLE* state.

2.3.5.3.14.2 Upon receipt of a D-ABORT indication, if the CPDLC-air-ASE is not in the *IDLE* state, and if the D-ABORT *Originator* parameter is “provider” and if the D-ABORT *User Data* parameter is provided, the D-ABORT *User Data* parameter contains a GroundPDUs [CPDLCProviderAbortReason] APDU, the CPDLC-air-ASE shall:

- a) Stop any timer,
- b) If the CPDLC-air-user is an active user, invoke CPDLC-provider-abort service indication with the D-ABORT *User Data* parameter as the CPDLC-provider-abort service *Reason* parameter value, if provided,
- c) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- d) Enter the *IDLE* state.

2.3.5.3.15 D-P-ABORT Indication

2.3.5.3.15.1 Upon receipt of a D-P-ABORT indication, if the CPDLC-air-ASE is not in the *IDLE* state, the CPDLC-air-ASE shall:

- a) Stop any timer,
- b) If the CPDLC-air-user is an active user, invoke CPDLC-provider-abort service indication with the CPDLC-provider-abort service *Reason* parameter set to the abstract value “communication-service-failure”,
- c) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- d) Enter the *IDLE* state.

2.3.5.4 CPDLC-Air-ASE Exception Handling

2.3.5.4.1 A Timer Expires

2.3.5.4.1.1 If a CPDLC-air-ASE detects that a timer has expired, that CPDLC-air-ASE shall:

- a) Interrupt any current activity,
- b) Create an AircraftPDUs APDU with a CPDLCProviderAbortReason [timer-expired] APDU message element,
- c) Invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter value, and
- d) If the CPDLC-air-user is an active user, invoke CPDLC-provider-abort service indication with the abstract value “timer-expired” as the CPDLC-provider abort service *Reason* parameter value,
- e) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- f) Enter the *IDLE* state.

2.3.5.4.2 Unrecoverable System Error

2.3.5.4.2.1 **Recommendation.** — *If a CPDLC-air-ASE has an unrecoverable system error, the CPDLC-air-ASE should:*

- a) *Stop any timer,*
- b) *Create an AircraftPDUs APDU with a CPDLCProviderAbortReason [undefined-error] APDU message element,*
- c) *Invoke D-ABORT request with:*
 - 1) *the abstract value “provider” as the D-ABORT Originator parameter value, and*
 - 2) *the APDU as the D-ABORT User Data parameter value, and*
- d) *If the CPDLC-air-user is an active user, invoke CPDLC-provider-abort service indication with the abstract value “undefined-error” as the CPDLC-provider abort service Reason parameter value,*
- e) *If DSC has the abstract value “true”, set DSC to the abstract value “false”, and*

- f) Enter the *IDLE* state.

2.3.5.4.3 Invalid PDU

2.3.5.4.3.1 If the *User Data* parameter of a D-END confirmation with *Result* parameter set to the abstract value “rejected”, or if the *User Data* parameter of a D-START indication, a D-DATA indication, or a D-END indication, does not contain a valid PDU, the CPDLC-air-ASE shall:

- a) Stop any timer,
- b) Create an AircraftPDUs APDU with a CPDLCProviderAbortReason [invalid-PDU] APDU message element,
- c) Invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter value, and
- d) If the CPDLC-air-user is an active user, invoke CPDLC-provider-abort service indication with the abstract value “invalid-PDU” as the CPDLC-provider abort service *Reason* parameter value,
- e) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- f) Enter the *IDLE* state.

2.3.5.4.3.2 If the *User Data* parameter of a D-START confirmation with *Result* set to the abstract value “rejected (permanent)”, or a D-END confirmation with *Result* set to the abstract value “accepted”, is not a valid PDU then the CPDLC-air-ASE shall:

- a) Stop any timer,
- b) If the CPDLC-air-user is an active user, invoke CPDLC-provider-abort service indication with the CPDLC-provider-abort service *Reason* parameter set to the abstract value “invalid-PDU”,
- c) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- d) Enter the *IDLE* state.

2.3.5.4.4 Protocol Error

2.3.5.4.4.1 If the *User Data* parameter of a D-START indication, D-DATA indication, or D-END indication is a valid PDU, but is not a PDU for which action is described within a given state in 2.3.5.3, the CPDLC-air-ASE shall:

- a) Stop any timer,

- b) Create an AircraftPDUs APDU with a CPDLCProviderAbortReason [protocol-error] APDU message element,
- c) Invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter value, and
- d) If the CPDLC-air-user is an active user, invoke CPDLC-provider-abort service indication with the abstract value “protocol-error” as the CPDLC-provider abort service *Reason* parameter value ,
- e) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- f) Enter the *IDLE* state.

2.3.5.4.4.2 If a D-START confirmation with the *Result* parameter set to the abstract value “accepted” contains a *User Data* parameter the CPDLC-air-ASE shall:

- a) Stop any timer,
- b) Create an AircraftPDUs APDU with a CPDLCProviderAbortReason [protocol-error] APDU message element,
- c) Invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter value, and
- d) If the CPDLC-air-user is an active user, invoke CPDLC-provider-abort service indication with the abstract value “protocol-error” as the CPDLC-provider-abort service *Reason* parameter value,
- e) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- f) Enter the *IDLE* state.

2.3.5.4.4.3 If the *User Data* parameter of a D-END confirmation is a valid PDU, but is not a permitted PDU as defined in 2.3.5.3, the CPDLC-air-ASE shall:

- a) If the D-END *Result* parameter is set to the abstract value “rejected”, then
 - 1) Stop any timer,

- 2) Create an AircraftPDUs APDU with a CPDLCProviderAbortReason [protocol-error] APDU message element,
- 3) Invoke D-ABORT request with:
 - i) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - ii) the APDU as the D-ABORT *User Data* parameter value, and
- b) If the CPDLC-air-user is an active user, invoke CPDLC-provider-abort service indication with the abstract value “protocol-error” as the CPDLC-provider-abort service *Reason* parameter value,
- c) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- d) Enter the *IDLE* state.

2.3.5.4.4.4 Upon receipt of a Dialogue service primitive for which there are no instructions in 2.3.5.3 (i.e. the primitive was not expected or was expected under other conditions or with other parameter values), the CPDLC-air-ASE shall:

- a) Stop any timer,
- b) Create an AircraftPDUs APDU with a CPDLCProviderAbortReason [protocol-error] APDU message element,
- c) If a dialogue exists, invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter value, and
- d) If the CPDLC-air-user is an active user, invoke CPDLC-provider-abort service indication with the abstract value “protocol-error” as the CPDLC-provider abort service *Reason* parameter value ,
- e) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- f) Enter the *IDLE* state.

2.3.5.4.5 D-START Confirmation *Result* or *Reject Source* Not as Expected

2.3.5.4.5.1 If a D-START confirmation *Result* parameter has the abstract value of “rejected (transient)” or if the *Reject Source* parameter has the abstract value of “DS provider”, the CPDLC-air-ASE shall:

- a) Stop any timer,

- b) If the CPDLC-air-user is an active user, invoke CPDLC-provider-abort service indication with the CPDLC-provider-abort service *Reason* parameter set to the abstract value “communication-service-error”,
- c) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- d) Enter the *IDLE* state.

2.3.5.4.6 D-START Indication *Quality of Service* Not as Expected

2.3.5.4.6.1 If a D-START indication *QOS Priority* parameter does not have the abstract value of “high priority flight safety messages” or if the *QOS Residual Error Rate* parameter does not have the abstract value of “low”, or if the *QOS Routing Class* parameter does not have one of the abstract values specified in Table 2.3.6-1, the CPDLC-air-ASE shall:

- a) Stop any timer,
- b) Create an AircraftPDUs APDU with a CPDLCProviderAbortReason [invalid-QOS-parameter] APDU message element,
- c) Invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as D-ABORT *User Data* parameter value,
- d) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- e) Enter the *IDLE* state.

2.3.5.4.7 Expected PDU Missing

2.3.5.4.7.1 If the *User Data* parameter of a D-START indication or D-DATA indication does not contain a PDU, the CPDLC-air-ASE shall:

- a) Stop any timer,
- b) Create an AircraftPDUs APDU with a CPDLCProviderAbortReason [expected-PDU-missing] APDU message element,
- c) Invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter values,

- d) If the CPDLC-air-user is an active user, invoke CPDLC-provider-abort service indication with the abstract value “not-permitted-PDU” as the CPDLC-provider-abort service *Reason* parameter value,
- e) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- f) Enter the *IDLE* state.

2.3.5.5 CPDLC-Ground-ASE Protocol Description

2.3.5.5.1 Introduction

2.3.5.5.1.1 If no actions are described for a CPDLC service primitive when a CPDLC-ground-ASE is in specific state, then the invocation of that service primitive shall be prohibited while the CPDLC-ground-ASE is in that state.

2.3.5.5.1.2 Upon receipt of a PDU, if no actions are described for the arrival of that PDU when a CPDLC-ground-ASE is in a specific state, then that PDU is considered not permitted, and exception handling procedures as described in 2.3.5.6.4 shall apply.

2.3.5.5.1.3 If a PDU is received that cannot be decoded, then exception handling procedures as described in 2.3.5.6.3 for invalid PDU shall apply.

2.3.5.5.1.4 If a PDU is not received when one is required, then exception handling as described in 2.3.5.6.3 shall apply.

Note 1.— The states defined for the CPDLC-ground-ASE are the following.

- a) *IDLE*
- b) *START-REQ,*
- c) *START-IND,*
- d) *DIALOGUE,*
- e) *END, and*
- f) *FORWARD.*

Note 2.— The CPDLC-ground-user is an active user from:

- a) *the time it has invoked the CPDLC-start service request until:*
 - 1) *receipt of a CPDLC-start service confirmation with Result parameter equal to the abstract value “rejected”, or*
 - 2) *receipt of a CPDLC-end service confirmation with the Result parameter equal to the abstract value “accepted”, or*

- 3) invocation of a CPDLC-user-abort service request, or
- 4) receipt of a CPDLC-user-abort service indication, or
- 5) receipt of a CPDLC-provider-abort service indication; or
- b) the time it has received the CPDLC-start service indication until:
 - 1) invocation of a CPDLC-start service response with Result parameter set to the abstract value “rejected”, or
 - 2) receipt of a CPDLC-end service confirmation with the Result parameter equal to the abstract value “accepted”, or
 - 3) invocation of a CPDLC-user-abort service request, or
 - 4) receipt of CPDLC-user-abort service indication, or
 - 5) receipt of a CPDLC-provider-abort service indication; or
- c) the time it has received the DSC-start service indication until:
 - 1) invocation of a DSC-start service response with Result parameter equal to the abstract value “rejected”, or
 - 2) invocation of a CPDLC-user-abort service request, or
 - 3) receipt of CPDLC-user-abort service indication, or
 - 4) receipt of a CPDLC-provider-abort service indication; or
- d) the time it has invoked the CPDLC-forward service request until:
 - 1) receipt of a CPDLC-forward service confirmation,
 - 2) invocation of a CPDLC-user-abort service request, or
 - 3) receipt of CPDLC-user-abort service indication, or
 - 4) receipt of a CPDLC-provider-abort service indication.

2.3.5.5.1.5 On initiation the CPDLC-ground-ASE shall be in the *IDLE* state.

Note.— The CPDLC-ground-ASE contains a Boolean called *DSC*. *DSC* has the abstract value “true” when the dialogue is a *DSC* dialogue, and has the abstract value “false” otherwise.

2.3.5.5.1.6 On the initiation of a CPDLC-ground-ASE, *DSC* shall be set to the abstract value “false”.

2.3.5.5.2 D-START Indication

2.3.5.5.2.1 Upon receipt of a D-START indication, if the CPDLC-ground-ASE is in the *IDLE* state, and the abstract value of the D-START *Calling Peer ID* parameter is a 24-bit aircraft address, and the D-START *User Data* parameter contains an AircraftPDUs [StartDownMessage] APDU with the APDU-element mode “cpdlc”, and the D-START *QOS Priority* parameter has the abstract value “high priority flight safety messages” and the D-START *QOS Residual Error Rate* parameter has the abstract value “low” and the D-START *QOS Routing Class* parameter has one of the abstract values specified in Table 2.3.6-1, the CPDLC-ground-ASE shall:

- a) Invoke CPDLC-start service indication containing the following:
 - 1) the D-START *Calling Peer ID* parameter value as the CPDLC-start service *Calling Peer Identifier* parameter value,
 - 2) the D-START *QOS Routing Class* parameter value as the CPDLC-start service *Class of Communication* parameter value,
 - 3) if the AircraftPDUs APDU-element contained in the D-START *User Data* parameter is an ATCDownlinkMessage, set the AircraftPDUs APDU-element as the CPDLC-start service *CPDLC Message* parameter value, and
- b) Enter the *START-IND* state.

2.3.5.5.2.2 Upon receipt of a D-START indication, if the CPDLC-ground-ASE is in the *IDLE* state, and the abstract value of the D-START *Calling Peer ID* parameter is a 24-bit aircraft address, and the D-START *User Data* parameter contains an AircraftPDUs [StartDownMessage]APDU with the APDU-element mode “dsc”, and the D-START *QOS Priority* parameter has the abstract value “high priority flight safety messages” and the D-START *QOS Residual Error Rate* parameter has the abstract value “low”, and the D-START *QOS Routing Class* parameter has one of the abstract values specified in Table 2.3.6-1 the CPDLC-ground-ASE shall:

- a) Invoke DSC-start service indication containing the following:
 - 1) the D-START *Calling Peer ID* parameter value as the DSC-start service *Aircraft Address* parameter value,
 - 2) the D-START *QOS Routing Class* parameter value as the CPDLC-start service *Class of Communication* parameter value,
 - 3) if the APDU AircraftPDUs APDU contained in the D-START is an ATCDownlinkMessage, set the AircraftPDUs APDU as the DSC-start service *CPDLC Message* parameter value,
- b) Set DSC to “true”, and
- c) Enter the *START-IND* state.

2.3.5.5.2.3 Upon receipt of a D-START indication, if the CPDLC-ground-ASE is in the *IDLE* state, and the abstract value of the D-START *Calling Peer ID* parameter is a Facility Designation, and the D-START *User Data* parameter contains a GroundPDUs [ATCForwardMessage] APDU and the CPDLC-ground-ASE supports the CPDLC-forward service, and the D-START *QOS Priority* parameter has the abstract value “high priority flight safety messages” and the D-START *QOS Residual Error Rate* parameter has the abstract value “low”, the CPDLC-ground-ASE shall:

- a) If the D-START *DS User Version Number* parameter value is equal to the CPDLC-ground-ASE version number:
 - 1) Invoke CPDLC-forward service indication containing the following:
 - i) the D-START *Calling Peer ID* parameter value as the CPDLC-forward service *Calling Facility Designation* parameter value,
 - ii) set the D-START GroundPDUs APDU-element as the CPDLC-forward service *CPDLC Message* parameter value, and
 - 2) Create a GroundPDUs APDU with an ATCForwardResponse [success] APDU element,
 - 3) Invoke D-START response with the following:
 - i) the APDU as the D-START *User Data* parameter value, and
 - ii) the abstract value “rejected (permanent)” as the D-START *Result* parameter value, and
 - 4) Remain in the *IDLE* state.
- b) If the D-START *DS User Version Number* parameter value is not equal to the CPDLC-ground-ASE version number:
 - 1) Create a GroundPDUs APDU with an ATCForwardResponse [version-not-equal] APDU element,
 - 2) Invoke D-START response with the following:
 - i) the CPDLC-ground-ASE version number as the D-START *DS User Version Number* parameter value,
 - ii) the APDU as the D-START *User Data* parameter value, and
 - iii) the abstract value “rejected (permanent)” as the D-START *Result* parameter value, and
 - 3) Remain in the *IDLE* state.

2.3.5.5.2.4 Upon receipt of a D-START indication, if the CPDLC-ground-ASE is in the *IDLE* state, and the abstract value of the D-START *Calling Peer ID* parameter is a Facility Designation, and the D-START *User Data* parameter contains a GroundPDUs APDU and the APDU element is an ATCForwardMessage and the CPDLC-ground-ASE does not support the CPDLC-forward service, the CPDLC-ground-ASE shall:

- a) Create a GroundPDUs APDU with an ATCForwardResponse [service-not-supported] APDU element,
- b) Invoke D-START response with the following:
 - 1) the APDU as the D-START *User Data* parameter value, and
 - 2) the abstract value “rejected (permanent)” as the D-START *Result* parameter value, and
- c) Remain in the *IDLE* state.

2.3.5.5.3 D-START Confirmation

2.3.5.5.3.1 Upon receipt of a D-START confirmation, if the CPDLC-ground-ASE is in the *START-REQ* state and if the D-START *Result* parameter has the abstract value “accepted”, and DSC has the abstract value of “false” and D-START *User Data* parameter is not provided, the CPDLC-ground-ASE shall:

- a) Stop timer t_{start} ,
- b) Invoke CPDLC-start service confirmation containing the abstract value “accepted” as the CPDLC-start service *Result* parameter value, and
- c) Enter the *DIALOGUE* state.

2.3.5.5.3.2 Upon receipt of a D-START confirmation, if the CPDLC-ground-ASE is in the *START-REQ* state and the D-START *Result* parameter has the abstract value “rejected (permanent)” and the D-START *Reject Source* parameter has the abstract value “DS user” and DSC has the abstract value “false” and if the D-START *User Data* parameter is provided, the *User Data* parameter contains a AircraftPDUs [ATCDownlinkMessage] APDU, the CPDLC-ground-ASE shall:

- a) Stop timer t_{start} ,
- b) Invoke CPDLC-start service confirmation containing the following:
 - 1) if the D-START *User Data* parameter is provided, the APDU contained in the D-START *User Data* parameter as the CPDLC-start service *Reject Reason* parameter value, and
 - 2) the abstract value “rejected” as the CPDLC-start service *Result* parameter value, and
- c) Enter the *IDLE* state.

2.3.5.5.3.3 Upon receipt of a D-START confirmation, if the CPDLC-ground-ASE is in the *FORWARD* state and if the D-START *Result* parameter has the abstract value “rejected (permanent)” and the *Reject Source* parameter has the abstract value “DS user” and the D-START *User Data* parameter contains a GroundPDUs [ATCForwardResponse] APDU, the CPDLC-ground-ASE shall:

- a) If the D-START *DS User Version Number* parameter value is equal to the CPDLC-ground-ASE version number:
 - 1) Stop timer t_{start} ,
 - 2) Invoke CPDLC-forward service confirmation with the D-START GroundPDUs APDU element as the CPDLC-forward service *Result* parameter value, and
 - 3) Enter the *IDLE* state.
- b) If the D-START *DS User Version Number* parameter value is not equal to the CPDLC-ground-ASE version number:
 - 1) Stop timer t_{start} ,
 - 2) Invoke CPDLC-forward service confirmation with the following:
 - i) the D-START GroundPDUs APDU-element as the CPDLC-forward service *Result* parameter value, and
 - ii) the D-START *DS User Version Number* parameter value as the CPDLC-forward service *ASE Version Number* parameter value, and
 - 3) Enter the *IDLE* state.

2.3.5.5.4 D-DATA Indication

2.3.5.5.4.1 Upon receipt of a D-DATA indication, if the CPDLC-ground-ASE is in the *DIALOGUE* state and the APDU contained in the D-DATA *User Data* parameter is a AircraftPDUs [ATCDownlinkMessage] APDU, the CPDLC-ground-ASE shall:

- a) Invoke CPDLC-message service indication with the APDU contained in the D-DATA *User Data* parameter as the CPDLC-message service *CPDLC Message* parameter value, and
- b) Remain in the *DIALOGUE* state.

2.3.5.5.4.2 Upon receipt of a D-DATA indication, if the CPDLC-ground-ASE is in the *END* state and DSC has the abstract value “false” and the APDU contained in the D-DATA *User Data* parameter is an AircraftPDUs [ATCDownlinkMessage] APDU, the CPDLC-ground-ASE shall:

- a) Invoke CPDLC-message service indication with the APDU contained in the D-DATA *User Data* parameter as the CPDLC-message service *CPDLC Message* parameter value, and
- b) Remain in the *END* state.

2.3.5.5.5 D-END Indication

2.3.5.5.5.1 Upon receipt of a D-END indication, if the CPDLC-ground-ASE is in the *DIALOGUE* state, and DSC has the abstract value “true”, and if the D-END *User Data* parameter is provided, the *User Data* parameter contains an AircraftPDUs [ATCDownlinkMessage] APDU, the CPDLC-ground-ASE shall:

- a) Invoke DSC-end service indication with the APDU contained in the D-END *User Data* parameter as the DSC-end service *CPDLC Message* parameter value, if provided, and
- b) Enter the *END* state.

2.3.5.5.6 D-END Confirmation

2.3.5.5.6.1 Upon receipt of a D-END confirmation, if the CPDLC-ground-ASE is in the *END* state and the D-END *Result* parameter has the abstract value “accepted” and DSC has the abstract value “false” and if the D-END *User Data* parameter is provided, the *User Data* parameter contains an AircraftPDUs [ATCDownlinkMessage] APDU, the CPDLC-ground-ASE shall:

- a) Invoke CPDLC-end service confirmation with:
 - 1) The APDU contained in the D-END *User Data* parameter as the CPDLC-end service *CPDLC Message* parameter value, if provided, and
 - 2) The abstract value “accepted” as the CPDLC-end service *Result* parameter value, and
- b) Enter the *IDLE* state.

2.3.5.5.6.2 Upon receipt of a D-END confirmation, if the CPDLC-ground-ASE is in the *END* state and the D-END *Result* has the abstract value “rejected” and DSC has the abstract value “false”, and if the D-END *User Data* parameter is provided, the *User Data* parameter contains an AircraftPDUs [ATCDownlinkMessage] APDU, the CPDLC-ground-ASE shall:

- a) Invoke CPDLC-end service confirmation with:
 - 1) The APDU contained in the D-END *User Data* parameter as the CPDLC-end service *CPDLC Message* parameter value, if provided, and

- 2) The abstract value “rejected” as the CPDLC-end service *Result* parameter value, and

- b) Enter the *DIALOGUE* state.

2.3.5.5.7 CPDLC-start Service Request

2.3.5.5.7.1 Upon receipt of a CPDLC-start service request, if the CPDLC-ground-ASE is in the *IDLE* state, the CPDLC-ground-ASE shall:

- a) Create a GroundPDUs APDU with an UplinkMessage APDU element containing:
 - 1) if provided, the *CPDLC Message* parameter as the UplinkMessage, or
 - 2) else, NULL as the UplinkMessage,
- b) Invoke D-START request with the following:
 - 1) the CPDLC-start service *Called Peer Identifier* parameter value as the D-START *Called Peer ID* parameter value,
 - 2) the CPDLC-start service *Calling Peer Identifier* parameter value as the D-START *Calling Peer ID* parameter value,
 - 3) the D-START *Quality of Service* parameters set as follows:
 - i) if provided, the CPDLC-start service *Class of Communication* parameter value as the D-START *QOS Routing Class* parameter value,
 - ii) The abstract value of “high priority flight safety messages”, as the D-START *QOS Priority* parameter value, and
 - iii) The abstract value of “low” as the D-START *QOS Residual Error Rate* parameter value, and
 - 4) the APDU as the D-START *User Data* parameter value;
- c) Start timer t_{start} and
- d) Enter the *START-REQ* state.

2.3.5.5.8 CPDLC-start Service Response

2.3.5.5.8.1 Upon receipt of a CPDLC-start service response, if the CPDLC-ground-ASE is in the *START-IND* state and the CPDLC-start service *Result* parameter has the abstract value “accepted” and the CPDLC-start service *Reject Reason* parameter is not provided, and DSC has the abstract value “false”, the CPDLC-ground-ASE shall:

- a) Invoke D-START response with the abstract value “accepted” as the D-START *Result* parameter value, and
- b) Enter the *DIALOGUE* state.

2.3.5.5.8.2 Upon receipt of a CPDLC-start service response, if the CPDLC-ground-ASE is in the *START-IND* state and the CPDLC-start service *Result* parameter has the abstract value “rejected” and DSC has the abstract value “false”, the CPDLC-ground-ASE shall:

- a) If the CPDLC-start service *Reject Reason* parameter is provided, create a GroundPDUs APDU with an ATCUplinkMessage APDU element based on the *Reject Reason* parameter,
- b) Invoke D-START response with the following:
 - 1) The APDU as the D-START *User Data* parameter value; if the *Reject Reason* parameter was provided, and
 - 2) the abstract value “rejected (permanent)” as the D-START *Result* parameter value, and
- c) Enter the *IDLE* state.

2.3.5.5.9 DSC-start Service Response

2.3.5.5.9.1 Upon receipt of a DSC-start service response, if the CPDLC-ground-ASE is in the *START-IND* state and the DSC-start service *Result* parameter has the abstract value “accepted” and DSC has the abstract value “true”, the CPDLC-ground-ASE shall:

- a) Invoke D-START response with the abstract value “accepted” as the D-START *Result* parameter value, and
- b) Enter the *DIALOGUE* state.

2.3.5.5.9.2 Upon receipt of a DSC-start service response, if the CPDLC-ground-ASE is in the *START-IND* state and the DSC-start service *Result* parameter has the abstract value “rejected” and DSC has the abstract value “true”, the CPDLC-ground-ASE shall:

- a) If the DSC-start service *Reject Reason* parameter is provided, create a GroundPDUs APDU with an ATCUplinkMessage APDU element based on the *Reject Reason* parameter,

- b) Invoke D-START response with the following:
 - 1) The APDU element as D-START *User Data* parameter value; if the *Reject Reason* parameter was provided, and
 - 2) the abstract value “rejected (permanent)” as the D-START *Result* parameter value,
- c) Set DSC to the abstract value “false”, and
- d) Enter the *IDLE* state.

2.3.5.5.10 CPDLC-message Service Request

2.3.5.5.10.1 Upon receipt of a CPDLC-message service request, if the CPDLC-ground-ASE is in the *DIALOGUE* state, the CPDLC-ground-ASE shall:

- a) Create a GroundPDUs APDU with an ATCUplinkMessage APDU-element based on the CPDLC-message service *CPDLC Message* parameter,
- b) Invoke D-DATA request with the APDU as the D-DATA *User Data* parameter value, and
- c) Remain in the *DIALOGUE* state.

2.3.5.5.10.2 Upon receipt of a CPDLC-message service request, if the CPDLC-ground-ASE is in the *END* state and DSC has the abstract value “true”, the CPDLC-ground-ASE shall:

- a) Create a GroundPDUs APDU with an ATCUplinkMessage APDU-element based on the CPDLC-message service *CPDLC Message* parameter,
- b) Invoke D-DATA request with the APDU as the D-DATA *User Data* parameter value, and
- c) Remain in the *END* state.

2.3.5.5.11 CPDLC-end Service Request

2.3.5.5.11.1 Upon receipt of a CPDLC-end service request, if the CPDLC-ground-ASE is in the *DIALOGUE* state and DSC has the abstract value “false”, the CPDLC-ground-ASE shall:

- a) Create a GroundPDUs APDU with an ATCUplinkMessage APDU-element based on the CPDLC-end service *CPDLC Message* parameter, if provided,
- b) Invoke D-END request with the APDU as the D-END *User Data* parameter value, if provided and,
- c) Enter the *END* state.

2.3.5.5.12 DSC-end Service Response

2.3.5.5.12.1 Upon receipt of a DSC-end service response, if the CPDLC-ground-ASE is in the *END* state and DSC has the abstract value “true”, and the DSC-end service *Result* parameter has the abstract value “accepted”, the CPDLC-ground-ASE shall:

- a) Create a GroundPDUs APDU with an ATCUplinkMessage APDU-element based on the DSC-end service *CPDLC Message* parameter, if provided,
- b) Invoke D-END response with the following:
 - 1) the APDU as the D-END *User Data* parameter; if provided, and
 - 2) the abstract value “accepted” as the D-END *Result* parameter value,
- c) Set DSC to the abstract value “false”, and
- d) Enter the *IDLE* state.

2.3.5.5.12.2 Upon receipt of a DSC-end service response, if the CPDLC-ground-ASE is in the *END* state and DSC has the abstract value “true”, and the DSC-end service *Result* parameter has the abstract value “rejected”, the CPDLC-ground-ASE shall:

- a) Create a GroundPDUs APDU with an ATCUplinkMessage APDU-element based on the DSC-end service *CPDLC Message* parameter, if provided,
- b) Invoke D-END response with the following:
 - 1) the APDU as the D-END *User Data* parameter; if provided, and
 - 2) the abstract value “rejected” as the D-END *Result* parameter value, and
- c) Enter the *DIALOGUE* state.

2.3.5.5.13 CPDLC-forward Service Request

2.3.5.5.13.1 Upon receipt of a CPDLC-forward service request, if the CPDLC-ground-ASE is in the *IDLE* state, the CPDLC-ground-ASE shall:

- a) Create a GroundPDUs APDU with an ATCForwardMessage APDU element based on the CPDLC-forward service *CPDLC Message* parameter,
- b) Invoke D-START request with the following:
 - 1) the CPDLC-forward service *Called Facility Designation* parameter value as the D-START *Called Peer ID* parameter value,
 - 2) the CPDLC-start service *Calling Facility Designation* parameter value as the D-START *Calling Peer ID* parameter value,

- 3) the D-START *Quality of Service* parameters set as follows:
 - i) if provided, the CPDLC-start service *Class of Communication* parameter value as the D-START QOS *Routing Class* parameter value,
 - ii) The abstract value of “high priority flight safety messages”, as the D-START QOS *Priority* parameter value, and
 - iii) The abstract value of “low” as the D-START QOS *Residual Error Rate* parameter value, and
 - 4) the APDU as the D-START *User Data* parameter value;
- c) Start timer t_{start} and
 - d) Enter the *FORWARD* state.

2.3.5.5.14 CPDLC-user-abort Service Request

2.3.5.5.14.1 Upon receipt of a CPDLC-user-abort service request, if the CPDLC-ground-ASE is not in the *IDLE* state, the CPDLC-ground-ASE shall:

- a) Stop any timer,
- b) If the CPDLC-user-abort service *Reason* parameter is provided, create a GroundPDUs APDU with a CPDLCUserAbortReason APDU-element based on the CPDLC-user-abort service *Reason* parameter,
- c) Else create a GroundPDUs APDU with a CPDLCUserAbortReason [undefined] APDU-element,
- d) Invoke D-ABORT request with the following:
 - 1) the D-ABORT *Originator* parameter set to the abstract value “user”, and
 - 2) the APDU as the D-ABORT *User Data* parameter value, and
- e) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- f) Enter the *IDLE* state.

2.3.5.5.15 D-ABORT Indication

2.3.5.5.15.1 Upon receipt of a D-ABORT indication, if the CPDLC-ground-ASE is not in the *IDLE* state and the D-ABORT *Originator* parameter is “user” and the D-ABORT *User Data* parameter contains an AircraftPDUs [CPDLCUserAbortReason] APDU, the CPDLC-ground-ASE shall:

- a) Stop any timer,

- b) If the CPDLC-ground-user is an active user, invoke CPDLC-user-abort service indication with the APDU contained in the D-ABORT *User Data* parameter as the CPDLC-user-abort service *Reason* parameter value,
- c) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- d) Enter the *IDLE* state.

2.3.5.5.15.2 Upon receipt of a D-ABORT indication, if the CPDLC-ground-ASE is not in the *IDLE* state, and if the D-ABORT *Originator* parameter is “provider” and if the D-ABORT *User Data* parameter is provided, the D-ABORT *User Data* parameter contains either an AircraftPDUs [CPDLCProviderAbortReason] APDU or a GroundPDUs [CPDLCProviderAbortReason] APDU , the CPDLC-ground-ASE shall:

- a) Stop any timer,
- b) If the CPDLC-ground-user is an active user, invoke CPDLC-provider-abort service indication with the D-ABORT *User Data* parameter as the CPDLC-provider-abort service *Reason* parameter value, if provided,
- c) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- d) Enter the *IDLE* state.

2.3.5.5.16 D-P-ABORT Indication

2.3.5.5.16.1 Upon receipt of a D-P-ABORT indication, if the CPDLC-ground-ASE is not in the *IDLE* state, the CPDLC-ground-ASE shall:

- a) Stop any timer,
- b) If the CPDLC-ground-user is an active user, invoke CPDLC-provider-abort service indication with the CPDLC-provider-abort service *Reason* parameter set to the abstract value “communication-service-failure”,
- c) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- d) Enter the *IDLE* state.

2.3.5.6 CPDLC-Ground-ASE Exception Handling

2.3.5.6.1 A Timer Expires

2.3.5.6.1.1 If a CPDLC-ground-ASE detects that a timer has expired, that CPDLC-ground-ASE shall:

- a) Interrupt any current activity,
- b) Create a GroundPDUs APDU with a CPDLCProviderAbortReason [timer-expired] APDU message element,

- c) Invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter value, and
- d) If the CPDLC-ground-user is an active user, invoke CPDLC-provider-abort service indication with the abstract value “timer-expired” as the CPDLC-provider abort service *Reason* parameter value”,
- e) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- f) Enter the *IDLE* state.

2.3.5.6.2 Unrecoverable System Error

2.3.5.6.2.1 **Recommendation.** — *If a CPDLC-ground-ASE has an unrecoverable system error, the CPDLC-ground-ASE should:*

- a) *Stop any timer,*
- b) *Create a GroundPDUs APDU with a CPDLCProviderAbortReason [undefined-error] APDU message element,*
- c) *Invoke D-ABORT request with:*
 - 1) *the abstract value “provider” as the D-ABORT Originator parameter value, and*
 - 2) *the APDU as the D-ABORT User Data parameter value, and*
- d) *If the CPDLC-ground-user is an active user, invoke CPDLC-provider-abort service indication with the abstract value “undefined-error” as the CPDLC-provider abort service Reason parameter value”,*
- e) *If DSC has the abstract value “true”, set DSC to the abstract value “false”, and*
- f) *Enter the IDLE state.*

2.3.5.6.3 Invalid PDU

2.3.5.6.3.1 If the *User Data* parameter of a D-END confirmation with *Result* parameter set to the abstract value “rejected”, a D-START indication, a D-DATA indication, or a D-END indication, does not contain a valid PDU, the CPDLC-ground-ASE shall:

- a) Stop any timer,

- b) Create a GroundPDUs APDU with a CPDLCProviderAbortReason [invalid-PDU] APDU message element,
- c) Invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter value, and
- d) If the CPDLC-ground-user is an active user, invoke CPDLC-provider-abort service indication with the abstract value “invalid-PDU” as the CPDLC-provider abort service *Reason* parameter value,
- e) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- f) Enter the *IDLE* state.

2.3.5.6.3.2 If the *User Data* parameter of a D-START confirmation with *Result* set to the abstract value “rejected (permanent)”, or a D-END confirmation with *Result* set to the abstract value “accepted”, is not a valid PDU the CPDLC-ground-ASE shall:

- a) Stop any timer,
- b) If the CPDLC-ground-user is an active user, invoke CPDLC-provider-abort service indication with the CPDLC-provider-abort service *Reason* parameter set to the abstract value “invalid-PDU”,
- c) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- d) Enter the *IDLE* state.

2.3.5.6.4 Protocol Error

2.3.5.6.4.1 If the *User Data* parameter of a D-START indication, D-DATA indication, or a D-END indication is a valid PDU, but is not a PDU for which action is described within a given state as defined in 2.3.5.5, the CPDLC-ground-ASE shall:

- a) Stop any timer,
- b) Create a GroundPDUs APDU with a CPDLCProviderAbortReason [protocol-error] APDU message element,
- c) Invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter value, and

- d) If the CPDLC-ground-user is an active user, invoke CPDLC-provider-abort service indication with the abstract value “protocol-error” as the CPDLC-provider abort service *Reason* parameter value”,
- e) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- f) Enter the *IDLE* state.

2.3.5.6.4.2 If D-START confirmation with the *Result* parameter set to the abstract value “accepted” contains a *User Data* parameter, the CPDLC-ground-ASE shall:

- a) Stop any timer,
- b) Create a GroundPDUs APDU with a CPDLCProviderAbortReason [protocol-error] APDU message element,
- c) Invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter value, and
- d) If the CPDLC-ground-user is an active user, invoke CPDLC-provider-abort service indication with the abstract value “protocol-error” as the CPDLC-provider-abort service *Reason* parameter value,
- e) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- f) Enter the *IDLE* state.

2.3.5.6.4.3 If the *User Data* parameter of a D-END confirmation is a valid PDU, but is not a permitted PDU for which action is described within a given state as defined in 2.3.5.5, the CPDLC-ground-ASE shall:

- a) Stop any timer,
- b) If the D-END *Result* parameter is set to the abstract value “rejected”, then
 - 1) Create a GroundPDUs APDU with a CPDLCProviderAbortReason [protocol-error] APDU message element,
 - 2) Invoke D-ABORT request with:
 - i) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - ii) the APDU as the D-ABORT *User Data* parameter value, and

- c) If the CPDLC-ground-user is an active user, invoke CPDLC-provider-abort service indication with the abstract value “protocol-error” as the CPDLC-provider-abort service *Reason* parameter value,
- d) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- e) Enter the *IDLE* state.

2.3.5.6.4.4 Upon receipt of a Dialogue service primitive for which there are no instructions in 2.3.5.3 (i.e. the primitive was not expected or was expected under other conditions or with other parameter values), the CPDLC-ground-ASE shall:

- a) Stop any timer,
- b) Create a GroundPDUs APDU with a CPDLCProviderAbortReason [protocol-error] APDU message element,
- c) If a dialogue exists, invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter value, and
- d) If the CPDLC-ground-user is an active user, invoke CPDLC-provider-abort service indication with the abstract value “protocol-error” as the CPDLC-provider-abort service *Reason* parameter value,
- e) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- f) Enter the *IDLE* state.

2.3.5.6.5 D-START Confirmation Result or Reject Source Not as Expected

2.3.5.6.5.1 If a D-START confirmation *Result* parameter has the abstract value of “rejected (transient)” or if the *Reject Source* parameter has the abstract value of “DS provider”, the CPDLC-ground-ASE shall:

- a) Stop any timer,
- b) if the CPDLC-ground-user is an active user, invoke CPDLC-provider-abort service indication with the CPDLC-provider-abort service *Reason* parameter set to the abstract value “communication-service-error”,
- c) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- d) Enter the *IDLE* state.

2.3.5.6.6 D-START Indication Quality of Service Not as Expected

2.3.5.6.6.1 If a D-START indication *QOS Priority* parameter does not have the abstract value of “high priority flight safety messages” or if the *QOS Residual Error Rate* parameter does not have the abstract value of “low”, or if the *QOS Routing Class* parameter does not have one of the abstract values specified in Table 2.3.6-1, the CPDLC-ground-ASE shall:

- a) Stop any timer,
- b) Create a GroundPDUs APDU with a CPDLCProviderAbortReason [invalid-QOS-parameter] APDU message element,
- c) Invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter value, and
- d) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- e) Enter the *IDLE* state.

2.3.5.6.7 Expected PDU Missing

2.3.5.6.7.1 If the *User Data* parameter of a D-START indication or a D-DATA indication does not contain a PDU, the CPDLC-ground-ASE shall:

- a) Stop any timer,
- b) Create a GroundPDUs APDU with a CPDLCProviderAbortReason [expected-PDU-missing] APDU message element,
- c) Invoke D-ABORT request with:
 - 1) the abstract value “provider” as the D-ABORT *Originator* parameter value, and
 - 2) the APDU as the D-ABORT *User Data* parameter value, and
- d) If the CPDLC-ground-user is an active user, invoke CPDLC-provider-abort service indication with the abstract value “not-permitted-PDU” as the CPDLC-provider-abort service *Reason* parameter value,
- e) If DSC has the abstract value “true”, set DSC to the abstract value “false”, and
- f) Enter the *IDLE* state.

2.3.5.7 CPDLC ASE State Tables

2.3.5.7.1 Priority

2.3.5.7.1.1 If the state tables shown for the CPDLC-air-ASE and the CPDLC-ground-ASE shown below conflict with textual statements made elsewhere in this document, the textual statements shall take precedence.

Note 1.— In the following state tables, the statement “cannot occur” means that if the implementation conforms to the SARPs, it is impossible for this event to occur. If the event does occur, this implies that there is an error in the implementation. If such a situation is detected, it is suggested that the ASE aborts with the error “unrecoverable system error”.

Note 2.— In the following state tables, the statement “not permitted” means that the implementation must prevent this event from occurring through some local means. If the event does occur this implies that there is an error in the implementation. If such a situation is detected, it is suggested that the ASE performs a local rejection of the request rather than aborting the dialogue.

Table 2.3.5-2. CPDLC-Air-ASE State Table

STATE → EVENT ↓	IDLE	START-REQ	START-IND	DIALOGUE	END
Dialogue Service Events					
D-START Indication APDU = UplinkMessage	●CPDLC-start indication ⇒ <i>START-IND</i>	cannot occur	cannot occur	cannot occur	cannot occur
D-START Confirmation <i>Result</i> “accepted”, DSC = “false” No <i>User Data</i>	cannot occur	●Stop timer t_{start} ●CPDLC-start confirmation ⇒ <i>DIALOGUE</i>	cannot occur	cannot occur	cannot occur
D-START Confirmation <i>Result</i> “rejected (permanent)” and Reject Source “DS user”, DSC=“false” if <i>User Data</i> , APDU = ATCUplinkMessage	cannot occur	●Stop timer t_{start} ●CPDLC-start confirmation ⇒ <i>IDLE</i>	cannot occur	cannot occur	cannot occur
D-START Confirmation <i>Result</i> “accepted”, DSC = “true” No <i>User Data</i>	cannot occur	●Stop timer t_{start} ●DSC-start confirmation, ⇒ <i>DIALOGUE</i>	cannot occur	cannot occur	cannot occur
D-START Confirmation <i>Result</i> “rejected (permanent)” and Reject Source “DS user”, DSC= “true” if <i>User Data</i> , APDU = ATCUplinkMessage	cannot occur	●Stop timer t_{start} ●DSC-start confirmation ●Set DSC = “false” ⇒ <i>IDLE</i>	cannot occur	cannot occur	cannot occur
D-DATA Indication APDU = ATCUplinkMessage	cannot occur	cannot occur	cannot occur	●CPDLC-message indication ⇒ <i>DIALOGUE</i>	●if DSC=“true” ●CPDLC-message indication ⇒ <i>END</i> ●else not permitted
D-END Indication: DSC=“false” if <i>User Data</i> , APDU = ATCUplinkMessage	cannot occur	cannot occur	cannot occur	●CPDLC-end indication ⇒ <i>END</i>	cannot occur
D-END Confirmation: DSC=“true” <i>Result</i> “accepted” if <i>User Data</i> , APDU = ATCUplinkMessage	cannot occur	cannot occur	cannot occur	cannot occur	●DSC-end confirmation ●Set DSC “false” ⇒ <i>IDLE</i>

STATE → EVENT ↓	IDLE	START-REQ	START-IND	DIALOGUE	END
D-END Confirmation: DSC="true", Result "rejected" if User Data, APDU = ATCUplinkMessage	cannot occur	cannot occur	cannot occur	cannot occur	●DSC-end confirmation ⇒DIALOGUE
CPDLC-User Events					
CPDLC-start Request	●D-START request ●Start timer t_{start} ⇒START-REQ	not permitted	not permitted	not permitted	not permitted
CPDLC-start Response DSC="false" Result "accepted"	not permitted	not permitted	●D-START response ⇒DIALOGUE	not permitted	not permitted
CPDLC-start Response DSC="false" Result "rejected"	not permitted	not permitted	●D-START response ⇒IDLE	not permitted	not permitted
DSC-start Request	●D-START request ●set DSC = "true" ●Start timer t_{start} ⇒START-REQ	not permitted	not permitted	not permitted	not permitted
CPDLC-message Request	not permitted	not permitted	not permitted	●D-DATA request ⇒DIALOGUE	●if DSC="false" ●D-DATA request ⇒END ●else not permitted
CPDLC-end Service Response DSC = "false" Result "accepted"	cannot occur	cannot occur	cannot occur	not permitted	●D-END response ⇒IDLE
CPDLC-end Service Response DSC = "false" Result "rejected"	cannot occur	cannot occur	cannot occur	not permitted	●D-END response ⇒DIALOGUE
DSC-end Request: DSC = "true"	not permitted	not permitted	not permitted	●D-END request ⇒END	not permitted
ABORT Events					
CPDLC-user-abort Request	not permitted	●Stop timer t_{start} , if set ●D-ABORT request ●If DSC = "true", set DSC = "false" ⇒IDLE	●Stop timer t_{start} , if set ●D-ABORT request ●If DSC = "true", set DSC = "false" ⇒IDLE	●D-ABORT request ●If DSC = "true", set DSC = "false" ⇒IDLE	●D-ABORT request ●If DSC = "true", set DSC = "false" ⇒IDLE

STATE → EVENT ↓	IDLE	START-REQ	START-IND	DIALOGUE	END
D-ABORT Indication <i>Originator</i> “user”	cannot occur	●Stop timer T_{start}, if set ●If active user: CPDLC-user-abort indication ●If DSC = “true”, set DSC = “false” ⇒<i>IDLE</i>	●Stop timer T_{start}, if set ●If active user: CPDLC-user-abort indication ●If DSC = “true”, set DSC = “false” ⇒<i>IDLE</i>	●If active user: CPDLC-user-abort indication ●If DSC = “true”, set DSC = “false” ⇒<i>IDLE</i>	●If active user: CPDLC-user-abort indication ●If DSC = “true”, set DSC = “false” ⇒<i>IDLE</i>
D-ABORT Indication <i>Originator</i> “provider”	cannot occur	●Stop timer T_{start}, if set ●If active user: CPDLC-provider-abort indication ●If DSC = “true”, set DSC = “false” ⇒<i>IDLE</i>	●Stop timer T_{start}, if set ●If active user: CPDLC-provider-abort indication ●If DSC = “true”, set DSC = “false” ⇒<i>IDLE</i>	●If active user: CPDLC-provider-abort indication ●If DSC = “true”, set DSC = “false” ⇒<i>IDLE</i>	●If active user: CPDLC-provider-abort indication ●If DSC = “true”, set DSC = “false” ⇒<i>IDLE</i>
D-P-ABORT indication	cannot occur	●Stop timer t_{start}, if set ●If active user: CPDLC-provider-abort indication ●If DSC = “true”, set DSC = “false” ⇒<i>IDLE</i>	●Stop timer t_{start}, if set ●If active user: CPDLC-provider-abort indication ●If DSC = “true”, set DSC = “false” ⇒<i>IDLE</i>	●If active user: CPDLC-provider-abort indication ●If DSC = “true”, set DSC = “false” ⇒<i>IDLE</i>	●If active user: CPDLC-provider-abort indication ●If DSC = “true”, set DSC = “false” ⇒<i>IDLE</i>
T_{start} Expires	cannot occur	●D-ABORT request ●CPDLC-provider-abort indication ●If DSC = “true”, set DSC = “false” ⇒<i>IDLE</i>	cannot occur	cannot occur	cannot occur

Table 2.3.5-3. CPDLC-Ground-ASE State Table

STATE → EVENT ↓	IDLE	START-REQ	START-IND	DIALOGUE	END	FORWARD
Dialogue Service Events						
D-START Indication APDU Aircraft mode “cpdlc”	●CPDLC-start indication ⇒<i>START-IND</i>	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur
D-START Indication APDU Aircraft mode “dsc”	●DSC-start indication, ●Set DSC = “true” ⇒<i>START-IND</i>	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur
D-START Indication APDU Forward version equal and function supported	●CPDLC-forward indication ●D-START response ⇒<i>IDLE</i>	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur
D-START Indication APDU Forward version not equal or function not supported	●D-START response ⇒<i>IDLE</i>	cannot occur	cannot occur	cannot occur	cannot occur	cannot occur
D-START Confirmation <i>Result</i> “accepted” DSC = “false”	cannot occur	●Stop timer t_{start} ●CPDLC-start confirmation ⇒<i>DIALOGUE</i>	cannot occur	cannot occur	cannot occur	cannot occur
D-START Confirmation <i>Result</i> “rejected (permanent)” and <i>Reject Source</i> “DS user” DSC=“false”	cannot occur	●Stop timer t_{start} ●CPDLC-start confirmation ⇒<i>IDLE</i>	cannot occur	cannot occur	cannot occur	●Stop timer t_{start} ●CPDLC-forward confirmation ⇒<i>IDLE</i>
D-START Confirmation <i>Result</i> “accepted” DSC = “true”	cannot occur	●Stop timer t_{start} ●DSC-start confirmation, ⇒<i>DIALOGUE</i>	cannot occur	cannot occur	cannot occur	cannot occur
D-START Confirmation <i>Result</i> “rejected (permanent)” and <i>Reject Source</i> “DS user” DSC = “true”	cannot occur	●Stop timer t_{start} ●DSC-start confirmation ●Set DSC = “false” ⇒<i>IDLE</i>	cannot occur	cannot occur	cannot occur	cannot occur

STATE → EVENT ↓	IDLE	START-REQ	START-IND	DIALOGUE	END	FORWARD
D-DATA Indication	cannot occur	cannot occur	cannot occur	●CPDLC-message indication ⇒<i>DIALOGUE</i>	●if DSC = “false” ●CPDLC-message indication ⇒<i>END</i> ●else not permitted	cannot occur
D-END Indication: DSC=“true”	cannot occur	cannot occur	cannot occur	●DSC-end indication ⇒<i>END</i>	cannot occur	cannot occur
D-END Confirmation: DSC = “false” Result “accepted”	cannot occur	cannot occur	cannot occur	cannot occur	●CPDLC-end confirmation ⇒<i>IDLE</i>	cannot occur
D-END Confirmation: DSC = “false” Result “rejected”	cannot occur	cannot occur	cannot occur	cannot occur	●CPDLC-end confirmation ⇒<i>DIALOGUE</i>	cannot occur
CPDLC-User Events						
CPDLC-start Request	●D-START request ●Start timer t_{start} ⇒<i>START-REQ</i>	not permitted	not permitted	not permitted	not permitted	not permitted
CPDLC-start Response DSC = “false” Result “accepted”	not permitted	not permitted	●D-START response ⇒<i>DIALOGUE</i>	not permitted	not permitted	not permitted
CPDLC-start Response DSC = “false” Result “rejected”	not permitted	not permitted	●D-START response ⇒<i>IDLE</i>	not permitted	not permitted	not permitted
DSC-start Response DSC = “true” Result “accepted”	not permitted	not permitted	●D-START response ⇒<i>DIALOGUE</i>	not permitted	not permitted	not permitted
DSC-start Response DSC = “true” Result “rejected”	not permitted	not permitted	●D-START response ●Set DSC= “false” ⇒<i>IDLE</i>	not permitted	not permitted	not permitted

STATE → EVENT ↓	IDLE	START-REQ	START-IND	DIALOGUE	END	FORWARD
CPDLC-message Request	not permitted	not permitted	not permitted	•D-DATA request ⇒<i>DIALOGUE</i>	•If DSC= “true” •D-DATA request ⇒<i>END</i> •Else not permitted	not permitted
CPDLC-end Request: DSC = “false”	not permitted	not permitted	not permitted	•D-END request ⇒<i>END</i>	not permitted	not permitted
DSC-end Service Response DSC = “true” Result “accepted”	cannot occur	cannot occur	cannot occur	not permitted	•D-END response •Set DSC = “false” ⇒<i>IDLE</i>	not permitted
DSC-end Service Response DSC = “true” Result “rejected”	cannot occur	cannot occur	cannot occur	not permitted	•D-END response ⇒<i>DIALOGUE</i>	not permitted
CPDLC-forward Request	•D-START request •Start timer t_{start} ⇒<i>FORWARD</i>	not permitted	not permitted	not permitted	not permitted	not permitted
ABORT Events						
CPDLC-user-abort Request	not permitted	•Stop timer t_{start}, if set •D-ABORT request •If DSC = “true”, set DSC= “false” ⇒<i>IDLE</i>	•Stop timer t_{start}, if set •D-ABORT request •If DSC = “true”, set DSC= “false” ⇒<i>IDLE</i>	•D-ABORT request •If DSC = “true”, set DSC= “false” ⇒<i>IDLE</i>	•D-ABORT request •If DSC = “true”, set DSC= “false” ⇒<i>IDLE</i>	•Stop timer t_{start}, if set •D-ABORT request ⇒<i>IDLE</i>
D-ABORT Indication Originator “provider”	cannot occur	•Stop timer T_{start}, if set •If active user: CPDLC-provider-abort indication •If DSC = “true”, set DSC= “false” ⇒<i>IDLE</i>	•Stop timer T_{start}, if set •If active user: CPDLC-provider-abort indication •If DSC = “true”, set DSC= “false” ⇒<i>IDLE</i>	•If active user: CPDLC-provider-abort indication •If DSC = “true”, set DSC= “false” ⇒<i>IDLE</i>	•If active user: CPDLC-provider-abort indication •If DSC = “true”, set DSC= “false” ⇒<i>IDLE</i>	•Stop timer T_{start}, if set •If active user: CPDLC-provider-abort indication ⇒<i>IDLE</i>

STATE → EVENT ↓	IDLE	START-REQ	START-IND	DIALOGUE	END	FORWARD
D-ABORT Indication <i>Originator</i> “user”	cannot occur	● Stop timer T_{start}, if set ● If active user: CPDLC-user-abort indication ● If DSC = “true”, set DSC= “false” ⇒ <i>IDLE</i>	● Stop timer T_{start}, if set ● If active user: CPDLC-user-abort indication ● If DSC = “true”, set DSC= “false” ⇒ <i>IDLE</i>	● If active user: CPDLC-user-abort indication ● If DSC = “true”, set DSC= “false” ⇒ <i>IDLE</i>	● If active user: CPDLC-user-abort indication ● If DSC = “true”, set DSC= “false” ⇒ <i>IDLE</i>	● Stop timer T_{start}, if set ● If active user: CPDLC-user-abort indication ⇒ <i>IDLE</i>
D-P-ABORT indication	cannot occur	● Stop timer t_{start}, if set ● If active user: CPDLC-provider-abort indication ● If DSC = “true”, set DSC= “false” ⇒ <i>IDLE</i>	● Stop timer t_{start}, if set ● If active user: CPDLC-provider-abort indication ● If DSC = “true”, set DSC= “false” ⇒ <i>IDLE</i>	● If active user: CPDLC-provider-abort indication ● If DSC = “true”, set DSC= “false” ⇒ <i>IDLE</i>	● If active user: CPDLC-provider-abort indication ● If DSC = “true”, set DSC= “false” ⇒ <i>IDLE</i>	● Stop timer t_{start}, if set ● If active user: CPDLC-provider-abort indication ⇒ <i>IDLE</i>
T_{start} Expires	cannot occur	● D-ABORT request ● CPDLC-provider-abort indication ● If DSC = “true”, set DSC= “false” ⇒ <i>IDLE</i>	cannot occur	cannot occur	cannot occur	● D-ABORT request ● CPDLC-provider -abort indication ⇒ <i>IDLE</i>

2.3.6 COMMUNICATION REQUIREMENTS

2.3.6.1 Encoding Rules

2.3.6.1.1 The CPDLC application shall use PER as defined in ISO/IEC 8825-2, using the Basic Unaligned variant to encode/decode the ASN.1 message structure and content specified in 2.3.4.

2.3.6.2 Dialogue Service Requirements

2.3.6.2.1 Primitive Requirements

2.3.6.2.1.1 Where dialogue service primitives, that is D-START, D-DATA, D-END, D-ABORT, and D-P-ABORT are described as being invoked in 2.3.5, the CPDLC-ground-ASE and the CPDLC-air-ASE shall exhibit external behavior consistent with the dialogue service, as described in 4.2 having been implemented and its primitives invoked.

2.3.6.2.2 Quality-of-Service Requirements

2.3.6.2.2.1 The application service priority for CPDLC shall have the abstract value of “high priority flight safety messages”.

2.3.6.2.2.2 The RER Quality of Service Parameter of the D-START shall be set to the abstract value of “low”.

2.3.6.2.2.3 The CPDLC-ASE shall map the CPDLC-start service or DSC-start service *Class of Communication* parameter abstract value to the ATSC routing class abstract value part of the D-START QOS parameter as presented in Table 2.3.6-1.

Table 2.3.6-1. Mapping Between Class of Communication and Routing Class Abstract Values

Class of Communication Abstract Value	Routing Class Abstract Value
No Preference	No Traffic Type Policy Preference
A	Traffic follows Class A ATSC route(s)
B	Traffic follows Class B ATSC route(s)
C	Traffic follows Class C ATSC route(s)
D	Traffic follows Class D ATSC route(s)
E	Traffic follows Class E ATSC route(s)
F	Traffic follows Class F ATSC route(s)
G	Traffic follows Class G ATSC route(s)
H	Traffic follows Class H ATSC route(s)

Note.— ATSC values are defined in 1.3.

2.3.7 CPDLC USER REQUIREMENTS

2.3.7.1 General

Note 1.— Requirements imposed on CPDLC-user concerning CPDLC messages and interfacing with the CPDLC-ASEs are presented in this chapter.

Note 2.— Where reference is made to the “CPDLC-user”, this implies both the CPDLC-air-user and the CPDLC-ground-user.

Note 3.— A CPDLC message is:

- a) *what the CPDLC-user provides the CPDLC-service as the CPDLC Message or Reject Reason parameter, when invoking a CPDLC service request or response primitive, or*
- b) *what the CPDLC-user receives in the same parameters from the CPDLC service indication or confirmation primitives.*

Note 4.— The terms CPDLC message, message, uplink message and downlink message are used interchangeably, and equate to a CPDLC message. When the terms “send” and “transmit” are used this means that the CPDLC-user has invoked a CPDLC service request or response primitive. When the term “receive” is used this means that a CPDLC indication or confirmation primitive parameter containing a CPDLC message has been provided by the CPDLC service.

2.3.7.1.1 General CPDLC Service Requirements

2.3.7.1.1.1 A CPDLC-ground-user shall invoke CPDLC-start service, DSC-start service, CPDLC-message service, CPDLC-end service, and DSC-end service only when communicating with a CPDLC-air-user.

2.3.7.1.1.2 A CPDLC-ground-user shall invoke CPDLC-forward service, only when communicating with another CPDLC-ground-user.

Note 1.— When a CPDLC-user invokes the CPDLC-start service, the DSC-start service, or the CPDLC-forward service and requires a particular class of communication service, the CPDLC-user sets the Class of Communication Service parameter.

Note 2.— When a CPDLC-user does not require a particular class of communication, the user does not set the Class of Communication Service parameter.

2.3.7.2 CPDLC Message Generation Requirements

Note 1.— A response message is a message which is a reply to a received message. It contains a message reference number identical to the message identification number of the message to which it refers. Only response messages contain a message reference number.

Note 2.— Message response attributes dictate a) if a response is required or prohibited; b) if a response is required, dictate the permitted response messages.

Note 3.— A closure response message is a reply to a message or series of messages which terminates a sequence of message exchanges. However due to the multiple element capability of a CPDLC message, a closure message may contain message element(s) in addition to the required closure message element that initiate a new sequence of messages.

2.3.7.2.1 Message Composition

2.3.7.2.1.1 A CPDLC message shall be composed of a message header, and from one to five message elements.

2.3.7.2.1.2 For air/ground messages, the message header shall be composed of a message identification number, a message reference number, if required, a time stamp, and a logical acknowledgment requirement (optional).

2.3.7.2.1.3 For ground/ground messages, the message header shall be composed of a time stamp, the aircraft flight identification, and the aircraft address to which the message refers.

2.3.7.2.1.4 A message element shall consist of a message element identifier, data as indicated by the specified message element, and associated message element attributes.

2.3.7.2.1.5 For each CPDLC message the CPDLC-user sends air/ground it shall provide the following information:

- a) a message identification number,
- b) a message reference number only if the message is a response message,
- c) date and time,
- d) a logical acknowledgment indication, if required,
- e) from one to five message element identifiers, and
- f) data as required for each message element identification included.

2.3.7.2.1.6 For each CPDLC message the CPDLC-user sends ground/ground it shall provide the following information:

- a) the aircraft flight identification to which the ground/ground message refers,
- b) date and time,
- c) from one to five message element identifiers, and
- d) data as required for each message element identification included.

2.3.7.2.2 Message Identification Number

Note 1.— A message identification number pertains to a single peer to peer dialogue.

Note 2.— Message identification numbers used by a CPDLC ground system for uplink messages to an aircraft have no relationship to the message identification numbers used by the same ground system with another aircraft.

Note 3.— Similarly, message identification numbers used by a CPDLC aircraft for downlink messages to a CPDLC ground system have no relationship to the message identification numbers used by the same aircraft with another ground system.

Note 4.— There is no relationship between message identification numbers assigned and managed by a CPDLC ground system and those message identification numbers assigned and managed by the aircraft.

2.3.7.2.2.1 The message identification number provided by the CPDLC-user shall be different from any other message identification number currently in use.

2.3.7.2.2.2 A message identification number shall be deemed currently in use until:

- a) if the message does not allow a response, the message is sent, or
- b) if the message requires a response, the closure response is received.

2.3.7.2.2.3 If a CPDLC or DSC dialogue is terminated, all message identification numbers pertaining to that dialogue shall be considered available.

2.3.7.2.3 Message Elements That Cannot Be Combined With Other Message Elements in a Message

2.3.7.2.3.1 The LOGICAL ACKNOWLEDGMENT message element (uplink message element 227 and downlink message element 100) shall be sent only as a single message element CPDLC message.

2.3.7.2.3.2 The NEXT DATA AUTHORITY [facility] message element (uplink message element 160) shall be sent only as a single message element CPDLC message.

2.3.7.2.4 Restriction on Route Clearance Variable Message Elements

2.3.7.2.4.1 A CPDLC message shall contain no more than two message elements with the [routeClearance] variable.

2.3.7.3 CPDLC Message Receipt Requirements

2.3.7.3.1 Message Attributes

Note 1.— Message attributes dictate certain message handling requirements for the CPDLC-user receiving a message. Each CPDLC message has Urgency, Alert, and Response attributes.

Note 2.— Message element attribute table entries are listed in order of precedence (i.e., a precedence value of 1 is highest followed by 2, etc.).

2.3.7.3.1.1 When a message contains a single message element, the message attributes shall be the message element attributes.

2.3.7.3.1.2 When a message contains multiple message elements, the highest precedence message element attribute for each attribute type associated with any element in the message shall be the message attribute for each attribute type for the entire message.

2.3.7.3.2 Urgency Requirements

Note.— The Urgency (URG) attribute delineates the queuing requirements for received messages that are displayed to the end-user. The same Urgency attribute types are used for both air/ground and ground/ground messages.

2.3.7.3.2.1 Each message element shall have associated Urgency attributes with precedence as defined in table 2.3.7-1.

Table 2.3.7-1. Urgency Attribute (Air/Ground and Ground/Ground)

Type	Description	Precedence
D	Distress	1
U	Urgent	2
N	Normal	3
L	Low	4

2.3.7.3.2.2 When a CPDLC-user queues received messages, messages with the highest Urgency type shall be placed at the beginning of the queue.

2.3.7.3.2.3 When a CPDLC-user queues received messages, messages with the same Urgency type shall be queued in order of receipt.

2.3.7.3.3 Alerting Requirements

Note.— The alert (ALRT) attribute delineates the type of end-user alerting required by the CPDLC-user upon message receipt. The same Alert attribute types are used for both air/ground and ground/ground messages.

2.3.7.3.3.1 Each message element shall have associated Alert attributes with precedence as defined in table 2.3.7-2.

Table 2.3.7-2. Alert Attribute (Air/Ground and Ground/Ground)

Type	Description	Precedence
H	High	1
M	Medium	2
L	Low	3
N	No alerting required	4

2.3.7.3.3.2 Upon receipt of a CPDLC message, the CPDLC-user shall provide one of three distinct alerts as determined by the received message alert attribute.

2.3.7.3.4 Response Attribute

Note.— The response (RESP) attribute mandates CPDLC-user response requirements for a given message element. Response message attribute only apply to air/ground messages.

2.3.7.3.4.1 Each uplink message element shall have associated Response attributes with precedence as defined in table 2.3.7-3.

Table 2.3.7-3. Response Attribute (Up-Link)

<i>Type</i>	<i>Response Required</i>	<i>Valid Responses Description</i>	<i>Precedence</i>
W/U	Yes	Response required: WILCO, UNABLE, STANDBY, NOT CURRENT DATA AUTHORITY, NOT AUTHORIZED NEXT DATA AUTHORITY, ERROR, LOGICAL ACKNOWLEDGMENT (only if required)	1
A/N	Yes	Response required: AFFIRM, NEGATIVE, STANDBY, NOT CURRENT DATA AUTHORITY, NOT AUTHORIZED NEXT DATA AUTHORITY, ERROR, LOGICAL ACKNOWLEDGMENT (only if required)	2
R	Yes	Response required: ROGER, UNABLE, STANDBY, NOT CURRENT DATA AUTHORITY, NOT AUTHORIZED NEXT DATA AUTHORITY, ERROR, LOGICAL ACKNOWLEDGMENT (only if required),	3
Y	Yes	Any CPDLC downlink message, LOGICAL ACKNOWLEDGMENT (only if required),	4
N	No, unless logical acknowledgment required	LOGICAL ACKNOWLEDGMENT (only if required), ERROR, NOT CURRENT DATA AUTHORITY or NOT AUTHORIZED NEXT DATA AUTHORITY	5

2.3.7.3.4.2 Each downlink message element shall have associated Response attributes with precedence as defined in table 2.3.7-4.

Table 2.3.7-4. Response Attribute (Down-Link)

<i>Type</i>	<i>Response Required</i>	<i>Valid Responses Description</i>	<i>Precedence</i>
Y	Yes	Any CPDLC uplink message, <i>LOGICAL ACKNOWLEDGMENT</i> (only if required)	1
N	No, unless logical acknowledgment required	<i>LOGICAL ACKNOWLEDGMENT</i> (only if required), <i>ERROR</i> , <i>SERVICE UNAVAILABLE</i> , or <i>FLIGHT PLAN NOT HELD</i>	2

2.3.7.3.5 CPDLC/DSC Distinction

2.3.7.3.5.1 Upon receipt of a CPDLC message the CPDLC-air-user shall provide a distinction between CPDLC messages received from the Current Data Authority and those received from a Downstream Data Authority.

2.3.7.3.6 Air/Ground - Ground/Ground Distinction

2.3.7.3.6.1 Upon receipt of a CPDLC message the CPDLC-ground-user shall provide a distinction between CPDLC messages received from an aircraft and those received from another ground system.

2.3.7.3.7 Logical Acknowledgment Prohibited

2.3.7.3.7.1 Upon receipt of the CPDLC message *USE OF LOGICAL ACKNOWLEDGMENT PROHIBITED* the CPDLC-air-user shall be prohibited from requiring a logical acknowledgment for any message sent for the duration of the CPDLC or DSC dialogue.

2.3.7.3.7.2 If the CPDLC-ground-user receives a CPDLC message requiring a logical acknowledgment where the use of logical acknowledgment has been prohibited as above, the CPDLC-ground-user shall invoke the CPDLC-message service with a message containing the *ERROR* [errorinformation] message element with the [logicalAcknowledgmentNotAccepted] value as the CPDLC Message parameter and discard the content of the received message.

2.3.7.3.8 Message Reference Numbers

2.3.7.3.8.1 If a received message requires a response, the CPDLC-user shall provide a message reference number for each response message sent.

2.3.7.3.8.2 The message reference number shall be identical to the message identification number of the received message to which it refers.

2.3.7.3.9 Message Response Requirements

2.3.7.3.9.1 **Recommendation.** — *A message sequence initiated by data link should be closed by data link.*

2.3.7.3.9.2 **Recommendation.** — *If a message sequence exchange initiated by data link is subsequently closed by voice, local procedures should be in place to ensure deletion of outstanding data link messages requiring closure.*

2.3.7.3.9.3 A CPDLC-user shall only be permitted to respond to a received message in its entirety.

2.3.7.3.9.4 Only one closure response shall be permitted for a given message.

2.3.7.3.9.5 If the CPDLC-air-user has not issued a DSC-end service request primitive, or if the CPDLC-air-user has issued a DSC-end service request primitive for which a DSC-end service confirmation primitive has been received with the *Result* parameter containing the abstract value “rejected” then, if a message is received that requires a response, the CPDLC-air-user shall either:

- a) send any permitted response messages and then send a closure response message,
 or
- b) send a closure response message.

2.3.7.3.9.6 If the CPDLC-ground-user has not issued a CPDLC-end service request primitive, or if the CPDLC-ground-user has issued a CPDLC-end service request primitive for which a CPDLC-end service confirmation primitive has been received with the *Result* parameter containing the abstract value “rejected” then, if a message is received that requires a response, the CPDLC-ground-user shall either:

- a) send any permitted response messages and then send a closure response message,
 or
- b) send a closure response message.

Note.— *In above cases, the LOGICAL ACKNOWLEDGMENT message element is not sent.*

2.3.7.3.9.7 For a given message, once the CPDLC-user has sent the closure response message, no other response messages shall be sent referring to the given message.

2.3.7.3.9.8 When a message is received by the CPDLC-user requiring a logical acknowledgment response,

- a) when the CPDLC-user is a CPDLC-air-user, the CPDLC-air-user shall respond with either a CPDLC message containing a LOGICAL ACKNOWLEDGMENT message element, or with a message containing an ERROR, NOT CURRENT DATA AUTHORITY, or NOT AUTHORIZED NEXT DATA AUTHORITY message element as appropriate, or
- b) when the CPDLC-user is a CPDLC-ground-user, the CPDLC-ground-user shall respond with a CPDLC message containing a LOGICAL ACKNOWLEDGMENT message element, or with a message containing an ERROR, SERVICE UNAVAILABLE, or FLIGHT PLAN NOT HELD message element as appropriate.

2.3.7.3.9.9 A logical acknowledgment response message, if required, shall be sent prior to sending any other related response message(s), except:

- a) when the response is an uplink message, a response message containing an ERROR, SERVICE UNAVAILABLE, or FLIGHT PLAN NOT HELD message element as appropriate, or
- b) when the response is a downlink message, a response message containing an ERROR, NOT CURRENT DATA AUTHORITY, or NOT AUTHORIZED NEXT DATA AUTHORITY message element as appropriate.

Note.— When case a) or b) occurs, the LOGICAL ACKNOWLEDGEMENT message element is not sent.

2.3.7.3.9.10 When the CPDLC-air-user receives a message with a W/U RESP attribute, the only permitted responses shall be messages that contain a LOGICAL ACKNOWLEDGMENT (if required), STANDBY, WILCO, UNABLE, NOT CURRENT DATA AUTHORITY, NOT AUTHORIZED NEXT DATA AUTHORITY or ERROR message element.

2.3.7.3.9.11 When the CPDLC-air-user receives a message with a W/U RESP attribute, the closure response message shall contain at least a WILCO, UNABLE, NOT CURRENT DATA AUTHORITY, NOT AUTHORIZED NEXT DATA AUTHORITY or ERROR message element.

2.3.7.3.9.12 When the CPDLC-air-user receives a message with an A/N RESP attribute, the only permitted responses shall be messages that contain a LOGICAL ACKNOWLEDGMENT (if required), STANDBY, AFFIRM, NEGATIVE, NOT CURRENT DATA AUTHORITY, NOT AUTHORIZED NEXT DATA AUTHORITY or ERROR message element.

2.3.7.3.9.13 When the CPDLC-air-user receives a message with an A/N RESP attribute, the closure response message shall contain at least a AFFIRM, NEGATIVE, NOT CURRENT DATA AUTHORITY, NOT AUTHORIZED NEXT DATA AUTHORITY or ERROR message element.

2.3.7.3.9.14 When the CPDLC-air-user receives a message with a R RESP attribute, the only permitted responses shall be messages that contain a LOGICAL ACKNOWLEDGMENT (if required), STANDBY, ROGER, UNABLE, NOT CURRENT DATA AUTHORITY, NOT AUTHORIZED NEXT DATA AUTHORITY or ERROR message element.

2.3.7.3.9.15 When the CPDLC-air-user receives a message with a R RESP attribute, the closure response message shall contain at least a ROGER, UNABLE, NOT CURRENT DATA AUTHORITY, NOT AUTHORIZED NEXT DATA AUTHORITY or ERROR message element.

2.3.7.3.9.16 When the CPDLC-air-user receives a message with a Y RESP attribute, a LOGICAL ACKNOWLEDGMENT only when requested, and all other CPDLC messages shall be permitted as a response message.

2.3.7.3.9.17 When the CPDLC-air-user receives a message with a Y RESP attribute, the first response message sent that does not contain a STANDBY or LOGICAL ACKNOWLEDGMENT shall constitute the closure response message.

2.3.7.3.9.18 When the CPDLC-ground-user receives an air/ground message with a Y RESP attribute, a LOGICAL ACKNOWLEDGMENT, only when requested and all other CPDLC messages shall be permitted as a response message.

2.3.7.3.9.19 When the CPDLC-ground-user receives an air/ground message with a Y RESP attribute, the first response message sent that does not contain a STANDBY, REQUEST DEFERRED, or LOGICAL ACKNOWLEDGMENT message element shall constitute the closure response message.

2.3.7.3.9.20 When the CPDLC-air-user receives a message with a N RESP attribute, but requiring a logical acknowledgment, the only permitted response shall be a message that contains a LOGICAL ACKNOWLEDGMENT, NOT CURRENT DATA AUTHORITY, NOT AUTHORIZED NEXT DATA AUTHORITY or ERROR message element. This response message is the closure message.

2.3.7.3.9.21 When the CPDLC-ground-user receives an air/ground message with a N RESP attribute, but requiring a logical acknowledgment the only permitted response shall be a message that contains a LOGICAL ACKNOWLEDGMENT, SERVICE UNAVAILABLE, FLIGHT PLAN NOT HELD or ERROR message element. This response message is the closure message.

2.3.7.3.9.22 When the CPDLC-ground-user receives a ground/ground message the ground-user shall be prohibited from generating a ground/ground response message.

Note.— Ground/ground forwarding of messages is a one-way exchange on a one message per dialogue basis. There are no message identification or message reference numbers contained in the header of a ground/ground message.

2.3.7.3.10 Invalid Message Elements

2.3.7.3.10.1 The CPDLC-ground-user shall be prohibited from sending any CPDLC message containing the uplink message elements 33, 40, 41, or 178.

2.3.7.3.11 Error Conditions

2.3.7.3.11.1 Duplicate Message Identification Numbers

2.3.7.3.11.1.1 If a CPDLC message is received containing an identification number identical to that of an identification number currently in use the CPDLC-user shall invoke the CPDLC-user-abort request service with a CPDLC message containing the CPDLCUserAbortReason with value [duplicate-message-identification-number] as the *Reason* parameter.

2.3.7.3.11.2 Invalid Reference Number

2.3.7.3.11.2.1 If the CPDLC-user receives a message containing a message reference number which is not identical to any message identification number currently in use, the CPDLC-user shall:

- a) invoke CPDLC-message request with the ERROR [errorinformation] message element with the [unrecognizedMsgReferenceNumber] value as the *CPDLC Message* parameter, and
- b) disregard the received message.

2.3.7.3.11.3 No Available Message Identification Numbers

2.3.7.3.11.3.1 If the CPDLC-user attempts to send a CPDLC message and all message identification numbers are currently in use, the CPDLC-user shall invoke the CPDLC-user-abort request with a CPDLC message containing the CPDLCUserAbortReason with the value [no-message-identification-numbers-available] as the *Reason* parameter.

2.3.7.3.11.4 Insufficient Resources

2.3.7.3.11.4.1 If the CPDLC-user receives a message and has insufficient resources to handle the message, the CPDLC-user shall:

- a) invoke CPDLC-message request with the ERROR [errorinformation] message element with the [insufficientResources] value as the *CPDLC Message* parameter, and
- b) disregard the received message.

2.3.7.3.11.5 Invalid Message Element Combination

2.3.7.3.11.5.1 If a message is received containing:

- a) a LOGICAL ACKNOWLEDGMENT message element in combination with any other message element in a single message,
- b) a NEXT DATA AUTHORITY [facility] message element in combination with any other message element in a single message, or
- c) a message containing more than two message elements with the [routeClearance] variable,

the CPDLC-user shall invoke the CPDLC-message request with the ERROR [errorinformation] message element with the [invalidMessageElementCombination] value as the *CPDLC Message* parameter, and disregard the received message.

2.3.7.3.11.6 Invalid Message Elements

2.3.7.3.11.6.1 If the CPDLC-air-user receives a message containing any of the uplink message element identifiers 33, 40, 41 or 178 the CPDLC-air-user shall invoke the CPDLC-message request with the ERROR [errorinformation] message element with the [invalidMessageElement] value as the *CPDLC Message* parameter, and disregard the received message.

2.3.7.3.11.7 System Management Responses

2.3.7.3.11.7.1 If the CPDLC-air-user sends a message containing the NOT CURRENT DATA AUTHORITY, NOT AUTHORIZED NEXT DATA AUTHORITY or ERROR message element instead of the expected response message, the NOT CURRENT DATA AUTHORITY, NOT AUTHORIZED NEXT DATA AUTHORITY or ERROR message shall contain the received message identification number as the message reference number.

2.3.7.3.11.7.2 If the CPDLC-ground-user sends a message containing the SERVICE UNAVAILABLE, FLIGHT PLAN NOT HELD or ERROR message element instead of the expected response message, the SERVICE UNAVAILABLE, FLIGHT PLAN NOT HELD or ERROR message shall contain the received message identification number as the message reference number.

2.3.7.3.11.7.3 If the CPDLC-air-user sends a NOT CURRENT DATA AUTHORITY, NOT AUTHORIZED NEXT DATA AUTHORITY or ERROR message in response to a CPDLC message, the received CPDLC message shall be disregarded.

2.3.7.3.11.7.4 If the CPDLC-ground-user sends a SERVICE UNAVAILABLE, FLIGHT PLAN NOT HELD or ERROR message in response to a CPDLC message, the received CPDLC message shall be disregarded.

2.3.7.3.11.8 Invalid Message Response

2.3.7.3.11.8.1 If the CPDLC-ground-user sends a message that has a W/U response attribute, and a response to this message is received by the CPDLC-ground-user that does not contain any of the following message elements: WILCO, UNABLE, STANDBY, LOGICAL ACKNOWLEDGMENT, NOT CURRENT DATA AUTHORITY, NOT AUTHORIZED NEXT DATA AUTHORITY or ERROR [errorInformation] the CPDLC-ground-user shall invoke CPDLC-user-abort request with a CPDLC message containing the CPDLCUserAbortReason with the value [invalid-response].

2.3.7.3.11.8.2 If the CPDLC-ground-user sends a message that has an A/N response attribute, and a response to this message is received by the CPDLC-ground-user that does not contain any of the following message elements: AFFIRM, NEGATIVE, STANDBY, LOGICAL ACKNOWLEDGMENT, NOT CURRENT DATA AUTHORITY, NOT AUTHORIZED NEXT DATA AUTHORITY or ERROR [errorInformation] the CPDLC-ground-user shall invoke CPDLC-user-abort request with a CPDLC message containing the CPDLCUserAbortReason with the value [invalid-response].

2.3.7.3.11.8.3 If the CPDLC-ground-user sends a message that has a R response attribute, and a response to this message is received by the CPDLC-ground-user that does not contain any of the following message elements: ROGER, UNABLE, STANDBY, LOGICAL ACKNOWLEDGMENT, NOT CURRENT DATA AUTHORITY, NOT AUTHORIZED NEXT DATA AUTHORITY or ERROR [errorInformation] the CPDLC-ground-user shall invoke CPDLC-user-abort request with a CPDLC message containing the CPDLCUserAbortReason with the value [invalid-response].

2.3.7.3.11.8.4 If the CPDLC-ground-user sends a message that has a N response attribute and requires a logical acknowledgment, and a response to this message is received by the CPDLC-ground-user that does not contain any of the following message elements: LOGICAL ACKNOWLEDGMENT, NOT CURRENT DATA AUTHORITY, NOT AUTHORIZED NEXT DATA AUTHORITY or ERROR [errorInformation] the CPDLC-ground-user shall invoke CPDLC-user-abort request with a CPDLC message containing the CPDLCUserAbortReason with the value [invalid-response].

2.3.7.3.11.8.5 If the CPDLC-air-user sends a message that has a N response attribute and requires a logical acknowledgment, and a response to this message is received by the CPDLC-air-user that does not contain any of the following message elements: LOGICAL ACKNOWLEDGMENT, SERVICE UNAVAILABLE, FLIGHT PLAN NOT HELD or ERROR [errorInformation] the CPDLC-air-user shall invoke CPDLC-user-abort request with a CPDLC message containing the CPDLCUserAbortReason with the value [invalid-response].

2.3.7.4 CPDLC-air-user Requirements

2.3.7.4.1 The CPDLC-start Service

2.3.7.4.1.1 Invoking the CPDLC-start request

2.3.7.4.1.1.1 If there is no CPDLC service, the only CPDLC service primitives the CPDLC-air-user shall be permitted to invoke are the CPDLC-start request or the DSC-start request.

2.3.7.4.1.1.2 The CPDLC-air-user shall only be permitted to invoke the CPDLC-start request with:

- a) any ground system, if there is no existing CPDLC service for the CPDLC-air-user, or
- b) the Next Data Authority, if the CPDLC-air-user has received a message from the Current Data Authority designating a Next Data Authority.

2.3.7.4.1.1.3 If a CPDLC-air-user has invoked a CPDLC-start request, the CPDLC-air-user shall be prohibited from invoking any CPDLC-service primitive on the CPDLC dialogue initiated by the CPDLC-start service request, except the CPDLC-user-abort request, until after it has received a CPDLC-start confirmation.

2.3.7.4.1.2 Receipt of a CPDLC-start Indication and Invoking CPDLC-start Response

2.3.7.4.1.2.1 Upon receipt of a CPDLC-start service indication, the CPDLC-air-user shall invoke a CPDLC-start service response within 0.5 seconds.

2.3.7.4.1.2.2 Upon receipt of a CPDLC-start indication the CPDLC-air-user shall invoke the CPDLC-start response, with the response parameters set as follows:

- a) The *Result* parameter to the abstract value of “accepted” if:
 - 1) There is no existing CPDLC service, or
 - 2) CPDLC service exists and the request is from either the Current Data Authority or Next Data Authority,
- b) Else set the *Result* parameter is set to the abstract value “rejected” and the *Reject Reason* parameter to a CPDLC message with the message element NOT AUTHORIZED NEXT DATA AUTHORITY.

2.3.7.4.1.2.3 If a CPDLC-start indication is received from either the Current Data Authority or the Next Data Authority, and this results in a second CPDLC dialogue being established with a given ground system, the CPDLC-air-user shall invoke the CPDLC-user-abort request primitive for the first connection with that ground system.

2.3.7.4.1.2.4 If the CPDLC-air-user sets the CPDLC-start response *Result* parameter to the abstract value “rejected” any CPDLC message contained in the CPDLC-start indication *CPDLC Message* parameter shall be disregarded.

2.3.7.4.1.2.5 If the CPDLC-air-user sets the CPDLC-start response *Result* parameter to the abstract value “accepted” and the request is from the Current Data Authority any CPDLC message contained in the CPDLC-start indication *CPDLC Message* parameter shall be processed.

2.3.7.4.1.2.6 If the CPDLC-air-user sets the CPDLC-start response *Result* parameter to the abstract value “accepted” and the request is from the Next Data Authority any CPDLC message contained in the CPDLC-start indication *CPDLC Message* parameter shall be disregarded.

2.3.7.4.1.2.7 If the CPDLC-air-user sets the CPDLC-start response *Result* parameter to the abstract value “accepted” the CPDLC-air-user shall:

- a) Establish an association between a CPDLC-ASE invocation and a ground system facility designation contained in CPDLC-start indication *Calling Peer Identifier* parameter,
- b) If there is no Current Data Authority, associate this CPDLC-ASE invocation with the Current Data Authority, or
- c) If the facility designation contained in the CPDLC-start indication *Calling Peer Identifier* parameter is the Next Data Authority associate the CPDLC-ASE invocation with the Next Data Authority.

2.3.7.4.1.2.8 If a CPDLC-start indication has been received, the CPDLC-air-user shall be prohibited from invoking any CPDLC-service primitive with this ground system, except the CPDLC-user-abort request, until after it has invoked the CPDLC-start response.

2.3.7.4.1.3 Receipt of a CPDLC-start confirmation

2.3.7.4.1.3.1 If a CPDLC-start confirmation has been received with a *Result* parameter containing the abstract value “accepted” the CPDLC-air-user shall:

- a) Establish an association between a CPDLC-ASE invocation and the ground system facility designation contained in the CPDLC-start request *Called Peer Identifier* parameter,
- b) If there is no Current Data Authority, associate the CPDLC-ASE invocation with the Current Data Authority, or
- c) If the facility designation contained in the CPDLC-start request *Called Peer Identifier* parameter is the Next Data Authority associate the CPDLC-ASE invocation with the Next Data Authority.

2.3.7.4.2 The DSC-start Service

2.3.7.4.2.1 Invoking the DSC-start request

2.3.7.4.2.1.1 Only a CPDLC-air-user shall be permitted to invoke the DSC-start service request primitive.

2.3.7.4.2.1.2 A CPDLC-air-user shall only be permitted to invoke the DSC-start-service request primitive if the CPDLC-air-user has no existing DSC dialogue.

2.3.7.4.2.1.3 If a CPDLC-air-user has invoked a DSC-start request, the CPDLC-air-user shall be prohibited from invoking any CPDLC-service primitive on the DSC dialogue initiated by the DSC-start service request except the CPDLC-user-abort request, until after it has received a DSC-start confirmation.

2.3.7.4.2.2 Receipt of a DSC-start confirmation

2.3.7.4.2.2.1 If a DSC-start confirmation has been received with a *Result* parameter containing the abstract value “accepted” the CPDLC-air-user shall:

- a) Establish an association between a CPDLC-ASE invocation and the ground system facility designation contained in the DSC-start request *Facility Designation* parameter,
- b) Associate the CPDLC-ASE invocation with a Downstream Data Authority.

2.3.7.4.3 The CPDLC-message Service

2.3.7.4.3.1 Receipt of a CPDLC-message Indication

2.3.7.4.3.1.1 Upon receipt of a CPDLC-message indication, if the indication is from the Current Data Authority or a Downstream Data Authority the CPDLC-air-user shall process the CPDLC message contained in the *CPDLC Message* parameter.

2.3.7.4.3.1.2 If a CPDLC-message indication is received from the Current Data Authority containing only the uplink message element NEXT DATA AUTHORITY specifying a facility as the Next Data Authority the CPDLC-air-user shall do the following in the order listed:

- a) Check that there is no other Next Data Authority already established;
- b) if there is, invoke CPDLC-user-abort request with the established Next Data Authority with the *Reason* parameter set to CPDLCUserAbortReason value [no-longer-next-data-authority]; and
- c) then designate the ground system indicated in the CPDLC message from the CPDLC-message indication as the Next Data Authority.

2.3.7.4.3.1.3 If a CPDLC-message indication is received from the Current Data Authority containing only the uplink message element NEXT DATA AUTHORITY, indicating a “NULL” for the Next Data Authority the CPDLC-air-user shall do the following in the order listed:

- a) check if there is a Next Data Authority already established;
- b) if there is, invoke CPDLC-user-abort request with the established Next Data Authority with the *Reason* parameter set to CPDLCUserAbortReason value [no-longer-next-data-authority]; and

- c) cancel any existing Next Data Authority designation.

2.3.7.4.3.1.4 If a CPDLC-message indication is received containing the uplink message element NEXT DATA AUTHORITY, and it is not from the Current Data Authority the message shall be disregarded.

2.3.7.4.3.1.5 Upon receipt of a CPDLC-message indication, if the indication is not from the Current Data Authority or a Downstream Data Authority, the CPDLC-air-user shall:

- a) Invoke CPDLC-message service request with the *CPDLC Message* parameter containing a message with the message element NOT CURRENT DATA AUTHORITY, and
- b) Disregard the received message contained in the CPDLC-message indication *CPDLC Message* parameter.

2.3.7.4.4 The CPDLC-end Service

2.3.7.4.4.1 The CPDLC-end Service Request

2.3.7.4.4.1.1 The CPDLC-air-user shall be prohibited from invoking the CPDLC-end request.

2.3.7.4.4.2 Receipt of a CPDLC-end Indication and Invoking a CPDLC-end Response

2.3.7.4.4.2.1 If a CPDLC-end indication is received but it is not from the Current Data Authority the CPDLC-air-user shall:

- a) Invoke CPDLC-end response with
 - 1) the *CPDLC Message* parameter containing a message with the message element NOT CURRENT DATA AUTHORITY, and
 - 2) the *Result* parameter set to the abstract value “rejected”, and
- b) Disregard any message provided in the CPDLC-end indication *CPDLC Message* parameter.

Note 1.— A CPDLC-air-user is considered to have uplink open messages when the CPDLC-air-user has received a message(s) for which a response is required, and it has not yet sent the closure response to the message(s).

Note 2.— Uplink open messages are not considered in setting out the requirements for the CPDLC-end Service. The ground user is aware of any such messages, and desires to end the dialogue anyway. Any such messages are considered deleted upon transmission of a CPDLC-end response with the Result parameter set to the abstract value “accepted” on the airborne side, and upon receipt of a CPDLC-end confirmation with the Result parameter set to the abstract value “accepted” on the ground side.

Note 3.— A CPDLC-air-user is considered to have downlink open messages when the CPDLC-air-user has sent any messages that require a response for which it has not received a closure response.

Note 4.— Downlink open message are considered deleted upon transmission of a CPDLC-end response with the Result parameter set to the abstract value “accepted” on the airborne side, and upon receipt of a CPDLC-end confirmation with the Result parameter set to the abstract value “accepted” on the ground side.

Note 5.— Local procedures may dictate when a “rejected” response Result parameter is permitted by the CPDLC-air-user in the cases when section 2.3.7 gives the CPDLC-air-user a choice between “accepted” or “rejected”.

Note 6.— If a CPDLC-end service response is invoked with a “accepted” Result this will result in ending the dialogue regardless of any messages contained in the CPDLC Message parameter.

2.3.7.4.4.2.2 Uplink Message Contains Error

2.3.7.4.4.2.2.1 If a CPDLC-end indication is received from the Current Data Authority and there is a message in the *CPDLC Message* parameter that either has the response attribute N and requires a logical acknowledgment or has a W/U, A/N, R, or Y response attribute, and an error is detected in the message, then the CPDLC-air-user shall invoke CPDLC-end response with:

- a) the *CPDLC Message* parameter containing a CPDLC message with the message element ERROR [errorInformation],
- b) the *Result* parameter set to the abstract value “rejected”.

2.3.7.4.4.2.3 No Message in the CPDLC-end Indication

2.3.7.4.4.2.3.1 If a CPDLC-end indication is received from the Current Data Authority and the CPDLC-air-user does not have any downlink open messages and there is no message in the *CPDLC Message* parameter, then the CPDLC-air-user shall invoke CPDLC-end response with the *Result* parameter set to the abstract value “accepted”.

2.3.7.4.4.2.3.2 If a CPDLC-end indication is received from the Current Data Authority and the CPDLC-air-user has downlink open messages and there is no message in the *CPDLC Message* parameter, then the CPDLC-air-user shall invoke CPDLC-end response with the *Result* parameter set to the abstract value “accepted” or “rejected”.

2.3.7.4.4.2.4 Message in CPDLC-end Indication Does Not Require Response

2.3.7.4.4.2.4.1 If a CPDLC-end indication is received from the Current Data Authority and the CPDLC-air-user does not have any downlink open messages and there is a message in the *CPDLC Message* parameter with the response attribute N and not requiring a logical acknowledgment, then the CPDLC-air-user shall invoke CPDLC-end response with the *Result* parameter set to the abstract value “accepted”.

2.3.7.4.4.2.4.2 If a CPDLC-end indication is received from the Current Data Authority and the CPDLC-air-user has downlink open messages and there is a message in the *CPDLC Message* parameter with the response attribute N and not requiring a logical acknowledgment, then the CPDLC-air-user shall invoke CPDLC-end response with the *Result* parameter set to the abstract value “accepted” or “rejected”.

2.3.7.4.4.2.5 Message in CPDLC-end Indication Requires Only Logical Acknowledgment

2.3.7.4.4.2.5.1 If a CPDLC-end indication is received from the Current Data Authority and the CPDLC-air-user does not have any downlink open messages and there is a message in the *CPDLC Message* parameter with the response attribute N and requiring a logical acknowledgment, and no error is detected in the message, then the CPDLC-air-user shall invoke CPDLC-end response with:

- a) the *CPDLC Message* parameter containing a CPDLC message with the message element LOGICAL ACKNOWLEDGMENT, and
- b) the *Result* parameter set to the abstract value “accepted”.

2.3.7.4.4.2.5.2 If a CPDLC-end indication is received from the Current Data Authority and the CPDLC-air-user has downlink open messages and there is a message in the *CPDLC Message* parameter with the response attribute N and requiring a logical acknowledgment, and no error is detected in the message, then the CPDLC-air-user shall invoke CPDLC-end response with:

- a) the *CPDLC Message* parameter containing a CPDLC message with the message element LOGICAL ACKNOWLEDGMENT, and
- b) the *Result* parameter set to the abstract value “accepted” or “rejected”.

2.3.7.4.4.2.6 Message in CPDLC-end Indication With W/U, A/N, or R Attribute, Positive Response

Note.— In this case, positive response to a message in the CPDLC-end Indication also indicates acceptance of the end of the dialogue.

2.3.7.4.4.2.6.1 If a CPDLC-end indication is received from the Current Data Authority and there is a message in the *CPDLC Message* parameter with the response attribute W/U, A/N or R and no error is detected in the message, then if the CPDLC-air-user chooses to respond positively (WILCO, AFFIRM, or ROGER) to the message, then the CPDLC-air-user shall:

- a) if a logical acknowledgment is required, invoke CPDLC-message request with the *CPDLC Message* parameter containing a CPDLC message with only the message element LOGICAL ACKNOWLEDGMENT,
- b) if a STANDBY response is used, invoke CPDLC-message request with the *CPDLC Message* parameter containing a CPDLC message with at least the message element STANDBY, and
- c) invoke CPDLC-end response with:
 - 1) the *CPDLC Message* parameter containing a CPDLC message with at least the message element WILCO, AFFIRM, or ROGER as appropriate, and
 - 2) the *Result* parameter set to the abstract value “accepted”.

2.3.7.4.4.2.7 Message in CPDLC-end Indication With W/U, A/N, or R Attribute, Negative Response

Note.— In this case, negative response to a message in the CPDLC-end Indication also indicates rejection of the end of the dialogue.

2.3.7.4.4.2.7.1 If a CPDLC-end indication is received from the Current Data Authority and there is a message in the *CPDLC Message* parameter with the response attribute W/U, A/N or R and no error is detected in the message, then if the CPDLC-air-user chooses to respond negatively (UNABLE or NEGATIVE) to the message, then the CPDLC-air-user shall:

- a) if a logical acknowledgment is required, invoke CPDLC-message request with the *CPDLC Message* parameter containing a CPDLC message with only the message element LOGICAL ACKNOWLEDGMENT,
- b) if a STANDBY response is used, invoke CPDLC-message request with the *CPDLC Message* parameter containing a CPDLC message with at least the message element STANDBY, and
- c) invoke CPDLC-end response with:
 - 1) the *CPDLC Message* parameter containing a CPDLC message with at least the message element UNABLE or NEGATIVE as appropriate, and
 - 2) the *Result* parameter set to the abstract value “rejected”.

2.3.7.4.4.2.8 Message in CPDLC-end Indication With Y Response Attribute

2.3.7.4.4.2.8.1 If a CPDLC-end indication is received from the Current Data Authority and the CPDLC-air-user does not have any downlink open messages and there is a message in the *CPDLC Message* parameter with the response attribute Y and no error is detected in the message, the CPDLC-air-user shall:

- a) if a logical acknowledgment is required, invoke CPDLC-message request with the *CPDLC Message* parameter containing a CPDLC message with only the message element LOGICAL ACKNOWLEDGMENT,
- b) if a STANDBY response is used, invoke CPDLC-message request with the *CPDLC Message* parameter containing a CPDLC message with at least the message element STANDBY, and
- c) invoke CPDLC-end response with:
 - 1) the *CPDLC Message* parameter containing a Y attribute closure CPDLC message, and
 - 2) the *Result* parameter set to the abstract value “accepted”.

2.3.7.4.4.2.8.2 If a CPDLC-end indication is received from the Current Data Authority and the CPDLC-air-user has any downlink open messages and there is a message in the *CPDLC Message* parameter with the response attribute Y and no error is detected in the message, the CPDLC-air-user shall:

- a) if a logical acknowledgment is required, invoke CPDLC-message request with the *CPDLC Message* parameter containing a CPDLC message with only the message element LOGICAL ACKNOWLEDGMENT,
- b) if a STANDBY response is used, invoke CPDLC-message request with the *CPDLC Message* parameter containing a CPDLC message with at least the message element STANDBY, and
- c) invoke CPDLC-end response with:
 - 1) the *CPDLC Message* parameter containing a Y attribute closure CPDLC message, and
 - 2) the *Result* parameter set to the abstract value “accepted” or “rejected”.

2.3.7.4.4.2.9 Upon invoking a CPDLC-end response with *Result* parameter set to “accepted”, the CPDLC-air-user shall:

- a) delete any association with a ground system and Current Data Authority, and
- b) if a ground system is designated as Next Data Authority and an association with a CPDLC-ASE exists, replace the Next Data Authority association with a Current Data Authority association, or
- c) if a ground system is designated as Next Data Authority and no association with a CPDLC-ASE exists, delete Next Data Authority association with any ground system.

2.3.7.4.4.2.10 If the CPDLC-air-ASE associated with the Current Data Authority ceases to exist for any reason other than in response to a CPDLC-end request as specified above, any existing Next Data Authority designation and/or association shall cease to exist.

2.3.7.4.5 The DSC-end Service

2.3.7.4.5.1 The DSC-end Request

2.3.7.4.5.1.1 Only the CPDLC-air-user shall be permitted to invoke the DSC-end request.

2.3.7.4.5.1.2 If a CPDLC-air-user has invoked a DSC-end service request primitive, the CPDLC-air-user shall be prohibited from invoking any CPDLC service primitive with this ground system (except the CPDLC-user-abort request primitive) until it receives a DSC-end service confirmation primitive.

2.3.7.4.6 The CPDLC-user-abort Service

2.3.7.4.6.1 Issuing a CPDLC-user-abort Request [commanded-termination]

2.3.7.4.6.1.1 The CPDLC-air-user shall have the capability to invoke CPDLC-user-abort request with the *Reason* parameter set to CPDLCUserAbortReason value [commanded-termination].

2.3.7.4.6.2 Invoking a CPDLC-user-abort Request

2.3.7.4.6.2.1 If the CPDLC-air-user invokes CPDLC-user-abort request with the Current Data Authority, the CPDLC-air-user shall:

- a) Delete any association of a ground system to a Current Data Authority,
- b) If a ground system is designated as Next Data Authority and an association with a CPDLC-ASE exists, invoke CPDLC-user-abort request with the *Reason* parameter set to the value [current-data-authority-abort] and
- c) Delete any association of a ground system to a Next Data Authority.

2.3.7.4.6.2.2 If the CPDLC-air-user invokes CPDLC-user-abort request with the Next Data Authority, and then does not set the *Reason* parameter as [current-data-authority-abort] or [no-longer-next-data-authority], the CPDLC-air-user shall continue to maintain the association of the ground system to the Next Data Authority.

2.3.7.4.6.3 Receipt of a CPDLC-abort Indication

2.3.7.4.6.3.1 If the CPDLC-air-user receives a CPDLC-user-abort indication from the Current Data Authority or a CPDLC-provider-abort indication that causes the ASE invocation associated with the Current Data Authority to cease to exist, the CPDLC-air-user shall:

- a) Delete any association of a ground system to a Current Data Authority,
- b) If a ground system is designated as Next Data Authority and an association with a CPDLC-ASE exists, invoke CPDLC-user-abort request with the *Reason* parameter set to the value [current-data-authority-abort], and
- c) Delete any association of a ground system to a Next Data Authority.

2.3.7.4.6.3.2 If the CPDLC-air-user receives a CPDLC-user-abort indication from the Next Data Authority or receives a CPDLC-provider-abort indication that causes the ASE invocation associated with the Next Data Authority to cease to exist, the CPDLC-air-user shall continue to maintain the association of the ground system to the Next Data Authority.

2.3.7.5 CPDLC-Ground-User Requirements

2.3.7.5.1 The CPDLC-start Service

2.3.7.5.1.1 Invoking the CPDLC-start request

2.3.7.5.1.1.1 If there is no CPDLC service, the only CPDLC service primitives the CPDLC-ground-user shall be permitted to invoke are the CPDLC-start-request or the CPDLC-forward request.

2.3.7.5.1.1.2 If a CPDLC-ground-user has invoked a CPDLC-start request, the CPDLC-ground-user shall be prohibited from invoking any CPDLC-service primitive on the CPDLC dialogue initiated by the CPDLC-start service request, except the CPDLC-user-abort request until after it has received a CPDLC-start confirmation.

2.3.7.5.1.2 Receipt of a CPDLC-start Indication and Invoking CPDLC-start Response

2.3.7.5.1.2.1 If a CPDLC-start indication is received from an aircraft with which the ground system currently has a CPDLC dialogue, the CPDLC-ground-user shall:

- a) invoke the CPDLC-start response with the *Result* parameter set to the abstract value “accepted”, and
- b) invoke the CPDLC-user-abort request for the first CPDLC dialogue with that aircraft.

2.3.7.5.1.2.2 The CPDLC-ground-user shall be prohibited from invoking the CPDLC-start response unless and until it has received a CPDLC-start indication.

2.3.7.5.1.2.3 If the CPDLC-ground-user sets the CPDLC-start response *Result* parameter to the abstract value “rejected” then the *Reject Reason* parameter shall be an uplink message containing either the SERVICE UNAVAILABLE or FLIGHT PLAN NOT HELD message element as appropriate.

2.3.7.5.1.2.4 If the CPDLC-ground-user sets the CPDLC-start response *Result* parameter to the abstract value “rejected” the CPDLC-ground-user shall disregard any CPDLC message contained in the CPDLC-start indication *CPDLC Message* parameter.

2.3.7.5.1.2.5 If the CPDLC-ground-user sets the CPDLC-start response *Result* parameter to the abstract value “accepted” any CPDLC message contained in the CPDLC-start indication *CPDLC Message* parameter shall be processed.

2.3.7.5.1.2.6 If the CPDLC-ground-user sets the CPDLC-start response *Result* parameter to the abstract value “accepted” the CPDLC-ground-user shall establish an association between a CPDLC-ASE invocation and a 24 bit aircraft address contained in CPDLC-start indication *Calling Peer Identifier* parameter.

2.3.7.5.1.2.7 If a CPDLC-start indication has been received, the CPDLC-ground-user shall be prohibited from invoking any CPDLC-service primitive, except the CPDLC-user-abort request with that aircraft, until after it has invoked the CPDLC-start response.

2.3.7.5.1.2.8 Upon receipt of a CPDLC-start service indication, the CPDLC-ground-user shall invoke a CPDLC-start service response within 0.5 seconds.

2.3.7.5.1.3 Receipt of a CPDLC-start confirmation

2.3.7.5.1.3.1 If a CPDLC-start confirmation has been received with a *Result* parameter containing the abstract value “accepted” the CPDLC-ground-user shall establish an association between a CPDLC-ASE invocation and a 24 bit aircraft address contained in CPDLC-start request *Called Peer Identifier* parameter.

2.3.7.5.2 The DSC-start Service

2.3.7.5.2.1 Receipt of a DSC-start Indication and Invoking DSC-start Response

2.3.7.5.2.1.1 The CPDLC-ground-user shall be prohibited from invoking the DSC-start response unless and until it has received a DSC-start indication.

2.3.7.5.2.1.2 If a DSC-start indication is received from an aircraft with which the ground system currently has a DSC dialogue, the CPDLC-ground-user shall:

- a) invoke the DSC-start response with the *Result* parameter set to the abstract value “accepted”, and
- b) invoke the CPDLC-user-abort request for the first DSC dialogue with that aircraft.

2.3.7.5.2.1.3 If the CPDLC-ground-user sets the DSC-start response *Result* parameter to the abstract value “rejected” then the *Reject Reason* parameter shall be an uplink message containing either the SERVICE UNAVAILABLE or FLIGHT PLAN NOT HELD message element as appropriate.

2.3.7.5.2.1.4 If the CPDLC-ground-user sets the DSC-start response *Result* parameter to the abstract value “rejected” the CPDLC-ground-user shall disregard any CPDLC message contained in the DSC-start indication *CPDLC Message* parameter.

2.3.7.5.2.1.5 If the CPDLC-ground-user sets the DSC-start response *Result* parameter to the abstract value “accepted” any CPDLC message contained in the DSC-start indication *CPDLC Message* parameter shall be processed.

2.3.7.5.2.1.6 If the CPDLC-ground-user sets the DSC-start response *Result* parameter to the abstract value “accepted” the CPDLC-ground-user shall establish an association between a CPDLC-ASE invocation and a 24 bit aircraft address contained in the DSC-start indication *Aircraft Address* parameter.

2.3.7.5.2.1.7 If DSC-start indication has been received, the CPDLC-ground-user shall be prohibited from invoking any CPDLC-service primitive, except the CPDLC-user-abort request, until after it has invoked the DSC-start response.

2.3.7.5.2.2 Receipt of a DSC-start Indication and Invoking a DSC-start Response

2.3.7.5.2.2.1 Upon receipt of a DSC-start indication, the CPDLC-ground-user shall invoke a DSC-start response within 0.5 seconds.

2.3.7.5.2.2.2 Upon receipt of a DSC-start indication, the CPDLC-ground-user shall invoke a DSC-start response, with the response parameters set as follows:

- a) the *Result* parameter set to the abstract value “accepted” or “rejected”, and
- b) if, and only if, the *Result* parameter is set to the abstract value “rejected”, then set the *Reject Reason* parameter to a CPDLC message with the message element SERVICE UNAVAILABLE or FLIGHT PLAN NOT HELD as appropriate.

2.3.7.5.3 The CPDLC-end Service

2.3.7.5.3.1 The CPDLC-end Request

2.3.7.5.3.1.1 Only the CPDLC-ground-user shall be permitted to invoke the CPDLC-end request.

2.3.7.5.3.1.2 If a CPDLC-ground-user has invoked a CPDLC-end service request primitive, the CPDLC-ground-user shall be prohibited from invoking any CPDLC service primitive with this aircraft, except the CPDLC-user-abort request primitive, until after it has received a CPDLC-end service confirmation primitive.

2.3.7.5.4 The DSC-end Service

2.3.7.5.4.1 Receipt of a DSC-end Indication and Invoking DSC-end Response

Note 1.— The CPDLC-ground-user is considered to have downlink open messages when the CPDLC-ground-user has received a message(s) for which a response is required, and it has not yet sent the closure response to the message(s).

Note 2.— Downlink open messages are not considered in setting out the requirements for the DSC-end Service. The air user is aware of any such messages, and desires to end the dialogue anyway. Any such messages are considered deleted upon transmission of a DSC-end response with the Result parameter set to the abstract value “accepted” on the ground side, and upon receipt of a DSC-end confirmation with the Result parameter set to the abstract value “accepted” on the airborne side.

Note 3.— The CPDLC-ground-user is considered to have uplink open messages when the CPDLC-ground-user has sent any messages that require a response for which it has not received a closure response.

Note 4.— Uplink open message are considered deleted upon transmission of a DSC-end response with the Result parameter set to the abstract value “accepted” on the ground side, and upon receipt of a DSC-end confirmation with the Result parameter set to the abstract value “accepted” on the airborne side.

Note 5.— Local procedures may dictate when a “rejected” response Result parameter is permitted by the CPDLC-ground-user in the cases when section 2.3.7 gives the CPDLC-ground-user a choice between “accepted” or “rejected”.

Note 6.— If a DSC-end service response is invoked with a “accepted” Result this will result in ending the dialogue regardless of any messages contained in the CPDLC Message parameter.

2.3.7.5.4.1.1 Downlink Message Contains Error

2.3.7.5.4.1.1.1 If a DSC-end indication is received and there is a message in the *CPDLC Message* parameter that either has the response attribute N and requires a logical acknowledgment or has Y response attribute, and an error is detected in the message, then the CPDLC-ground-user shall invoke DSC-end response with:

- a) the *CPDLC Message* parameter containing a CPDLC message with the message element ERROR [errorInformation],
- b) the *Result* parameter set to the abstract value “rejected”.

2.3.7.5.4.1.2 No Message in the DSC-end Indication

2.3.7.5.4.1.2.1 If a DSC-end indication is received and the CPDLC-ground-user does not have any uplink open messages and there is no message in the *CPDLC Message* parameter, then the CPDLC-ground-user shall invoke DSC-end response with the *Result* parameter set to the abstract value “accepted”.

2.3.7.5.4.1.2.2 If a DSC-end indication is received and the CPDLC-ground-user has uplink open messages and there is no message in the *CPDLC Message* parameter, then the CPDLC-ground-user shall invoke DSC-end response with the *Result* parameter set to the abstract value “accepted” or “rejected”.

2.3.7.5.4.1.3 Message in DSC-end Indication Does Not Require Response

2.3.7.5.4.1.3.1 If a DSC-end indication is received and the CPDLC-ground-user does not have any uplink open messages and there is a message in the *CPDLC Message* parameter with the response attribute N and not requiring a logical acknowledgment, then the CPDLC-ground-user shall invoke DSC-end response with the *Result* parameter set to the abstract value “accepted”.

2.3.7.5.4.1.3.2 If a DSC-end indication is received and the CPDLC-ground-user has uplink open messages and there is a message in the *CPDLC Message* parameter with the response attribute N and not requiring a logical acknowledgment, then the CPDLC-ground-user shall invoke DSC-end response with the *Result* parameter set to the abstract value “accepted” or “rejected”.

2.3.7.5.4.1.4 Message in DSC-end Indication Requires Only Logical Acknowledgment

2.3.7.5.4.1.4.1 If a DSC-end indication is received and the CPDLC-ground-user does not have any uplink open messages and there is a message in the *CPDLC Message* parameter with the response attribute N and requiring a logical acknowledgment, and no error is detected in the message, then the CPDLC-ground-user shall invoke DSC-end response with:

- a) the *CPDLC Message* parameter containing a CPDLC message with the message element LOGICAL ACKNOWLEDGMENT, and
- b) the *Result* parameter set to the abstract value “accepted”.

2.3.7.5.4.1.4.2 If a DSC-end indication is received and the CPDLC-ground-user has uplink open messages and there is a message in the *CPDLC Message* parameter with the response attribute N and requiring a logical acknowledgment, and no error is detected in the message, then the CPDLC-ground-user shall invoke DSC-end response with:

- a) the *CPDLC Message* parameter containing a CPDLC message with the message element LOGICAL ACKNOWLEDGMENT, and
- b) the *Result* parameter set to the abstract value “accepted” or “rejected”.

2.3.7.5.4.1.5 Message in DSC-end Indication With Y Response Attribute

2.3.7.5.4.1.5.1 If a DSC-end indication is received and the CPDLC-ground-user does not have any uplink open messages and there is a message in the *CPDLC Message* parameter with the response attribute Y and no error is detected in the message the CPDLC-ground-user shall:

- a) if a logical acknowledgment is required, invoke CPDLC-message request with the *CPDLC Message* parameter containing a CPDLC message with only the message element LOGICAL ACKNOWLEDGMENT,
- b) if a STANDBY response is used, invoke CPDLC-message request with the *CPDLC Message* parameter containing a CPDLC message with at least the message element STANDBY,
- c) if a REQUEST DEFERRED response is used, invoke CPDLC-message request with the *CPDLC Message* parameter containing a CPDLC message with at least the message element REQUEST DEFERRED, and
- d) invoke DSC-end response with:
 - 1) the *CPDLC Message* parameter containing a Y attribute closure CPDLC message, and
 - 2) the *Result* parameter set to the abstract value “accepted”.

2.3.7.5.4.1.5.2 If a DSC-end indication is received and the CPDLC-ground-user has uplink open messages and there is a message in the *CPDLC Message* parameter with the response attribute Y and no error is detected in the message the CPDLC-ground-user shall:

- a) if a logical acknowledgment is required, invoke CPDLC-message request with the *CPDLC Message* parameter containing a CPDLC message with only the message element LOGICAL ACKNOWLEDGMENT,
- b) if a STANDBY response is used, invoke CPDLC-message request with the *CPDLC Message* parameter containing a CPDLC message with at least the message element STANDBY, and
- c) if a REQUEST DEFERRED response is used, invoke CPDLC-message request with the *CPDLC Message* parameter containing a CPDLC message with at least the message element REQUEST DEFERRED, and
- d) invoke DSC-end response with:
 - 1) the *CPDLC Message* parameter containing a Y attribute closure CPDLC message, and
 - 2) the *Result* parameter set to the abstract value “accepted” or “rejected”.

2.3.7.5.5 The CPDLC-forward Service

2.3.7.5.5.1 Invoking the CPDLC-forward Request

2.3.7.5.5.1.1 Only the CPDLC-ground-user shall be permitted to invoke the CPDLC-forward Request.

2.3.7.5.6 The CPDLC-user-abort Service

2.3.7.5.6.1 Issuing a CPDLC-user-abort Request

2.3.7.5.6.1.1 The CPDLC-ground-user shall have the capability to invoke CPDLC-user-abort request with the *Reason* parameter set to CPDLCUserAbortReason value [commanded-termination].

2.3.7.6 Message Intent

2.3.7.6.1 Purpose

Note 1.— 2.3.7.6 contains the message set for CPDLC. Message attributes, message presentation guidance, and data structure presentation guidance are presented. The actual information exchanged between an aircraft and ground peer or a ground and ground peer CPDLC applications is defined in 2.3.4; however, 2.3.4 does not mandate any particular method for presenting this information. The presentation of information to the controller and aircraft crew is a local implementation. The message presentation recommendations contained in Tables 2.3.7-5 to 2.3.7-28 are one possible means of presenting the information. These recommendations are generally consistent with current ICAO practices for displaying ATC information.

Note 2.— Tables 2.3.7-5 to 2.3.7-28 which are aligned with the provisions contained in Appendix 5 of the 13th Edition (including Amendment 2, dated 5 November 1998) of ICAO Doc 4444 — Procedures for Air Navigation Services — Rules of the Air and Air Traffic Services (PANS-RAC) are included in this document for completeness.

2.3.7.6.2 Uplink message elements shall comply with the intent, use, and element attributes as presented in Tables 2.3.7-5 to 2.3.7-16.

Table 2.3.7-5. Responses/Acknowledgments (uplink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
0	Indicates that ATC cannot comply with the request.	UNABLE	N	M	N
1	Indicates that ATC has received the message and will respond.	STANDBY	N	L	N
2	Indicates that ATC has received the request but it has been deferred until later.	REQUEST DEFERRED	N	L	N
3	Indicates that ATC has received and understood the message.	ROGER	N	L	N
4	Yes.	AFFIRM	N	L	N

	Message Intent/Use	Message Element	URG	ALRT	RESP
5	No.	NEGATIVE	N	L	N
235	Notification of receipt of unlawful interference message.	ROGER 7500	U	H	N
211	Indicates that the ATC has received the request and has passed it to the next control authority.	REQUEST FORWARDED	N	L	N
218	Indicates to the pilot that the request has already been received on the ground.	REQUEST ALREADY RECEIVED	L	N	N

Table 2.3.7-6. Vertical Clearances (uplink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
6	Notification that a level change instruction should be expected.	EXPECT [level]	L	L	R
7	Notification that an instruction should be expected for the aircraft to commence climb at the specified time.	EXPECT CLIMB AT [time]	L	L	R
8	Notification that an instruction should be expected for the aircraft to commence climb at the specified position.	EXPECT CLIMB AT [position]	L	L	R
9	Notification that an instruction should be expected for the aircraft to commence descent at the specified time.	EXPECT DESCENT AT [time]	L	L	R
10	Notification that an instruction should be expected for the aircraft to commence descent at the specified position.	EXPECT DESCENT AT [position]	L	L	R
11	Notification that an instruction should be expected for the aircraft to commence cruise climb at the specified time.	EXPECT CRUISE CLIMB AT [time]	L	L	R

	Message Intent/Use	Message Element	URG	ALRT	RESP
12	Notification that an instruction should be expected for the aircraft to commence cruise climb at the specified position.	EXPECT CRUISE CLIMB AT [position]	L	L	R
13	Notification that an instruction should be expected for the aircraft to commence climb at the specified time to the specified level.	AT [time] EXPECT CLIMB TO [level]	L	L	R
14	Notification that an instruction should be expected for the aircraft to commence climb at the specified position to the specified level.	AT [position] EXPECT CLIMB TO [level]	L	L	R
15	Notification that an instruction should be expected for the aircraft to commence descent at the specified time to the specified level.	AT [time] EXPECT DESCENT TO [level]	L	L	R
16	Notification that an instruction should be expected for the aircraft to commence descent at the specified position to the specified level.	AT [position] EXPECT DESCENT TO [level]	L	L	R
17	Notification that an instruction should be expected for the aircraft to commence cruise climb at the specified time to the specified level.	AT [time] EXPECT CRUISE CLIMB TO [level]	L	L	R
18	Notification that an instruction should be expected for the aircraft to commence cruise climb at the specified position to the specified level.	AT [position] EXPECT CRUISE CLIMB TO [level]	L	L	R
19	Instruction to maintain the specified level.	MAINTAIN [level]	N	M	W/U
20	Instruction that a climb to a specified level is to commence and once reached the specified level is to be maintained.	CLIMB TO [level]	N	M	W/U

	Message Intent/Use	Message Element	URG	ALRT	RESP
21	Instruction that at the specified time a climb to the specified level is to commence and once reached the specified level is to be maintained.	AT [time] CLIMB TO [level]	N	M	W/U
22	Instruction that at the specified position a climb to the specified level is to commence and once reached the specified level is to be maintained.	AT [position] CLIMB TO [level]	N	M	W/U
185	Instruction that after passing the specified position a climb to the specified level is to commence and once reached the specified level is to be maintained.	AFTER PASSING [position] CLIMB TO [level]	N	M	W/U
23	Instruction that a descent to a specified level is to commence and once reached the specified level is to be maintained.	DESCEND TO [level]	N	M	W/U
24	Instruction that at a specified time a descent to a specified level is to commence and once reached the specified level is to be maintained.	AT [time] DESCEND TO [level]	N	M	W/U
25	Instruction that at the specified position a descent to the specified level is to commence and once reached the specified level is to be maintained.	AT [position] DESCEND TO [level]	N	M	W/U
186	Instruction that after passing the specified position a descent to the specified level is to commence and once reached the specified level is to be maintained.	AFTER PASSING [position] DESCEND TO [level]	N	M	W/U
26	Instruction that a climb is to commence at a rate such that the specified level is reached at or before the specified time.	CLIMB TO REACH [level] BY [time]	N	M	W/U

	Message Intent/Use	Message Element	URG	ALRT	RESP
27	Instruction that a climb is to commence at a rate such that the specified level is reached at or before the specified position.	CLIMB TO REACH [level] BY [position]	N	M	W/U
28	Instruction that a descent is to commence at a rate such that the specified level is reached at or before the specified time.	DESCEND TO REACH [level] BY [time]	N	M	W/U
29	Instruction that a descent is to commence at a rate such that the specified level is reached at or before the specified position.	DESCEND TO REACH [level] BY [position]	N	M	W/U
192	Instruction that a change of level is to continue, but at a rate such that the specified level is reached at or before the specified time.	REACH [level] BY [time]	N	M	W/U
209	Instruction that a change of level is to continue, but at a rate such that the specified level is reached at or before the specified position.	REACH [level] BY [position]	N	M	W/U
30	Instruction that a level within the defined vertical range specified is to be maintained.	MAINTAIN BLOCK [level] TO [level]	N	M	W/U
31	Instruction that a climb to a level within the vertical range defined is to commence.	CLIMB TO AND MAINTAIN BLOCK [level] TO [level]	N	M	W/U
32	Instruction that a descent to a level within the vertical range defined is to commence.	DESCEND TO AND MAINTAIN BLOCK [level] TO [level]	N	M	W/U
34	Instruction that a cruise climb is to commence and continue until the specified level is reached.	CRUISE CLIMB TO [level]	N	M	W/U
35	Instruction that a cruise climb can commence once above the specified level.	CRUISE CLIMB ABOVE [level]	N	M	W/U
219	Instruction to stop the climb below the previously assigned level.	STOP CLIMB AT [level]	U	M	W/U

	Message Intent/Use	Message Element	URG	ALRT	RESP
220	Instruction to stop the descent above the previously assigned level.	STOP DESCENT AT [level]	U	M	W/U
36	Instruction that the climb to the specified level should be made at the aircraft's best rate.	EXPEDITE CLIMB TO [level]	U	M	W/U
37	Instruction that the descent to the specified level should be made at the aircraft's best rate.	EXPEDITE DESCENT TO [level]	U	M	W/U
38	Urgent instruction to immediately climb to the specified level.	IMMEDIATELY CLIMB TO [level]	D	H	W/U
39	Urgent instruction to immediately descend to the specified level.	IMMEDIATELY DESCEND TO [level]	D	H	W/U
40	(reserved)		L	L	Y
41	(reserved)		L	L	Y
171	Instruction to climb at not less than the specified rate.	CLIMB AT [vertical rate] MINIMUM	N	M	W/U
172	Instruction to climb at not above the specified rate.	CLIMB AT [vertical rate] MAXIMUM	N	M	W/U
173	Instruction to descend at not less than the specified rate.	DESCEND AT [vertical rate] MINIMUM	N	M	W/U
174	Instruction to descend at not above the specified rate.	DESCEND AT [vertical rate] MAXIMUM	N	M	W/U
33	(reserved)		L	L	Y

Note.— Wherever the variable “level” is specified, the message can specify either a single level or a vertical range, i.e. block level.

Table 2.3.7-7. Crossing Constraints (uplink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
42	Notification that a level change instruction should be expected which will require the specified position to be crossed at the specified level.	EXPECT TO CROSS [position] AT [level]	L	L	R
43	Notification that a level change instruction should be expected which will require the specified position to be crossed at or above the specified level.	EXPECT TO CROSS [position] AT OR ABOVE [level]	L	L	R
44	Notification that a level change instruction should be expected which will require the specified position to be crossed at or below the specified level.	EXPECT TO CROSS [position] AT OR BELOW [level]	L	L	R
45	Notification that a level change instruction should be expected which will require the specified position to be crossed at the specified level which is to be maintained subsequently.	EXPECT TO CROSS [position] AT AND MAINTAIN [level]	L	L	R
46	Instruction that the specified position is to be crossed at the specified level. This may require the aircraft to modify its climb or descent profile.	CROSS [position] AT [level]	N	M	W/U
47	Instruction that the specified position is to be crossed at or above the specified level.	CROSS [position] AT OR ABOVE [level]	N	M	W/U
48	Instruction that the specified position is to be crossed at or below the specified level.	CROSS [position] AT OR BELOW [level]	N	M	W/U
49	Instruction that the specified position is to be crossed at the specified level and that level is to be maintained when reached.	CROSS [position] AT AND MAINTAIN [level]	N	M	W/U
50	Instruction that the specified position is to be crossed at a level between the specified levels.	CROSS [position] BETWEEN [level] AND [level]	N	M	W/U

	Message Intent/Use	Message Element	URG	ALRT	RESP
51	Instruction that the specified position is to be crossed at the specified time.	CROSS [position] AT [time]	N	M	W/U
52	Instruction that the specified position is to be crossed at or before the specified time.	CROSS [position] AT OR BEFORE [time]	N	M	W/U
53	Instruction that the specified position is to be crossed at or after the specified time.	CROSS [position] AT OR AFTER [time]	N	M	W/U
54	Instruction that the specified position is to be crossed at a time between the specified times.	CROSS [position] BETWEEN [time] AND [time]	N	M	W/U
55	Instruction that the specified position is to be crossed at the specified speed and the specified speed is to be maintained until further advised.	CROSS [position] AT [speed]	N	M	W/U
56	Instruction that the specified position is to be crossed at a speed equal to or less than the specified speed and the specified speed or less is to be maintained until further advised.	CROSS [position] AT OR LESS THAN [speed]	N	M	W/U
57	Instruction that the specified position is to be crossed at a speed equal to or greater than the specified speed and the specified speed or greater is to be maintained until further advised.	CROSS [position] AT OR GREATER THAN [speed]	N	M	W/U
58	Instruction that the specified position is to be crossed at the specified time and the specified level.	CROSS [position] AT [time] AT [level]	N	M	W/U
59	Instruction that the specified position is to be crossed at or before the specified time and at the specified level.	CROSS [position] AT OR BEFORE [time] AT [level]	N	M	W/U
60	Instruction that the specified position is to be crossed at or after the specified time and at the specified level.	CROSS [position] AT OR AFTER [time] AT [level]	N	M	W/U

	Message Intent/Use	Message Element	URG	ALRT	RESP
61	Instruction that the specified position is to be crossed at the specified level and speed, and the level and speed are to be maintained.	CROSS [position] AT AND MAINTAIN [level] AT [speed]	N	M	W/U
62	Instruction that at the specified time the specified position is to be crossed at the specified level and the level is to be maintained.	AT [time] CROSS [position] AT AND MAINTAIN [level]	N	M	W/U
63	Instruction that at the specified time the specified position is to be crossed at the specified level and speed, and the level and speed are to be maintained.	AT [time] CROSS [position] AT AND MAINTAIN [level] AT [speed]	N	M	W/U

Note.— Wherever the variable “level” is specified, the message can specify either a single level or a vertical range, i.e. block level.

Table 2.3.7-8. Lateral Offsets (uplink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
64	Instruction to fly a parallel track to the cleared route at a displacement of the specified distance in the specified direction.	OFFSET [specified distance] [direction] OF ROUTE	N	M	W/U
65	Instruction to fly a parallel track to the cleared route at a displacement of the specified distance in the specified direction and commencing at the specified position.	AT [position] OFFSET [specified distance] [direction] OF ROUTE	N	M	W/U
66	Instruction to fly a parallel track to the cleared route at a displacement of the specified distance in the specified direction and commencing at the specified time.	AT [time] OFFSET [specified distance] [direction] OF ROUTE	N	M	W/U
67	Instruction that the cleared flight route is to be rejoined.	PROCEED BACK ON ROUTE	N	M	W/U
68	Instruction that the cleared flight route is to be rejoined at or before the specified position.	REJOIN ROUTE BY [position]	N	M	W/U

	Message Intent/Use	Message Element	URG	ALRT	RESP
69	Instruction that the cleared flight route is to be rejoined at or before the specified time.	REJOIN ROUTE BY [time]	N	M	W/U
70	Notification that a clearance may be issued to enable the aircraft to rejoin the cleared route at or before the specified position.	EXPECT BACK ON ROUTE BY [position]	L	L	R
71	Notification that a clearance may be issued to enable the aircraft to rejoin the cleared route at or before the specified time.	EXPECT BACK ON ROUTE BY [time]	L	L	R
72	Instruction to resume own navigation following a period of tracking or heading clearances. May be used in conjunction with an instruction on how or where to rejoin the cleared route.	RESUME OWN NAVIGATION	N	M	W/U

Table 2.3.7-9. Route Modifications (uplink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
73	Notification to the aircraft of the instructions to be followed from departure until the specified clearance limit.	[departure clearance]	N	M	W/U
74	Instruction to proceed directly from its present position to the specified position.	PROCEED DIRECT TO [position]	N	M	W/U
75	Instruction to proceed, when able, directly to the specified position.	WHEN ABLE PROCEED DIRECT TO [position]	N	M	W/U
76	Instruction to proceed, at the specified time, directly to the specified position.	AT [time] PROCEED DIRECT TO [position]	N	M	W/U
77	Instruction to proceed, at the specified position, directly to the next specified position.	AT [position] PROCEED DIRECT TO [position]	N	M	W/U
78	Instruction to proceed, upon reaching the specified level, directly to the specified position.	AT [level] PROCEED DIRECT TO [position]	N	M	W/U

	Message Intent/Use	Message Element	URG	ALRT	RESP
79	Instruction to proceed to the specified position via the specified route.	CLEARED TO [position] VIA [route clearance]	N	M	W/U
80	Instruction to proceed via the specified route.	CLEARED [route clearance]	N	M	W/U
81	Instruction to proceed in accordance with the specified procedure.	CLEARED [procedure name]	N	M	W/U
236	Instruction to leave controlled airspace.	LEAVE CONTROLLED AIRSPACE	N	M	W/U
82	Approval to deviate up to the specified distance from the cleared route in the specified direction.	CLEARED TO DEVIATE UP TO [specified distance] [direction] OF ROUTE	N	M	W/U
83	Instruction to proceed from the specified position via the specified route.	AT [position] CLEARED [route clearance]	N	M	W/U
84	Instruction to proceed from the specified position via the specified procedure.	AT [position] CLEARED [procedure name]	N	M	W/U
85	Notification that a clearance to fly on the specified route may be issued.	EXPECT [route clearance]	L	L	R
86	Notification that a clearance to fly on the specified route from the specified position may be issued.	AT [position] EXPECT [route clearance]	L	L	R
87	Notification that a clearance to fly directly to the specified position may be issued.	EXPECT DIRECT TO [position]	L	L	R
88	Notification that a clearance to fly directly from the first specified position to the next specified position may be issued.	AT [position] EXPECT DIRECT TO [position]	L	L	R
89	Notification that a clearance to fly directly to the specified position commencing at the specified time may be issued.	AT [time] EXPECT DIRECT TO [position]	L	L	R

	Message Intent/Use	Message Element	URG	ALRT	RESP
90	Notification that a clearance to fly directly to the specified position commencing when the specified level is reached may be issued.	AT [level] EXPECT DIRECT TO [position]	L	L	R
91	Instruction to enter a holding pattern with the specified characteristics at the specified position and level.	HOLD AT [position] MAINTAIN [level] INBOUND TRACK [degrees] [direction] TURNS [leg type]	N	M	W/U
92	Instruction to enter a holding pattern with the published characteristics at the specified position and level.	HOLD AT [position] AS PUBLISHED MAINTAIN [level]	N	M	W/U
93	Notification that an onwards clearance may be issued at the specified time.	EXPECT FURTHER CLEARANCE AT [time]	L	L	R
94	Instruction to turn left or right as specified on to the specified heading.	TURN [direction] HEADING [degrees]	N	M	W/U
95	Instruction to turn left or right as specified on to the specified track.	TURN [direction] GROUND TRACK [degrees]	N	M	W/U
215	Instruction to turn a specified number of degrees left or right.	TURN [direction] [degrees]	N	M	W/U
190	Instruction to fly on the specified heading.	FLY HEADING [degrees]	N	M	W/U
96	Instruction to continue to fly on the current heading.	CONTINUE PRESENT HEADING	N	M	W/U
97	Instruction to fly on the specified heading from the specified position.	AT [position] FLY HEADING [degrees]	N	M	W/U
221	Instruction to stop turn at the specified heading prior to reaching the previously assigned heading.	STOP TURN HEADING [degrees]	U	M	W/U
98	Instruction to turn immediately left or right as specified on to the specified heading.	IMMEDIATELY TURN [direction] HEADING [degrees]	D	H	W/U
99	Notification that a clearance may be issued for the aircraft to fly the specified procedure.	EXPECT [procedure name]	L	L	R

Note.— Wherever the variable “level” is specified, the message can specify either a single level or a vertical range, i.e. block level.

Table 2.3.7-10. Speed Changes (uplink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
100	Notification that a speed instruction may be issued to be effective at the specified time.	AT [time] EXPECT [speed]	L	L	R
101	Notification that a speed instruction may be issued to be effective at the specified position.	AT [position] EXPECT [speed]	L	L	R
102	Notification that a speed instruction may be issued to be effective at the specified level.	AT [level] EXPECT [speed]	L	L	R
103	Notification that a speed range instruction may be issued to be effective at the specified time.	AT [time] EXPECT [speed] TO [speed]	L	L	R
104	Notification that a speed range instruction may be issued to be effective at the specified position.	AT [position] EXPECT [speed] TO [speed]	L	L	R
105	Notification that a speed range instruction may be issued to be effective at the specified level.	AT [level] EXPECT [speed] TO [speed]	L	L	R
106	Instruction that the specified speed is to be maintained.	MAINTAIN [speed]	N	M	W/U
188	Instruction that after passing the specified position the specified speed is to be maintained.	AFTER PASSING [position] MAINTAIN [speed]	N	M	W/U
107	Instruction that the present speed is to be maintained.	MAINTAIN PRESENT SPEED	N	M	W/U
108	Instruction that the specified speed or a greater speed is to be maintained.	MAINTAIN [speed] OR GREATER	N	M	W/U
109	Instruction that the specified speed or a lesser speed is to be maintained.	MAINTAIN [speed] OR LESS	N	M	W/U
110	Instruction that a speed within the specified range is to be maintained.	MAINTAIN [speed] TO [speed]	N	M	W/U

	Message Intent/Use	Message Element	URG	ALRT	RESP
111	Instruction that the present speed is to be increased to the specified speed and maintained until further advised.	INCREASE SPEED TO [speed]	N	M	W/U
112	Instruction that the present speed is to be increased to the specified speed or greater, and maintained at or above the specified speed until further advised.	INCREASE SPEED TO [speed] OR GREATER	N	M	W/U
113	Instruction that the present speed is to be reduced to the specified speed and maintained until further advised.	REDUCE SPEED TO [speed]	N	M	W/U
114	Instruction that the present speed is to be reduced to the specified speed or less and maintained at or below the specified speed until further advised.	REDUCE SPEED TO [speed] OR LESS	N	M	W/U
115	Instruction that the specified speed is not to be exceeded.	DO NOT EXCEED [speed]	N	M	W/U
116	Notification that the aircraft need no longer comply with the previously issued speed restriction.	RESUME NORMAL SPEED	N	M	W/U
189	Instruction that the present speed is to be changed to the specified speed.	ADJUST SPEED TO [speed]	N	M	W/U
222	Notification that the aircraft may keep its preferred speed without restriction.	NO SPEED RESTRICTION	L	L	R
223	Instruction to reduce present speed to the minimum safe approach speed.	REDUCE TO MINIMUM APPROACH SPEED	N	M	W/U

Note.— Wherever the variable “level” is specified, the message can specify either a single level or a vertical range, i.e. block level.

Table 2.3.7-11. Contact/Monitor/Surveillance Requests (uplink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
117	Instruction that the ATS unit with the specified ATS unit name is to be contacted on the specified frequency.	CONTACT [unitname] [frequency]	N	M	W/U
118	Instruction that at the specified position the ATS unit with the specified ATS unit name is to be contacted on the specified frequency.	AT [position] CONTACT [unitname] [frequency]	N	M	W/U
119	Instruction that at the specified time the ATS unit with the specified ATS unit name is to be contacted on the specified frequency.	AT [time] CONTACT [unitname] [frequency]	N	M	W/U
120	Instruction that the ATS unit with the specified ATS unit name is to be monitored on the specified frequency.	MONITOR [unitname] [frequency]	N	M	W/U
121	Instruction that at the specified position the ATS unit with the specified ATS unit name is to be monitored on the specified frequency.	AT [position] MONITOR [unitname] [frequency]	N	M	W/U
122	Instruction that at the specified time the ATS unit with the specified ATS unit name is to be monitored on the specified frequency.	AT [time] MONITOR [unitname] [frequency]	N	M	W/U
123	Instruction that the specified code (SSR code) is to be selected.	SQUAWK [code]	N	M	W/U
124	Instruction that the SSR transponder responses are to be disabled.	STOP SQUAWK	N	M	W/U
125	Instruction that the SSR transponder responses should include level information.	SQUAWK MODE CHARLIE	N	M	W/U
126	Instruction that the SSR transponder responses should no longer include level information.	STOP SQUAWK MODE CHARLIE	N	M	W/U

	Message Intent/Use	Message Element	URG	ALRT	RESP
179	Instruction that the 'ident' function on the SSR transponder is to be actuated.	SQUAWK IDENT	N	M	W/U

Table 2.3.7-12. Report/Confirmation Requests (uplink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
127	Instruction to report when the aircraft is back on the cleared route.	REPORT BACK ON ROUTE	N	L	W/U
128	Instruction to report when the aircraft has left the specified level.	REPORT LEAVING [level]	N	L	W/U
129	Instruction to report when the aircraft is in level flight at the specified level.	REPORT MAINTAINING [level]	N	L	W/U
175	Instruction to report when the aircraft has reached the specified level.	REPORT REACHING [level]	N	L	W/U
200	Instruction used in conjunction with a level clearance to report reaching the level assigned.	REPORT REACHING	N	M	W/U
180	Instruction to report when the aircraft is within the specified vertical range.	REPORT REACHING BLOCK [level] TO [level]	N	L	W/U
130	Instruction to report when the aircraft has passed the specified position.	REPORT PASSING [position]	N	L	W/U
181	Instruction to report the present distance to or from the specified position.	REPORT DISTANCE [to/from] [position]	N	M	Y
184	Instruction to report at the specified time the distance to or from the specified position.	AT TIME [time] REPORT DISTANCE [to/from] [position]	N	L	Y
228	Instruction to report the estimated time of arrival at the specified position.	REPORT ETA [position]	L	L	Y
131	Instruction to report the amount of fuel remaining and the number of persons on board.	REPORT REMAINING FUEL AND PERSONS ON BOARD	U	M	Y

	Message Intent/Use	Message Element	URG	ALRT	RESP
132	Instruction to report the present position.	REPORT POSITION	N	M	Y
133	Instruction to report the present level.	REPORT PRESENT LEVEL	N	M	Y
134	Instruction to report the requested speed.	REPORT [speed type] [speed type] [speed type] SPEED	N	M	Y
135	Instruction to confirm and acknowledge the currently assigned level.	CONFIRM ASSIGNED LEVEL	N	L	Y
136	Instruction to confirm and acknowledge the currently assigned speed.	CONFIRM ASSIGNED SPEED	N	L	Y
137	Instruction to confirm and acknowledge the currently assigned route.	CONFIRM ASSIGNED ROUTE	N	L	Y
138	Instruction to confirm the previously reported time over the last reported waypoint.	CONFIRM TIME OVER REPORTED WAYPOINT	N	L	Y
139	Instruction to confirm the identity of the previously reported waypoint.	CONFIRM REPORTED WAYPOINT	N	L	Y
140	Instruction to confirm the identity of the next waypoint.	CONFIRM NEXT WAYPOINT	N	L	Y
141	Instruction to confirm the previously reported estimated time at the next waypoint.	CONFIRM NEXT WAYPOINT ETA	N	L	Y
142	Instruction to confirm the identity of the next but one waypoint.	CONFIRM ENSUING WAYPOINT	N	L	Y
143	The request was not understood. It should be clarified and resubmitted.	CONFIRM REQUEST	N	L	Y
144	Instruction to report the selected (SSR) code.	CONFIRM SQUAWK	N	L	Y
145	Instruction to report the present heading.	REPORT HEADING	N	M	Y
146	Instruction to report the present ground track.	REPORT GROUND TRACK	N	M	Y

	Message Intent/Use	Message Element	URG	ALRT	RESP
182	Instruction to report the identification code of the last ATIS received.	CONFIRM ATIS CODE	N	L	Y
147	Instruction to make a position report.	REQUEST POSITION REPORT	N	M	Y
216	Instruction to file a flight plan.	REQUEST FLIGHT PLAN	N	M	Y
217	Instruction to report that the aircraft has landed.	REPORT ARRIVAL	N	M	Y
229	Instruction to report the preferred alternate aerodrome for landing.	REPORT ALTERNATE AERODROME	L	L	Y
231	Instruction to indicate the pilot's preferred level.	STATE PREFERRED LEVEL	L	L	Y
232	Instruction to indicate the pilot's preferred time and/or position to commence descent to the aerodrome of intended arrival.	STATE TOP OF DESCENT	L	L	Y

Note.— Wherever the variable “level” is specified, the message can specify either a single level or a vertical range, i.e. block level.

Table 2.3.7-13. Negotiation Requests (uplink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
148	Request for the earliest time at which the specified level can be accepted.	WHEN CAN YOU ACCEPT [level]	N	L	Y
149	Instruction to report whether or not the specified level can be accepted at the specified position.	CAN YOU ACCEPT [level] AT [position]	N	L	A/N
150	Instruction to report whether or not the specified level can be accepted at the specified time.	CAN YOU ACCEPT [level] AT [time]	N	L	A/N
151	Instruction to report the earliest time when the specified speed can be accepted.	WHEN CAN YOU ACCEPT [speed]	N	L	Y
152	Instruction to report the earliest time when the specified offset track can be accepted.	WHEN CAN YOU ACCEPT [specified distance] [direction] OFFSET	N	L	Y

Note.— Wherever the variable “level” is specified, the message can specify either a single level or a vertical range, i.e. block level.

Table 2.3.7-14. Air Traffic Advisories (uplink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
153	ATS advisory that the altimeter setting should be the specified setting.	ALTIMETER [altimeter]	N	L	R
213	ATS advisory that the specified altimeter setting relates to the specified facility.	[facility designation] ALTIMETER [altimeter]	N	L	R
154	ATS advisory that the radar service is terminated.	RADAR SERVICE TERMINATED	N	L	R
191	ATS advisory that the aircraft is entering airspace in which no air traffic services are provided and all existing air traffic services are terminated.	ALL ATS TERMINATED	N	M	R
155	ATS advisory that radar contact has been established at the specified position.	RADAR CONTACT [position]	N	M	R
156	ATS advisory that radar contact has been lost.	RADAR CONTACT LOST	N	M	R
210	ATS advisory that the aircraft has been identified on radar at the specified position.	IDENTIFIED [position]	N	M	R
193	Notification that radar identification has been lost.	IDENTIFICATION LOST	N	M	R
157	Instruction that a continuous transmission is detected on the specified frequency. Check the microphone button.	CHECK STUCK MICROPHONE [frequency]	U	M	N
158	ATS advisory that the ATIS information identified by the specified code is the current ATIS information.	ATIS [atis code]	N	L	R
212	ATS advisory that the specified ATIS information at the specified airport is current.	[facility designation] ATIS [atis code] CURRENT	N	L	R
214	ATS advisory that indicates the RVR value for the specified runway.	RVR RUNWAY [runway] [rvr]	N	M	R

	Message Intent/Use	Message Element	URG	ALRT	RESP
224	ATS advisory that no delay is expected.	NO DELAY EXPECTED	N	L	R
225	ATS advisory that the expected delay has not been determined.	DELAY NOT DETERMINED	N	L	R
226	ATS advisory that the aircraft may expect to be cleared to commence its approach procedure at the specified time.	EXPECTED APPROACH TIME [time]	N	L	R

Table 2.3.7-15. System Management Messages (uplink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
159	A system generated message notifying that the ground system has detected an error.	ERROR [error information]	U	M	N
160	Notification to the avionics that the specified data authority is the next data authority. If no data authority is specified, this indicates that any previously specified next data authority is no longer valid.	NEXT DATA AUTHORITY [facility]	L	N	N
161	Notification to the avionics that the data link connection with the current data authority is being terminated.	END SERVICE	L	N	N
162	Notification that the ground system does not support this message.	SERVICE UNAVAILABLE	L	L	N
234	Notification that the ground system does not have a flight plan for that aircraft.	FLIGHT PLAN NOT HELD	L	L	N
163	Notification to the pilot of an ATSU identifier.	[facility designation]	L	N	N
227	Confirmation to the aircraft system that the ground system has received the message to which the logical acknowledgment refers and found it acceptable for display to the responsible person.	LOGICAL ACKNOWLEDGMENT	N	M	N

	Message Intent/Use	Message Element	URG	ALRT	RESP
233	Notification to the pilot that messages sent requiring a logical acknowledgment will not be accepted by this ground system.	USE OF LOGICAL ACKNOWLEDGMENT PROHIBITED	N	M	N

Table 2.3.7-16. Additional Messages (uplink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
164	The associated instruction may be complied with at any future time.	WHEN READY	L	N	N
230	The associated instruction is to be complied with immediately.	IMMEDIATELY	D	H	N
165	Used to link two messages, indicating the proper order of execution of clearances/ instructions.	THEN	L	N	N
166	The associated instruction is issued due to traffic considerations.	DUE TO [traffic type] TRAFFIC	L	N	N
167	The associated instruction is issued due to airspace restrictions.	DUE TO AIRSPACE RESTRICTION	L	N	N
168	The indicated communication should be ignored.	DISREGARD	U	M	R
176	Notification that the pilot is responsible for maintaining separation from other traffic and is also responsible for maintaining visual meteorological conditions.	MAINTAIN OWN SEPARATION AND VMC	N	M	W/U
177	Used in conjunction with a clearance/instruction to indicate that the pilot may execute when prepared to do so.	AT PILOTS DISCRETION	L	L	N
178	(reserved)		L	L	Y
169		[free text]	N	L	R
170		[free text]	D	H	R
183		[free text]	N	M	N
187		[free text]	L	N	N
194		[free text]	N	L	Y

	Message Intent/Use	Message Element	URG	ALRT	RESP
195		[free text]	L	L	R
196		[free text]	N	M	W/U
197		[free text]	U	M	W/U
198		[free text]	D	H	W/U
199		[free text]	N	L	N
201	Not used		L	L	N
202	Not used		L	L	N
203		[free text]	N	M	R
204		[free text]	N	M	Y
205		[free text]	N	M	A/N
206		[free text]	L	N	Y
207		[free text]	L	L	Y
208		[free text]	L	L	N

Note.— Free text message elements have no associated message intent. The capability to send a free text message with any of the attribute combinations already used in the message set has been provided.

2.3.7.6.3 Downlink message elements shall comply with the intent, use, and element attributes as presented in Tables 2.3.7-17 to 2.3.7-28.

Table 2.3.7-17. Responses (downlink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
0	The instruction is understood and will be complied with.	WILCO	N	M	N
1	The instruction cannot be complied with.	UNABLE	N	M	N
2	Wait for a reply.	STANDBY	N	M	N
3	Message received and understood.	ROGER	N	M	N
4	Yes.	AFFIRM	N	M	N
5	No.	NEGATIVE	N	M	N

Table 2.3.7-18. Vertical Requests (downlink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
6	Request to fly at the specified level.	REQUEST [level]	N	L	Y
7	Request to fly at a level within the specified vertical range.	REQUEST BLOCK [level] TO [level]	N	L	Y
8	Request to cruise climb to the specified level.	REQUEST CRUISE CLIMB TO [level]	N	L	Y
9	Request to climb to the specified level.	REQUEST CLIMB TO [level]	N	L	Y
10	Request to descend to the specified level.	REQUEST DESCENT TO [level]	N	L	Y
11	Request that at the specified position a climb to the specified level be approved.	AT [position] REQUEST CLIMB TO [level]	N	L	Y
12	Request that at the specified position a descent to the specified level be approved.	AT [position] REQUEST DESCENT TO [level]	N	L	Y
13	Request that at the specified time a climb to the specified level be approved.	AT [time] REQUEST CLIMB TO [level]	N	L	Y
14	Request that at the specified time a descent to the specified level be approved.	AT [time] REQUEST DESCENT TO [level]	N	L	Y
69	Request that a descent be approved on a see-and-avoid basis.	REQUEST VMC DESCENT	N	L	Y

Note.— Wherever the variable “level” is specified, the message can specify either a single level or a vertical range, i.e. block level.

Table 2.3.7-19. Lateral Off-Set Requests (downlink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
15	Request that a parallel track, offset from the cleared track by the specified distance in the specified direction, be approved.	REQUEST OFFSET [specified distance] [direction] OF ROUTE	N	L	Y
16	Request that a parallel track, offset from the cleared track by the specified distance in the specified direction, be approved from the specified position.	AT [position] REQUEST OFFSET [specified distance] [direction] OF ROUTE	N	L	Y
17	Request that a parallel track, offset from the cleared track by the specified distance in the specified direction, be approved from the specified time.	AT [time] REQUEST OFFSET [specified distance] [direction] OF ROUTE	N	L	Y

Table 2.3.7-20. Speed Requests (downlink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
18	Request to fly at the specified speed.	REQUEST [speed]	N	L	Y
19	Request to fly within the specified speed range.	REQUEST [speed] TO [speed]	N	L	Y

Table 2.3.7-21. Voice Contact Requests (downlink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
20	Request for voice contact.	REQUEST VOICE CONTACT	N	L	Y
21	Request for voice contact on the specified frequency.	REQUEST VOICE CONTACT [frequency]	N	L	Y

Table 2.3.7-22. Route Modification Requests (downlink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
22	Request to track from the present position direct to the specified position.	REQUEST DIRECT TO [position]	N	L	Y
23	Request for the specified procedure clearance.	REQUEST [procedure name]	N	L	Y
24	Request for a route clearance.	REQUEST CLEARANCE [route clearance]	N	L	Y
25	Request for a clearance.	REQUEST [clearance type] CLEARANCE	N	L	Y
26	Request for a weather deviation to the specified position via the specified route.	REQUEST WEATHER DEVIATION TO [position] VIA [route clearance]	N	M	Y
27	Request for a weather deviation up to the specified distance off track in the specified direction.	REQUEST WEATHER DEVIATION UP TO [specified distance] [direction] OF ROUTE	N	M	Y
70	Request a clearance to adopt the specified heading.	REQUEST HEADING [degrees]	N	L	Y
71	Request a clearance to adopt the specified ground track.	REQUEST GROUND TRACK [degrees]	N	L	Y

Table 2.3.7-23. Reports (downlink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
28	Notification of leaving the specified level.	LEAVING [level]	N	L	N
29	Notification of climbing to the specified level.	CLIMBING TO [level]	N	L	N
30	Notification of descending to the specified level.	DESCENDING TO [level]	N	L	N
31	Notification of passing the specified position.	PASSING [position]	N	L	N
78	Notification that at the specified time, the aircraft's position was as specified.	AT [time] [distance] [to/from] [position]	N	L	N
32	Notification of the present level.	PRESENT LEVEL [level]	N	L	N

	Message Intent/Use	Message Element	URG	ALRT	RESP
33	Notification of the present position.	PRESENT POSITION [position]	N	L	N
34	Notification of the present speed.	PRESENT SPEED [speed]	N	L	N
113	Notification of the requested speed.	[speed type] [speed type] [speed type] SPEED [speed]	N	L	N
35	Notification of the present heading in degrees.	PRESENT HEADING [degrees]	N	L	N
36	Notification of the present ground track in degrees.	PRESENT GROUND TRACK [degrees]	N	L	N
37	Notification that the aircraft is maintaining the specified level.	MAINTAINING [level]	N	L	N
72	Notification that the aircraft has reached the specified level.	REACHING [level]	N	L	N
76	Notification that the aircraft has reached a level within the specified vertical range.	REACHING BLOCK [level] TO [level]	N	L	N
38	Read-back of the assigned level.	ASSIGNED LEVEL [level]	N	M	N
77	Read-back of the assigned vertical range.	ASSIGNED BLOCK [level] TO [level]	N	M	N
39	Read-back of the assigned speed.	ASSIGNED SPEED [speed]	N	M	N
40	Read-back of the assigned route.	ASSIGNED ROUTE [route clearance]	N	M	N
41	The aircraft has regained the cleared route.	BACK ON ROUTE	N	M	N
42	The next waypoint is the specified position.	NEXT WAYPOINT [position]	N	L	N
43	The ETA at the next waypoint is as specified.	NEXT WAYPOINT ETA [time]	N	L	N
44	The next but one waypoint is the specified position.	ENSUING WAYPOINT [position]	N	L	N
45	Clarification of previously reported waypoint passage.	REPORTED WAYPOINT [position]	N	L	N
46	Clarification of time over previously reported waypoint.	REPORTED WAYPOINT [time]	N	L	N

	Message Intent/Use	Message Element	URG	ALRT	RESP
47	The specified (SSR) code has been selected.	SQUAWKING [code]	N	L	N
48	Position report.	POSITION REPORT [position report]	N	M	N
79	The code of the latest ATIS received is as specified.	ATIS [atis code]	N	L	N
89	The specified ATS unit is being monitored on the specified frequency.	MONITORING [unitname] [frequency]	U	M	N
102	Used to report that an aircraft has landed.	LANDING REPORT	N	N	N
104	Notification of estimated time of arrival at the specified position.	ETA [position][time]	L	L	N
105	Notification of the alternative aerodrome for landing.	ALTERNATE AERODROME [airport]	L	L	N
106	Notification of the preferred level.	PREFERRED LEVEL [level]	L	L	N
109	Notification of the preferred time to commence descent for approach.	TOP OF DESCENT [time]	L	L	N
110	Notification of the preferred position to commence descent for approach.	TOP OF DESCENT [position]	L	L	N
111	Notification of the preferred time and position to commence descent for approach.	TOP OF DESCENT [time] [position]	L	L	N

Note.— Wherever the variable “level” is specified, the message can specify either a single level or a vertical range, i.e. block level.

Table 2.3.7-24. Negotiation Requests (downlink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
49	Request for the earliest time at which a clearance to the specified speed can be expected.	WHEN CAN WE EXPECT [speed]	L	L	Y

	Message Intent/Use	Message Element	URG	ALRT	RESP
50	Request for the earliest time at which a clearance to a speed within the specified range can be expected.	WHEN CAN WE EXPECT [speed] TO [speed]	L	L	Y
51	Request for the earliest time at which a clearance to regain the planned route can be expected.	WHEN CAN WE EXPECT BACK ON ROUTE	L	L	Y
52	Request for the earliest time at which a clearance to descend can be expected.	WHEN CAN WE EXPECT LOWER LEVEL	L	L	Y
53	Request for the earliest time at which a clearance to climb can be expected.	WHEN CAN WE EXPECT HIGHER LEVEL	L	L	Y
54	Request for the earliest time at which a clearance to cruise climb to the specified level can be expected.	WHEN CAN WE EXPECT CRUISE CLIMB TO [level]	L	L	Y
87	Request for the earliest time at which a clearance to climb to the specified level can be expected.	WHEN CAN WE EXPECT CLIMB TO [level]	L	L	Y
88	Request for the earliest time at which a clearance to descend to the specified level can be expected.	WHEN CAN WE EXPECT DESCENT TO [level]	L	L	Y

Note.— Wherever the variable “level” is specified, the message can specify either a single level or a vertical range, i.e. block level.

Table 2.3.7-25. Emergency Messages (downlink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
55	Urgency prefix.	PAN PAN PAN	U	H	Y
56	Distress prefix.	MAYDAY MAYDAY MAYDAY	D	H	Y
112	Indicates specifically that the aircraft is being subjected to unlawful interference.	SQUAWKING 7500	U	H	N

	Message Intent/Use	Message Element	URG	ALRT	RESP
57	Notification of fuel remaining and number of persons on board.	[remaining fuel] OF FUEL REMAINING AND [persons on board] PERSONS ON BOARD	U	H	Y
58	Notification that the pilot wishes to cancel the emergency condition.	CANCEL EMERGENCY	U	M	Y
59	Notification that the aircraft is diverting to the specified position via the specified route due to an urgent need.	DIVERTING TO [position] VIA [route clearance]	U	H	Y
60	Notification that the aircraft is deviating the specified distance in the specified direction off the cleared route and maintaining a parallel track due to an urgent need.	OFFSETTING [specified distance] [direction] OF ROUTE	U	H	Y
61	Notification that the aircraft is descending to the specified level due to an urgent need.	DESCENDING TO [level]	U	H	Y
80	Notification that the aircraft is deviating up to the deviating distance from the cleared route in the specified direction due to an urgent need.	DEVIATING UP TO [specified distance] [direction] OF ROUTE	U	H	Y

Note.— Wherever the variable “level” is specified, the message can specify either a single level or a vertical range, i.e. block level.

Table 2.3.7-26. System Management Messages (downlink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
62	A system-generated message that the avionics has detected an error.	ERROR [error information]	U	L	N
63	A system-generated denial to any CPDLC message sent from a ground facility that is not the current data authority.	NOT CURRENT DATA AUTHORITY	L	L	N
99	A system-generated message to inform a ground facility that it is now the current data authority	CURRENT DATA AUTHORITY	L	L	N

	Message Intent/Use	Message Element	URG	ALRT	RESP
64	Notification to the ground system that the specified ATSU is the current data authority.	[facility designation]	L	L	N
107	A system-generated message sent to a ground system that tries to connect to an aircraft when a current data authority has not designated the ground system as the NDA.	NOT AUTHORIZED NEXT DATA AUTHORITY	L	L	N
73	A system-generated message indicating the software version number.	[version number]	L	L	N
100	Confirmation to the ground system that the aircraft system has received the message to which the logical acknowledgment refers and found it acceptable for display to the responsible person.	LOGICAL ACKNOWLEDGMENT	N	M	N

Table 2.3.7-27. Additional Messages (downlink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
65	Used to explain reasons for pilot's message.	DUE TO WEATHER	L	L	N
66	Used to explain reasons for pilot's message.	DUE TO AIRCRAFT PERFORMANCE	L	L	N
74	States a desire by the pilot to provide his/her own separation and remain in VMC.	REQUEST TO MAINTAIN OWN SEPARATION AND VMC	L	L	Y
75	Used in conjunction with another message to indicate that the pilot wishes to execute request when the pilot is prepared to do so.	AT PILOTS DISCRETION	L	L	N
101	Allows the pilot to indicate a desire for termination of CPDLC service with the current data authority.	REQUEST END OF SERVICE	L	L	Y
103	Allows the pilot to indicate that he/she has cancelled IFR flight plan.	CANCELLING IFR	N	L	Y

	Message Intent/Use	Message Element	URG	ALRT	RESP
108	Notification that de-icing action has been completed.	DE-ICING COMPLETE	L	L	N
67		[free text]	N	L	N
68		[free text]	D	H	Y
90		[free text]	N	M	N
91		[free text]	N	L	Y
92		[free text]	L	L	Y
93		[free text]	U	H	N
94		[free text]	D	H	N
95		[free text]	U	M	N
96		[free text]	U	L	N
97		[free text]	L	L	N
98		[free text]	N	N	N

Note.— Free text message elements have no associated message intent. The capability to send a free text message with any of the attribute combinations already used in the message set has been provided.

Table 2.3.7-28. Negotiation Responses (downlink)

	Message Intent/Use	Message Element	URG	ALRT	RESP
81	We can accept the specified level at the specified time.	WE CAN ACCEPT [level] AT [time]	L	L	N
82	We cannot accept the specified level.	WE CANNOT ACCEPT [level]	L	L	N
83	We can accept the specified speed at the specified time.	WE CAN ACCEPT [speed] AT [time]	L	L	N
84	We cannot accept the specified speed.	WE CANNOT ACCEPT [speed]	L	L	N
85	We can accept a parallel track offset the specified distance in the specified direction at the specified time.	WE CAN ACCEPT [specified distance] [direction] at [time]	L	L	N

	Message Intent/Use	Message Element	URG	ALRT	RESP
86	We cannot accept a parallel track offset the specified distance in the specified direction.	WE CANNOT ACCEPT [specified distance] [direction]	L	L	N

Note.— Wherever the variable “level” is specified, the message can specify either a single level or a vertical range, i.e. block level.

2.3.7.7 Parameter Value Unit, Range, and Resolution

2.3.7.7.1 A CPDLC-user shall interpret CPDLC message element variables unit, range, and resolution as defined in 2.3.4.

2.3.8 SUBSETTING RULES

2.3.8.1 General

Note.— This chapter specifies conformance requirements which all implementations of the CPDLC protocol obey.

2.3.8.1.1 An implementation of either the CPDLC ground based service or the CPDLC air based service claiming conformance to 2.3 shall support the CPDLC protocol features as shown in the tables below:

Note.— The ‘status’ column indicates the level of support required for conformance to the CPDLC-ASE protocol described in this part. The values are as follows:

‘M’ mandatory support is required,

‘O’ optional support is permitted for conformance to the CPDLC protocol,

‘N/A’ the item is not applicable, and

‘C.n’ the item is conditional where *n* is the number which identifies the condition which is applicable. The definitions for the conditional statements used in this chapter are written under the tables in which they first appear.

Table 2.3. 8-1. Protocol Versions Implemented

	Status	Associated Predicate
Version 1	M	none

Table 2.3.8-2. CPDLC Protocol Options

	Status	Associated Predicate
CPDLC-air-ASE	C.1	CPDLC/air
CPDLC-ground-ASE	C.1	CPDLC/ground
DSC function supported	if (CPDLC/air) O, else M	DSC-FU
DSC function supported by CPDLC-ground-user	if (CPDLC/ground) O, else N/A	DSC-USER
Forward function supported by initiating user	if (CPDLC/ground) O, else N/A	FWD-INIT
Forward function supported by receiving user	if (CPDLC/ground) O, else N/A	FWD-USER

C.1: a conforming implementation will support one and only one of these two options.

Table 2.3.8-3. CPDLC-air-ASE Conformant Configurations

	List of Predicates	Functionality Description
I.	CPDLC/air	a CPDLC-air-ASE supporting just the core* CPDLC functionality (no downstream clearance capability)
II.	CPDLC/air + DSC-FU	a CPDLC-air-ASE supporting the core CPDLC functionality and the downstream clearance capability (complete CPDLC-air-ASE functionality)
		* the core CPDLC functionality is defined as support for the CPDLC-start, message, end services plus abort services.

Table 2.3.8-4. CPDLC-ground-ASE Conformant Configurations

	List of Predicates	Functionality Description
I.	CPDLC/ground	a CPDLC-ground-ASE supporting the core CPDLC functionality plus: • functionality for receiving and CPDLC-ground-user rejecting a request for a DSC dialogue • functionality for receiving and indicating that the forward function is not supported
II.	CPDLC/ground + DCS-FU + DSC-USER	a CPDLC-ground-ASE supporting the core CPDLC functionality and downstream clearance function plus • functionality for receiving and indicating that the ground forward function is not supported
III.	CPDLC/ground + DSC-FU + FWD-INIT	a CPDLC-ground-ASE supporting the core CPDLC functionality plus: • functionality for receiving and CPDLC-ground-user rejecting a request for a DSC dialogue • functionality for receiving and indicating that the ground forward function is not supported • functionality for supporting the capability to initiate the ground forwarding of CPDLC messages
IV.	CPDLC/ground + DCS-FU + DSC-USER + FWD-INIT	a CPDLC-ground-ASE supporting the core CPDLC functionality and downstream clearance function plus • functionality for receiving and indicating that the ground forward function is not supported • functionality for supporting the capability to initiate the ground forwarding of CPDLC messages

	List of Predicates	Functionality Description
V.	CPDLC/ground + DSC-FU FWD-USER	a CPDLC-ground-ASE supporting the core CPDLC functionality plus: • functionality for receiving and CPDLC-ground-user rejecting a request for DSC dialogue.
VI.	CPDLC/ground + DCS-FU + DSC-USER + FWD-USER	a CPDLC-ground-ASE supporting the core CPDLC functionality and downstream clearance function plus • functionality for CPDLC-ground-user support for the receipt of CPDLC ground forward messages
VII.	CPDLC/ground + DSC-FU + FWD-INIT + FWD-USER	a CPDLC-ground-ASE supporting the core CPDLC functionality plus: • functionality for receiving and CPDLC-ground-user rejecting a request for a DSC dialogue • full CPDLC ground forwarding functionality (initiating and user receiving)
VIII.	CPDLC/ground + DSC-FU + DSC-USER + FWD-INIT + FWD-USER	a CPDLC-ground-ASE supporting the core CPDLC functionality and downstream clearance function plus • full CPDLC ground forwarding functionality (initiating and user receiving) (complete CPDLC-ground-ASE functionality)

2.4 FLIGHT INFORMATION SERVICES APPLICATION

2.4.1 INTRODUCTION

2.4.1.1 Introduction

2.4.1.1.1 The FIS application allows a pilot to request and receive FIS services from ground FIS systems. The FIS application is designed to enable FIS services to be provided to a pilot via the exchange of messages between aircraft avionics and ground FIS systems.

Note.— Structure:

- a) 2.4.1: *INTRODUCTION* contains the part's purpose and structure and a summary of the functions of FIS.
- b) 2.4.2: *GENERAL REQUIREMENTS* contains backwards compatibility and error processing requirements.
- c) 2.4.3: *ABSTRACT SERVICE* contains the description of the abstract service provided by the application service elements (ASE) defined for FIS.
- d) 2.4.4: *FORMAL DEFINITION OF MESSAGES* contains the formal definition of the messages exchanged by FIS-ASEs using the Abstract Syntax Notation Number One (ASN.1).
- e) 2.4.5: *PROTOCOL DEFINITION* describes the exchanges of messages allowed by the FIS protocol, as well as time constraints and the exception handling procedures associated with these exchanges. This chapter also describes the FIS protocol in terms of state tables.
- f) 2.4.6: *COMMUNICATION REQUIREMENTS* contains the requirements that the FIS-ASEs impose on the underlying communication system.
- g) 2.4.7: *FIS USER REQUIREMENTS* outlines the requirements that a user of a FIS-ASE must meet.
- h) 2.4.8: *SUBSETTING RULES* contains the conformance requirements which all implementations of the FIS protocol obey.

2.4.1.1.2 Two types of FIS contract may be established on request of the pilot:

- a) the *FIS Demand Contract* where the ground FIS system provides the information immediately and once only, and
- b) the *FIS Update Contract* where the ground FIS system provides the information and any subsequent update of this information.

Note.— The concept of the FIS Demand Contract and the FIS Update Contract used in 2.4 is equivalent to the concept of FIS Demand Mode and FIS Contract Mode developed in the ICAO Manual of Air Traffic Services (ATS) Data Link Applications.

2.4.1.1.3 Multiple “FIS services” may be supported by the FIS application, as for instance:

- a) Automatic Terminal Information Services (ATIS),
- b) Precipitation Map Service,
- c) Terminal Weather Service (TWS),
- d) Windshear Advisory Service,
- e) Pilot Report (PIREP) Service,
- f) Notice to Airmen (NOTAM) Service, and
- g) Runway Visual Range (RVR) Service.

2.4.1.1.4 Each of these services will be accessed and used independently of the others and are initiated by the aircraft avionics (and/or pilot). It will not be required that aircraft avionics include the capability for all of the FIS services.

2.4.1.1.5 The FIS application supports only the FIS service related to ATIS. Additional services and negotiation mechanisms could be incorporated in future packages.

Note.— Functional Descriptions

a) *The **FIS Demand Contract** function:*

- 1) *This function allows the airborne FIS system to establish a FIS demand contract with a ground FIS system. Realisation of the contract involves the sending of a single FIS report from the ground FIS system to the aircraft, optionally after the sending of a positive acknowledgement.*
- 2) *Multiple FIS demand contracts may be established in parallel with a ground FIS server.*
- 3) *The actions performed by the FIS systems supporting the FIS Demand Contract function are the following:*
 - i) *the airborne FIS system formats and sends to the ground FIS system a FIS-demand-contract message. This message identifies the type of FIS information requested and contains the parameters of the request, and*
 - ii) *the ground FIS system then determines whether or not it is able to comply with the request:*
 - A) *if the ground FIS system detects that the requested FIS information can be retrieved but is not yet available, the ground FIS system formats and sends to the airborne FIS*

- system a *FIS-positive-acknowledgement* message first to indicate its acceptance of the contract, and the *FIS-report* message later,
- B) if the ground *FIS* system can comply promptly with the *FIS* demand contract request it sends the *FIS-report* message as soon as possible, or
 - C) if there are errors in the *FIS-demand-contract* message, or if the ground *FIS* system cannot comply with the request, it sends a *FIS-contract-reject* message to the airborne *FIS* system indicating the reason for its inability to accept the contract.
- 4) When an unrecoverable error situation is detected by the airborne or the ground *FIS* system or when either of the users request the abrupt termination of the *FIS* demand contract, the *FIS* system formats and sends to the peer system a *FIS-abort* message indicating the source and the reason of the abort. All *FIS* contracts established between the ground *FIS* system and the airborne *FIS* system are aborted.
- 5) The ***FIS-demand-contract*** message contains the following information:
- i) the reference of the *FIS* contract,
 - ii) the type of *FIS* information requested, and
 - iii) the parameters of the *ATIS* request, i.e.:
 - A) the airport identifier, and
 - B) optionally, the type of the requested *ATIS* (arrival or departure).
- 6) The ***FIS-report*** message contains the following information:
- i) the reference of the *FIS* contract,
 - ii) the type of the *FIS* information returned, and
 - iii) the parameters of the *ATIS*, i.e.:
 - A) the airport identifier,
 - B) the version number of the returned *ATIS*,
 - C) the type of the returned *ATIS* (departure, arrival, both or combined),

- D) optionally, the time of observation of the returned ATIS, and
- E) the ATIS information elements, i.e.:
 - I) the mandatory ATIS elements: identification of the runway(s) in use, runway surface conditions, other operational information, surface winds, visual visibility, cloud, air temperature, dew point temperature, altimeter setting, SIGMET, specific ATIS instructions, and
 - II) the optional ATIS elements: approach type, braking action, holding delay, transition level, runway visibility range, present weather, trend type landing forecast.
- 7) The **FIS-positive-acknowledgement** message contains the reference of the FIS contract.
- 8) The **FIS-contract-reject** message contains the following information:
 - i) the reference of the FIS contract, and
 - ii) the reason of the rejection.
- 9) The **FIS-abort** message contains the following information:
 - i) the type of the FIS contract aborted,
 - ii) the source of the abort of the FIS contract (i.e. FIS service-provider or FIS service-user), and
 - iii) if the source is the FIS service-provider, the reason of the abort.
- b) The **FIS Update Contract** function:
 - 1) This function allows the airborne FIS system to establish an Update Contract with a ground FIS system. Realisation of the contract involves the sending of FIS reports from the ground FIS system to the aircraft each time the requested FIS information is modified.
 - 2) Multiple FIS update contracts may be established in parallel with a ground FIS server.

- 3) *The actions performed by the FIS systems supporting the FIS Update Contract function are the following:*
 - i) *the airborne FIS system formats and sends to the ground FIS system a FIS-update-contract message. This message identifies the type of FIS information requested and contains the parameters of the request, and*
 - ii) *if the ground FIS system can comply with the FIS-update-contract request, then*
 - A) *it sends the first FIS-report message, as soon as possible, and*
 - B) *whenever the requested FIS information is modified, it sends a new FIS-report message to the aircraft.*
 - iii) *if the ground FIS system detects that the requested FIS information can be retrieved but is not yet available, then*
 - A) *it formats and sends to the airborne FIS system a FIS-positive-acknowledgement message first to indicate its acceptance of the contract, and*
 - B) *then sends the FIS-report messages,*
 - iv) *if there are errors in the FIS-update-contract message, or if the ground FIS system cannot comply with the request, it sends a FIS-contract-reject message to the airborne FIS system indicating the reason for its inability to accept the contract, or*
 - v) *if the ground FIS system does not support the update contract function, it sends a FIS-contract-reject message containing, if available, the requested FIS information.*
- 4) *When an error situation is detected by the airborne or the ground FIS system or when either of the users request the abrupt termination of the FIS Update Contract, the FIS system formats and sends to the peer system a FIS-abort message indicating the source and the reason of the abort. All FIS contracts established between the ground FIS system and the airborne FIS system are aborted.*
- 5) *The **FIS-update-contract** message has the same contents as the FIS-demand-contract message as described for the FIS Demand Contract function.*
- 6) *The **FIS-report** messages have the same contents as the FIS-report message as described for the FIS Demand Contract function.*

- 7) The **FIS-contract-reject** message contains the following information:
 - i) the reference of the FIS contract,
 - ii) the reason of the rejection of the FIS contract, and
 - iii) the current value of the requested ATIS, if the reason of the rejection is “FIS Update Contract function not supported by the ground FIS system”.
 - 8) The **FIS-positive-acknowledgement** message has the same contents as the FIS-positive-acknowledgement message as described for the FIS Demand Contract function.
 - 9) The **FIS-abort** message has the same contents as the FIS-abort message as described for the FIS Demand Contract function.
- c) The **Cancellation of Contracts** function:
- 1) This function allows both air and ground FIS system to cancel a particular FIS update contract that is in operation, as follows:
 - i) a FIS-update-contract-cancel message is sent by the system initiating the termination. Any FIS-report message previously sent is delivered to the aircraft before the contract is effectively ended. Other pending FIS contracts are not disrupted by the termination of a particular FIS update contract, and
 - ii) the cancellation of the FIS update contract is confirmed to the FIS-user by the system receiving the FIS-update-contract-cancel message. A FIS-update-contract-cancel-accept message is sent back.
 - 2) The airborne FIS system may also cancel all FIS contracts (demand and update contracts) of the same type in a single FIS-cancel-contracts message. The ground FIS system cancels these contracts and acknowledges the cancellation by sending a FIS-cancel-contracts-accept message. The cancellation is made on the basis of the type(s) of contract supplied by the airborne FIS system.
 - 3) The **FIS-update-contract-cancel** message contains the following information:
 - i) the reference of the FIS contract, and
 - ii) the type of the FIS-update-contract.

- 4) The **FIS-update-contract-cancel-accept** message contains the following information:
 - i) the reference of the FIS contract, and
 - ii) the type of the FIS-update-contract cancelled.
- 5) The **FIS-cancel-contracts** message contains the types of the FIS contracts to be cancelled.
- 6) The **FIS-cancel-contracts-accept** message contains the types of the cancelled FIS contracts.

2.4.2 GENERAL REQUIREMENTS

2.4.2.1 FIS ASE Version Number

Note.— 2.4 describes the version 1 of the protocol operated by the FIS-ASEs. Best efforts will be made to ensure that subsequent versions of this protocol are backwards compatible.

2.4.2.1.1 For this version of the FIS SARPs, the FIS-air-ASE and FIS-ground-ASE version numbers shall both be set to one.

2.4.2.2 Error Processing Requirements

2.4.2.2.1 In the event of information input by the FIS-user being incompatible with that able to be processed by the system, the FIS-user shall be notified.

2.4.2.2.2 In the event of a FIS-user invoking a FIS service primitive when the FIS-ASE is not in a state specified in 2.4.5, the FIS-user shall be notified.

2.4.3 THE ABSTRACT SERVICE

2.4.3.1 Service Description

2.4.3.1.1 An implementation of either the FIS ground based service or the FIS air based service shall exhibit external behaviour consistent with having implemented a FIS-ground-ASE, or a FIS-air-ASE respectively.

Note 1.— 2.4.3 defines the abstract service interface for the FIS service. The FIS-ASE abstract service is described in this chapter from the viewpoint of the FIS-air-user, the FIS-ground-user and the FIS service-provider.

Note 2.— 2.4.3 defines the static behaviour (i.e., the format) of the FIS-ASE abstract service. Its dynamic behaviour (i.e., how it is used) is described in 2.4.7.

Note 3.— Figure 2.4.3-1 shows the functional model of the FIS Application. The functional modules identified in this model are the following:

- a) the FIS-user,
- b) the FIS Application Entity (FIS-AE) service interface,
- c) the FIS-AE,
- d) the FIS Control Function (FIS-CF),
- e) the FIS Application Service Element (FIS-ASE) service interface,
- f) the FIS-ASE, and
- g) the Dialogue Service (DS) interface.

Note 4.— The FIS-user represents the operational part of the FIS system. This user does not perform the communication functions but relies on a communication service provided to it via the FIS-AE through the FIS-AE service interface. The individual actions at this interface are called FIS-AE service primitives. Similarly, individual actions at other interfaces in the communication system are called service primitives at these interfaces.

Note 5.— The FIS-AE consists of several elements, including the FIS-ASE and the FIS-CF. The DS interface is made available by the FIS-CF to the FIS-ASE for communication with the peer FIS-ASE.

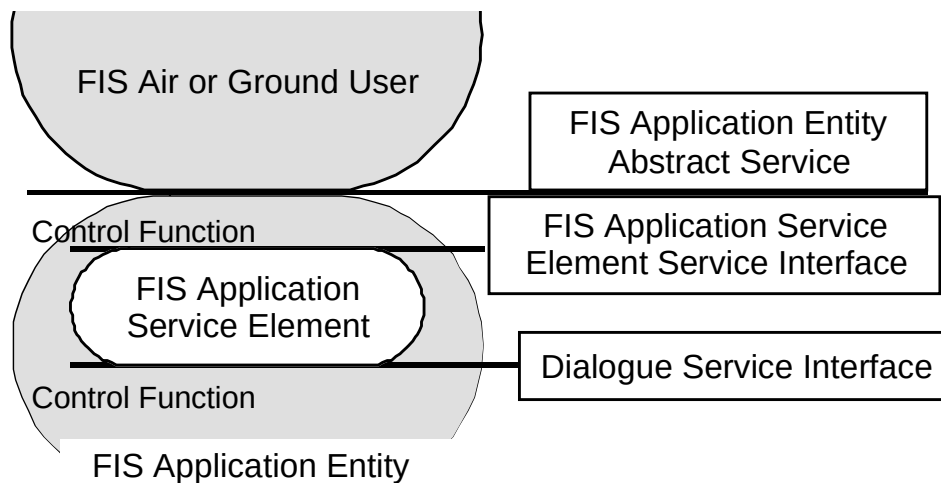


Figure 2.4.3-1. Functional Model of the FIS Application

Note 6.— The FIS-ASE is the element in the communication system which executes the FIS specific protocol. In other words, it takes care of the FIS specific service primitive sequencing actions, message creation, timer management, error and exception handling.

Note 7.— The FIS-ASE interfaces only with the FIS-CF. This FIS-CF is responsible for mapping service primitives received from one element (such as the FIS-ASE and the FIS-user) to other elements which interface with it. The part of the FIS-CF which is relevant from the point of view of these SARPs, i.e. the part between the FIS-user and the FIS-ASE, will map FIS-AE service primitives to FIS-ASE service primitives transparently.

Note 8.— The DS interface is the interface between the FIS-ASE and part of FIS-CF underneath the FIS-ASE, and provides the dialogue service as described in 4.2.

2.4.3.2 The FIS-ASE Abstract Service

Note.— There is no requirement to implement the service in a FIS product; however, it is necessary to implement the ground based and air based system in such a way that it will be impossible to detect (from the peer system) whether or not an interface has been built.

2.4.3.2.1 The FIS-ASE abstract service shall consist of a set of the following services as allowed by the subsetting rules defined in 2.4.8:

- a) *FIS-demand-contract* service as defined in 2.4.3.3,
- b) *FIS-update-contract* service as defined in 2.4.3.4,
- c) *FIS-report* service as defined in 2.4.3.5,

- d) *FIS-cancel-contracts* service as defined in 2.4.3.6,
- e) *FIS-cancel-update-contract* service as defined in 2.4.3.7,
- f) *FIS-user-abort* service as defined in 2.4.3.8, and
- g) *FIS-provider-abort* service as defined in 2.4.3.9.

Note 1.— For a given primitive, the presence of each parameter is described by one of the following values in the parameter tables of this chapter:

- a) **blank** not present,
- b) **C** conditional upon some predicate explained in the text,
- c) **C(=)** conditional upon the value of the parameter to the immediate left being present, and equal to that value,
- d) **M** mandatory,
- e) **M(=)** mandatory, and equal to the value of the parameter to the immediate left, and
- f) **U** user option.

Note 2.— The following abbreviations are used in this part:

- a) **Req** request; data is input by FIS-user initiating the service to its respective ASE,
- b) **Ind** indication; data is indicated by the receiving ASE to its respective FIS-user,
- c) **Rsp** response; data is input by receiving FIS-user to its respective ASE, and
- d) **Cnf** confirmation; data is confirmed by the initiating ASE to its respective FIS-user.

Note 3.— An unconfirmed service allows just one message to be transmitted, in one direction, without providing a corresponding response.

Note 4.— A confirmed service provides end-to-end confirmation that a message sent by one user was received by its peer user.

Note 5.— An abstract syntax is a syntactical description of a parameter which does not imply a specific implementation. Only when the FIS-ASE maps a parameter onto an APDU field, or vice-versa, is the abstract syntax of the parameter described by using the ASN.1 of 2.4.4 for this field.

2.4.3.3 FIS-demand-contract Service

Note.— The FIS-demand-contract service allows the FIS-air-user to request a FIS demand contract with a FIS-ground-user. It is a confirmed service, initiated by the FIS-air-user.

2.4.3.3.1 The FIS-demand-contract service shall contain primitives and parameters as presented in Table 2.4.3-1.

Table 2.4.3-1. FIS-demand-contract service parameters

Parameter Name	Req	Ind	Rsp	Cnf
ICAO Facility Designation	M			
FIS Contract Number	M	M(=)	M(=)	M(=)
Class of Communication Service	U			
FIS Contract Details	M	M(=)		
Result			M	M(=)
Reject Reason			C	C(=)
FIS Information			C	C(=)

2.4.3.3.2 ICAO Facility Designation

Note.— This parameter contains the FIS-ground-ASE ICAO Facility Designation.

2.4.3.3.2.1 The *ICAOFacilityDesignation* parameter value shall conform to the abstract syntax four to eight-character *ICAO Facility Designation*.

2.4.3.3.3 FIS Contract Number

Note.— This parameter contains the user-defined reference of the requested FIS demand contract.

2.4.3.3.3.1 The *FISContractNumber* parameter value shall conform to the ASN.1 abstract syntax *ContractNumber*.

2.4.3.3.4 Class Of Communication Service

Note.— This parameter contains the value of the required class of communication service, if specified by the FIS-air-user.

2.4.3.3.4.1 Where specified by the FIS-air-user, the *ClassOfCommunicationService* parameter value shall have one of the following abstract values: “A”, “B”, “C”, “D”, “E”, “F”, “G” or “H”.

Note 1.— If FIS contracts are currently in place, the ClassOfCommunicationService parameter is not used by the FIS service provider.

Note 2.— Where not specified by the FIS-air-user, when there are no FIS Contracts already in force, this indicates that there is no routing preference.

2.4.3.3.5 FIS Contract Details

Note.— This parameter contains the details of the FIS Demand Contract as requested by the FIS-air-user.

2.4.3.3.5.1 The *FISContractDetails* parameter value shall conform to the ASN.1 abstract syntax *FISRequestData*.

Note.— This parameter identifies also the type of the requested FIS information. For version 1 of the FIS-ASEs, the requested information is always of type “ATIS”.

2.4.3.3.6 Result

Note.— The value of this parameter indicates whether the FIS-demand-contract request has been accepted or rejected by the FIS-ground-user.

2.4.3.3.6.1 The *Result* parameter value shall conform to one of the following abstract values:

- a) “accepted”,
- b) “positive acknowledgment”, and
- c) “rejected”.

2.4.3.3.7 Reject Reason

Note.— This parameter contains the reason of the rejection.

2.4.3.3.7.1 The *RejectReason* parameter value shall conform to the ASN.1 abstract syntax *FISRejectReason*.

2.4.3.3.7.2 The *RejectReason* parameter shall be present if and only if the *Result* parameter contains the abstract value “rejected”.

2.4.3.3.8 FIS Information

Note.— This parameter contains the FIS information, as requested by the FIS-air-user.

2.4.3.3.8.1 The *FISInformation* parameter value shall conform to the ASN.1 abstract syntax *FISReportData*.

Note.— This parameter identifies also the type of the returned FIS information. For version 1 of the FIS-ASEs, the requested information is always of type “ATIS”.

2.4.3.3.8.2 The *FISInformation* parameter shall be present if and only if the *Result* parameter contains the abstract value “accepted”.

2.4.3.4 FIS-update-contract Service

Note.— The FIS-update-contract service allows the FIS-air-user to request a FIS update contract with a FIS-ground-user. It is a confirmed service, initiated by the FIS-air-user.

2.4.3.4.1 The FIS-update-contract service shall contain primitives and parameters as presented in Table 2.4.3-2.

Table 2.4.3-2. FIS-update-contract service parameters

Parameter Name	Req	Ind	Rsp	Cnf
ICAO Facility Designation	M			
FIS Contract Number	M	M(=)	M(=)	M(=)
Class Of Communication Service	U			
FIS Contract Details	M	M(=)		
Result			M	M(=)
Reject Reason			C	C(=)
FIS Information			C	C(=)

2.4.3.4.2 ICAO Facility Designation

Note.— This parameter contains the FIS-ground-ASE ICAO Facility Designation.

2.4.3.4.2.1 The *ICAOFacilityDesignation* parameter value shall conform to the abstract syntax four to eight-character *ICAO Facility Designation*.

2.4.3.4.3 FIS Contract Number

Note.— This parameter contains the user-defined reference of the requested FIS update contract.

2.4.3.4.3.1 The *FISContractNumber* parameter value shall conform to the ASN.1 abstract syntax *ContractNumber*.

2.4.3.4.4 Class Of Communication Service

Note.— This parameter contains the value of the required class of communication service, if specified by the FIS-air-user.

2.4.3.4.4.1 Where specified by the FIS-air-user, the *ClassOfCommunicationService* parameter value shall have one of the following abstract values: “A”, “B”, “C”, “D”, “E”, “F”, “G” or “H”.

Note 1.— If FIS contracts are currently in place, the *ClassOfCommunicationService* parameter is not used by the FIS service provider.

Note 2.— Where not specified by the FIS-air-user, when there are no FIS Contracts already in force, this indicates that there is no routing preference.

2.4.3.4.5 FIS Contract Details

Note.— This parameter contains the details of the FIS Update Contract as requested by the FIS-air-user.

2.4.3.4.5.1 The *FISContractDetails* parameter value shall conform to the ASN.1 abstract syntax *FISRequestData*.

Note.— This parameter identifies also the type of the requested FIS information. For version 1 of the FIS-ASEs, the requested information is always of type “ATIS”.

2.4.3.4.6 Result

Note.— The value of this parameter indicates whether the FIS-update-contract request has been accepted or rejected by the FIS-ground-user.

2.4.3.4.6.1 The *Result* parameter value shall conform to one of the following abstract values:

- a) “accepted”,
- b) “positive acknowledgment”, and
- c) “rejected”.

2.4.3.4.7 Reject Reason

Note.— This parameter contains the reason of the rejection.

2.4.3.4.7.1 The *RejectReason* parameter value shall conform to one of the following abstract values:

- a) “can not comply”,
- b) “FIS service unavailable”,
- c) “error detected in the FIS request”,
- d) “update contract function not supported by the FIS-ground-user”, and
- e) “undefined”.

2.4.3.4.7.2 The *RejectReason* parameter shall be present if and only if the *Result* parameter contains the abstract value “rejected”.

2.4.3.4.8 FIS Information

Note.— This parameter contains the FIS information requested by the FIS-air-user.

2.4.3.4.8.1 The *FISInformation* parameter value shall conform to the ASN.1 abstract syntax *FISReportData*.

Note.— This parameter identifies also the type of the returned FIS information. For version 1 of the FIS-ASEs, the requested information is always of type “ATIS”.

2.4.3.4.8.2 The *FISInformation* parameter shall be present if the *Result* parameter contains the abstract value “accepted”.

Note.— The *FISInformation* parameter is present upon user’s choice if the *Result* parameter contains the abstract value “rejected” and the *RejectReason* parameter contains the abstract value “update contract function not supported by the FIS-ground-user”.

2.4.3.5 FIS-report Service

Note.— The FIS-report service allows the FIS-ground-user to send to the FIS-air-user the requested FIS information and any update of this FIS information. It is an unconfirmed service initiated by the FIS-ground-user.

2.4.3.5.1 The FIS-report service shall contain primitives and parameters as presented in Table 2.4.3-3.

Table 2.4.3-3. FIS-report service parameters

Parameter Name	Req	Ind
FIS Contract Number	M	M(=)
FIS Information	M	M(=)

2.4.3.5.2 FIS Contract Number

Note.— This parameter contains the user-defined reference of the FIS demand contract the FIS Information is related to.

2.4.3.5.2.1 The *FISContractNumber* parameter value shall conform to the ASN.1 abstract syntax *ContractNumber*.

2.4.3.5.3 FIS Information

Note.— This parameter contains the FIS information requested by the FIS-air-user.

2.4.3.5.3.1 The *FISInformation* parameter value shall conform to the ASN.1 abstract syntax *FISReportData*.

Note.— This parameter identifies also the type of the returned FIS information. For version 1 of the FIS-ASEs, the requested information is always of type “ATIS”.

2.4.3.6 FIS-cancel-contracts Service

Note.— The FIS-cancel-contracts service allows the FIS-air-user to cancel all FIS demand and update contracts of the same type with a particular FIS-ground-user. It is a confirmed service initiated by the FIS-air-user.

2.4.3.6.1 The FIS-cancel-contracts service shall contain primitives and parameters as presented in Table 2.4.3-4.

Table 2.4.3-4. FIS-cancel-contracts service parameters

Parameter Name	Req	Ind	Cnf
FIS Service Type	M	M(=)	M(=)

2.4.3.6.2 FIS Service Type

Note.— This parameter identifies the types of the FIS contracts to be cancelled.

2.4.3.6.2.1 The *FISServiceType* parameter value shall conform to the ASN.1 abstract syntax *FISCancelContracts*.

Note.— For version 1 of the FIS-ASEs, this parameter will always identify the service type “ATIS”.

2.4.3.7 FIS-cancel-update-contract Service

Note.— The FIS-cancel-update-contract service allows the FIS-air-user or the FIS-ground-user to cancel in an orderly manner an active FIS update contract. The FIS reports in transit in the communication system are delivered before the FIS update contract is cancelled. This service does not affect the other FIS contracts. This is a confirmed service, initiated by the FIS-air-user or the FIS-ground-user.

2.4.3.7.1 The FIS-cancel-update-contract service shall contain primitives and parameters as presented in Table 2.4.3-5.

Table 2.4.3-5. FIS-cancel-update-contract service parameters

Parameter Name	Req	Ind	Cnf
FIS Contract Number	M	M(=)	M(=)

2.4.3.7.2 FIS Contract Number

Note.— This parameter contains the reference of the FIS update contract to be cancelled.

2.4.3.7.2.1 The *FISContractNumber* parameter value shall conform to the ASN.1 abstract syntax *ContractNumber*.

2.4.3.8 FIS-user-abort Service

Note 1.— The FIS-user-abort service allows a FIS-user to abort all active FIS contracts (both FIS demand contracts and FIS update contracts). As a consequence, all active FIS contracts processed by the ASE are cancelled. Messages in transit may be lost during this operation. This is an unconfirmed service. It can be invoked at any time that the FIS-user is aware that any FIS service is in operation.

Note 2.— If the FIS-user-abort service is invoked prior the complete establishment of the dialogue, the FIS-user-abort indication may not be provided. A FIS-provider-abort-indication may result instead.

2.4.3.8.1 The FIS-user-abort service shall contain the primitives as presented in Table 2.4.3-6.

Table 2.4.3-6. FIS-user-abort service parameters

Parameter Name	Req	Ind
none		

2.4.3.9 FIS-provider-abort Service

Note.— The FIS-provider-abort service is used by the FIS service-provider to inform its active users that it can no longer provide the FIS service. As a consequence, all active FIS contracts (both FIS demand contract and FIS update contract) are cancelled. This service is a FIS service-provider initiated service. Messages in transit may be lost during this operation.

2.4.3.9.1 The FIS-provider-abort service shall contain primitives and parameters as presented in Table 2.4.3-7.

Table 2.4.3-7. FIS-provider-abort service parameters

Parameter Name	Ind
Reason	M

2.4.3.9.2 Reason

Note.— This parameter contains the reason for the abort.

2.4.3.9.2.1 The Reason parameter value shall conform to one of the following abstract values:

- a) “timer expiration”,
- b) “protocol error”,
- c) “sequence error”,
- d) “decoding error”,

- e) “unrecoverable internal error”,
- f) “invalid contract number”,
- g) “dialogue end not supported”,
- h) “undefined”,
- i) “invalid QOS parameter”,
- j) “cannot establish contact with the peer”,
- k) “contact refused by the peer”, and
- l) “communication system failure”.

2.4.4 FORMAL DEFINITIONS OF MESSAGES

2.4.4.1 Encoding/Decoding Rules

2.4.4.1.1 A FIS-air-ASE shall be capable of encoding [FISDownlinkAPDU] APDUs and decoding [FISUplinkAPDU] APDUs.

2.4.4.1.2 A FIS-ground-ASE shall be capable of encoding [FISUplinkAPDU] APDUs and decoding [FISDownlinkAPDU] APDUs.

2.4.4.2 FIS ASN.1 Abstract Syntax

2.4.4.2.1 The abstract syntax of the FIS protocol data units shall comply with the description contained in the ASN.1 module FISMessageSetVersion1 (conforming to ISO/IEC 8824-1), as defined in 2.4.4.

FISMessageSetVersion1 DEFINITIONS::=

BEGIN

```
-- -----
-- FIS messages (Top level)
-- -----
```

FISDownlinkAPDU ::= SEQUENCE

```
{
    time                DateTimeGroup,
    fisDownlinkAPDU     DownlinkAPDU
}
```

FISUplinkAPDU ::= SEQUENCE

```
{
    time                DateTimeGroup,
    fisUplinkAPDU       UplinkAPDU
}
```

DownlinkAPDU ::= CHOICE

```
{
    FISRequest           [0] FISRequest,
    FISCancelUpdateContract [1] FISCancelUpdateContract,
    FISCancelUpdateAccept [2] FISCancelUpdateAccept,
    FISCancelContracts    [3] FISCancelContracts,
    FISAbort              [4] FISAbort,
    ...
}
```

UplinkAPDU ::= CHOICE

{	
fISAccept	[0] FISAccept,
fISReject	[1] FISReject,
fISReport	[2] FISReport,
fISCancelUpdateContract	[3] FISCancelUpdateContract,
fISCancelUpdateAccept	[4] FISCancelUpdateAccept,
fISCancelContractsAccept	[5] FISCancelContractsAccept,
fISAbort	[6] FISAbort,
...	
}	

 -- FIS messages (2nd level)

FISAbort ::= CHOICE

{	
-- Automatic Terminal Information Service (ATIS)	
atis	[0] FISProtocolErrorDiag,
...	
}	

FISAccept ::= SEQUENCE

{	
contractNumber	ContractNumber,
fISAcceptData	FISAcceptData
}	

FISAcceptData ::= CHOICE

{	
accept	[0] FISReportData,
positiveAcknowledgement	[1] NULL
}	

FISCancelContracts ::= FISServiceType

FISCancelContractsAccept ::= FISServiceType

FISCancelAcceptData::= CHOICE

```
{  
  -- Automatic Terminal Information Service (ATIS)  
  atis [0] NULL,  
  ...  
}
```

FISCancelUpdateAccept ::= SEQUENCE

```
{  
  flSUpdateContractNumber ContractNumber,  
  flSCancelAcceptData FISCancelAcceptData  
}
```

FISCancelUpdateContract ::= SEQUENCE

```
{  
  flSUpdateContractNumber ContractNumber,  
  flSCancelUpdateData FISCancelUpdateData  
}
```

FISCancelUpdateData ::= CHOICE

```
{  
  -- Automatic Terminal Information Service (ATIS)  
  atis [0] NULL,  
  ...  
}
```

FISReject ::= SEQUENCE

```
{  
  contractNumber ContractNumber,  
  flSRejectData FISRejectData  
}
```

FISRejectData ::= CHOICE

```
{  
  updateFunctionNotSupported [0] NULL,  
  updateFunctionNotSupportedWithReport [1] FISReportData,  
  otherReasons [2] FISRejectReason,  
  ...  
}
```

FISRejectReason ::= ENUMERATED

```
{
  canNotComply           (0),
  fISServiceUnavailable  (1),
  errorInRequest         (2),
  undefined               (3),
  ...
}
```

FISReport ::= SEQUENCE

```
{
  contractNumber          ContractNumber,
  fISReportData           FISReportData
}
```

FISReportData ::= CHOICE

```
{
  -- Automatic Terminal Information Service (ATIS)
  atis                     [0] ATISReport,
  ...
}
```

FISRequest ::= SEQUENCE

```
{
  contractNumber          ContractNumber,
  contractType            ContractType          DEFAULT demandContract,
  fISRequestData          FISRequestData
}
```

FISRequestData ::= CHOICE

```
{
  -- Automatic Terminal Information Service (ATIS)
  aTISRequest             [0] ATISRequest,
  ...
}
```

FISProtocolErrorDiag ::= ENUMERATED

```
{
  timerExpiration           (0),
  protocolError             (1),
  sequenceError             (2),
  decodingError             (3),
  unrecoverableInternalError (4),
  invalidContractNumber     (5),
  dialogueEndNotSupported   (6),
  undefined                 (7),
  invalidQosParameter       (8),
  ...
}
```

FISServiceType ::= BIT STRING

```
{
  -- Automatic Terminal Information Service (ATIS)
  atis                (0)
}
(SIZE (1,...))
```

 -- ATIS Messages

ATISInformation ::= CHOICE

```
{
  arrivalATIS           [0] ArrivalATIS,
  departureATIS         [1] DepartureATIS,
  combinedATIS          [2] CombinedATIS,
  arrivalAndDepartureATIS [3] ArrivalAndDepartureATIS
}
```

ATISReport ::= SEQUENCE

```
{
  aerodromeID           Aerodrome,
  aTISInformation       ATISInformation
}
```

ATISRequest ::= SEQUENCE

```
{
  aerodromeID           Aerodrome,
  arrivalDepartureIndicator ArrivalDepartureIndicator OPTIONAL
}
```

ArrivalAndDepartureATIS ::= SEQUENCE

{		
arrivalATIS	ArrivalATIS,	
departureATIS	DepartureATIS	
}		

ArrivalATIS ::= SEQUENCE

{			
aTISDesignator	[0] ATISDesignator,		
aTISTimeOfObservation	[1] Time	OPTIONAL,	
arrivalRunwaysInUse	[2] SEQUENCE (SIZE (1..36))		
	OF ArrivalRunway,		
commonATISInfo	[3] CommonATISInformation,		
arrivalATISInfo	[4] SpecificATISArrivalInfo		
}			

CombinedATIS ::= SEQUENCE

{			
aTISDesignator	[0] ATISDesignator,		
aTISTimeOfObservation	[1] Time	OPTIONAL,	
runwaysInUse	[2] SEQUENCE (SIZE (1..36))		
	OF RunwayType,		
commonATISInfo	[3] CommonATISInformation,		
arrivalATISInfo	[4] SpecificATISArrivalInfo		
}			

DepartureATIS ::= SEQUENCE

{			
aTISDesignator	[0] ATISDesignator,		
aTISTimeOfObservation	[1] Time	OPTIONAL,	
departureRunwaysInUse	[2] SEQUENCE (SIZE (1..36))		
	OF DepartureRunway,		
commonATISInfo	[3] CommonATISInformation		
}			

CommonATISInformation ::= SEQUENCE

{			
surfaceWind	[0] SurfaceWind,		
visibility	[1] Visibility	OPTIONAL,	
cloud	[2] Cloud,		
airTemperature	[3] Temperature,		
dewPointTemperature	[4] Temperature,		
altimeterSetting	[5] AltimeterSetting,		
presentWeather	[6] PresentWeather	OPTIONAL,	
significantMetPhenomena	[7] SignificantMetPhenomena	OPTIONAL,	
holdingDelay	[8] IA5String (SIZE(1..200))	OPTIONAL,	
}			

specificATISInstructions	[9] IA5String (SIZE (1..64))	OPTIONAL,
otherOperationInfo	[10] IA5String (SIZE (1..250))	OPTIONAL,
transitionLevel	[11] TransitionLevel	OPTIONAL,
...		
}		

SpecificATISArrivalInfo ::= SEQUENCE

{		
trendTypeLandingForecast	IA5String (SIZE (1..256))	OPTIONAL,
...		
}		

 -- ATIS fields

-- Note: this should be read in conjunction with ICAO Annexes 3, 4, 5, 11, 14 and 15

Aerodrome ::= IA5String (SIZE(4))

AltimeterSetting ::= SEQUENCE

{		
qNH	[0] PressureMeasure,	
qFEAerodrome	[1] PressureMeasure	OPTIONAL,
qFERunway	[2] SEQUENCE (SIZE(1..36))	
	OF RunwayQFE	OPTIONAL
}		

Approach ::= ENUMERATED

{	
ils	(0),
localizer	(1),
ndb	(2),
vor	(3),
vordme	(4),
nonprecisiongps	(5),
precisiongps	(6),
dmearc	(7),
precisionapproachradar	(8),
asr	(9),
visual	(10),
rnav	(11),
chartedvisualapproachprocedure	(12),
lda	(13),
fms	(14),
loran	(15),
mls	(16),
ilsdme	(17),
localizerbackcourse	(18),
localizerDME	(19),

vortac	(20),
tacan	(21),
ndbDme	(22),
vdf	(23),
sra	(24),
...	
}	

ApproachType ::= SEQUENCE

{		
approachType	[0] Approach,	
approachTypeOther	[1] IA5String (SIZE (1..30))	OPTIONAL
}		

ArrivalDepartureIndicator ::= ENUMERATED

{	
arrival	(0),
departure	(1),
combined	(2)
}	

ArrivalRunway ::= SEQUENCE

{		
runway	[0] Runway,	
approachType	[1] ApproachType	OPTIONAL,
circleRunway	[2] RunwayId	OPTIONAL
}		

ATISDesignator ::= IA5String (SIZE(1))

BrakingAction ::= SEQUENCE

{		
brakingActionFull	[0] BrakingActionDescription	OPTIONAL,
brakingActionFirstThird	[1] BrakingActionDescription	OPTIONAL,
brakingActionSecondThird	[2] BrakingActionDescription	OPTIONAL,
brakingActionLastThird	[3] BrakingActionDescription	OPTIONAL
}		

BrakingActionDescription ::= SEQUENCE

{	
brakingActionQuality	BrakingActionQuality,
brakingActionQualifier	IA5String (SIZE (1..25))
}	

BrakingActionQuality ::= ENUMERATED

```
{
  good                      (0),
  mediumToGood              (1),
  medium                    (2),
  mediumToPoor              (3),
  poor                      (4)
}
```

Cloud ::= CHOICE

```
{
  cloudInfo                  [0] CloudInformation,
  skyObscured                [1] SkyObscured,
  cAVOK                      [2] NULL
}
```

CloudAmount ::= ENUMERATED

```
{
  skyclear                  (0),
  few                       (1),
  scattered                  (2),
  broken                     (3),
  overcast                  (4)
}
```

CloudHeight ::= CHOICE

```
{
  lowCloudHeightMeters      [0] INTEGER (0..100),
  -- Units = meters, range (0..3000), resolution = 30
  lowCloudHeightFeet        [1] INTEGER (0..100),
  -- Units = feet, range (0..10000), resolution = 100
  highCloudHeightMeters     [2] INTEGER (11..67),
  -- Units = meters, range (3300..20100), resolution = 300
  highCloudHeightFeet       [3] INTEGER (11..60)
  -- Units = feet, range (11000..60000), resolution = 1000
}
```

CloudInformation ::= SEQUENCE

```
{
  cloudAmount                [0] CloudAmount,
  cloudHeight                [1] CloudHeight
  cloudType                  [2] CloudType
}
```

OPTIONAL,
OPTIONAL

CloudType ::= ENUMERATED

```
{
  cumulonimbus              (0),
  toweringCumulus           (1)
}
```

CombinedRunway ::= SEQUENCE

{		
runway	[0] Runway,	
approachType	[1] ApproachType	OPTIONAL,
circleRunway	[2] RunwayId	OPTIONAL
}		

ContractNumber ::= INTEGER (1..256)

ContractType ::= ENUMERATED

{	
demandContract	(0),
updateContract	(1)
}	

Date ::= SEQUENCE

{	
year	Year,
month	Month,
day	Day
}	

DateTimeGroup ::= SEQUENCE

{	
date	Date,
time	HHMMSS
}	

Day ::= INTEGER (1..31)

-- unit = day, range (1..31), resolution = 1

DepartureRunway ::= Runway

DescriptorQualifier ::= ENUMERATED

{	
shallow	(0),
partial	(1),
patches	(2),
lowDrifting	(3),
blowing	(4),
shower	(5),
thunderstorm	(6),
freezing	(7)
}	

HHMMSS ::= SEQUENCE

{	
timeHours	TimeHours,
timeMinutes	TimeMinutes,
timeSeconds	TimeSeconds
}	

IntensityQualifier ::= ENUMERATED

{	
light	(0),
moderate	(1),
heavy	(2)
}	

Month ::= INTEGER (1..12)

-- unit = month, range (1..12), resolution = 1

Obscuration ::= ENUMERATED

{	
mist	(0),
fog	(1),
smoke	(2),
volcanicAsh	(3),
widespreadDust	(4),
sand	(5),
haze	(6)
}	

OtherWeatherPhenomena ::= ENUMERATED

{	
dustSandWhirls	(0),
squalls	(1),
funnelCloudTornadoWaterspout	(2),
sandstorm	(3),
duststorm	(4)
}	

Precipitation ::= ENUMERATED

{	
drizzle	(0),
rain	(1),
snow	(2),
snowGrains	(3),
iceCrystals	(4),
icePellets	(5),
hail	(6),
smallHailAndOrSnowPellets	(7),
unknownPrecipitation	(8)
}	

PresentWeather ::= SEQUENCE

{			
presentWeather	[0] PresentWX,		
nosig	[1] NULL		OPTIONAL
}			

PresentWX ::= SEQUENCE

{			
type	[0] PresentWeatherType,		
intensityQualifier	[1] IntensityQualifier		OPTIONAL,
inTheVicinity	[2] NULL		OPTIONAL,
descriptorQualifier	[3] DescriptorQualifier		OPTIONAL
}			

PresentWeatherType ::= SEQUENCE

{			
precipitation	[0] Precipitation		OPTIONAL,
obscuration	[1] Obscuration		OPTIONAL,
otherWeatherPhenomena	[2] OtherWeatherPhenomena		OPTIONAL
}			

PressureMeasure ::= CHOICE

{		
hPa	[0] INTEGER (500..1250),	
-- units = hPa, range (500..1250), resolution = 1		
inches	[1] INTEGER (2200..3200)	
-- units = inches of Mercury, range (22.00..32.00), resolution= 0.01		
}		

Runway ::= SEQUENCE

{			
runwayId	[0] RunwayId,		
runwaySurfaceConditions	[1] RunwaySurfaceConditions		OPTIONAL,
brakingAction	[2] BrakingAction		OPTIONAL,
runwayArrestingSystem	[3] SEQUENCE (SIZE (1..2))		
	OF RunwayArrestingSystem		OPTIONAL,
runwayVisualRange	[4] RVR		OPTIONAL
}			

RASCondition ::= ENUMERATED

{	
up	(0),
down	(1),
unavailable	(2)
}	

RASLocation ::= ENUMERATED

{	
arrivalEnd	(0),
departureEnd	(1)
}	

RunwayArrestingSystem ::= SEQUENCE

{	
location	RASLocation,
condition	RASCondition
}	

RunwayDesignator ::= INTEGER (1..36)

RunwayId ::= SEQUENCE

{		
runwayDesignator	[0] RunwayDesignator,	
runwayLetter	[1] RunwayLetter	OPTIONAL
}		

RunwayLetter ::= ENUMERATED

{	
leftLeft	(0),
left	(1),
leftCenter	(2),
center	(3),
rightCenter	(4),
right	(5),
rightRight	(6)
}	

RunwayQFE ::= SEQUENCE

{	
runwayId	RunwayId,
qFE	PressureMeasure
}	

RunwaySurfaceConditions ::= SEQUENCE

{		
surfaceConditions	[0] SurfaceConditions,	
other	[1] IA5String (SIZE (1..256))	OPTIONAL
}		

RunwayType ::= CHOICE

{	
arrivalRunway	[0] ArrivalRunway,
departureRunway	[1] DepartureRunway,
combinedRunway	[2] CombinedRunway
}	

RunwayVisibility ::= CHOICE

{		
inoperative	[0] NULL,	
reported	[1] RVRVisibility	
}		

RVR ::= SEQUENCE

{			
touchdownRVR	[0] RunwayVisibility,		
midPointRVR	[1] RunwayVisibility	OPTIONAL,	
stopEndRVR	[2] RunwayVisibility	OPTIONAL,	
rVRVariationQualifier	[3] SEQUENCE (SIZE(2))		
	OF RVRVisibility	OPTIONAL	
}			

RVRVisibility ::= CHOICE

{	
lowVisibilityMeters	[0] INTEGER (0..32),
-- units = meters, range (0..800), resolution = 25	
highVisibilityMeters	[1] INTEGER (9..15),
-- units = meters, range (900..1500), resolution = 100	
lowVisibilityFeet	[2] INTEGER (0..10),
-- units = feet, range (0..1000), resolution = 100	
midVisibilityFeet	[3] INTEGER (6..15),
-- units = feet, range (1200..3000), resolution = 200	
highVisibilityFeet	[4] INTEGER (7..12)
-- units = feet, range (3500..6000), resolution = 500	
}	

SkyObscured ::= CHOICE

{	
noVerticalVisibility	[0] NULL,
verticalVisibility	[1] VerticalVisibility
}	

SignificantMetPhenomena ::= SEQUENCE

{		
approachAreaMet	[0] IA5String (SIZE (1..128))	OPTIONAL,
takeoffAreaMet	[1] IA5String (SIZE (1..128))	OPTIONAL,
climboutAreaMet	[2] IA5String (SIZE (1..128))	OPTIONAL
}		

SurfaceConditions ::= ENUMERATED

```
{
  damp                (0),
  wet                 (1),
  waterPatches        (2),
  flooded              (3),
  wetSnow              (4),
  drySnow              (5),
  snow                 (6),
  slush                (7),
  ice                  (8)
}
```

SurfaceWind ::= CHOICE

```
{
  calmIndicator        [0] NULL,
  surfaceWind          [1] SurfaceWD
}
```

SurfaceWD ::= SEQUENCE

```
{
  surfaceWindDirection [0] SurfaceWindDirection,
  surfaceWindSpeed      [1] SurfaceWindSpeed,
  surfaceWindVariations [2] SurfaceWindVariations OPTIONAL
}
```

SurfaceWindDirection ::= INTEGER (1..36)

-- units = degree, range (10..360), resolution = 10

-- wind direction is the direction from which the wind is coming

SurfaceWindSpeed ::= CHOICE

```
{
  windSpeedMeters      [0] INTEGER (0..500),
  -- units = kilometerperhour, range (0..500), resolution = 1
  windSpeedKnots       [1] INTEGER (0..200)
  -- units = knots, range (0..200), resolution = 1
}
```

SurfaceWindVariations ::= SEQUENCE

```
{
  direction1           [0] SurfaceWindDirection OPTIONAL,
  direction2           [1] SurfaceWindDirection OPTIONAL,
  speedMax             [2] SurfaceWindSpeed OPTIONAL
}
```

Temperature ::= CHOICE

```
{
  notAvailable          [0] NULL,
  temperature           [1] INTEGER (-80..60)
  -- units = degree Celsius, range (-80..60), resolution = 1
}
```

Time ::= SEQUENCE

```
{
  timeHours             TimeHours,
  timeMinutes           TimeMinutes
}
```

TimeHours ::= INTEGER (0..23)

-- units = hours, range (0..23), resolution = 1

TimeMinutes ::= INTEGER (0..59)

-- units = minutes, range (0..59), resolution = 1

TimeSeconds ::= INTEGER (0..59)

-- units = seconds, range (0..59), resolution = 1

TransitionLevel ::= CHOICE

```
{
  flightLevelMeter      [0] INTEGER (100..2000),
  -- units = meters, range (1000..20000), resolution = 10 meters
  flightLevelFeet       [1] INTEGER (30..600)
  -- units = feet, range (3000..60000), resolution = 100 feet
}
```

VerticalVisibility ::= CHOICE

```
{
  visibilityMeters       [0] INTEGER (0..100),
  -- units = meters, range (0..3000), resolution = 30
  visibilityFeet         [1] INTEGER (0..100),
  -- units = feet, range (0..10000), resolution = 100
  notAvailable           [2] NULL
}
```


VisibilityStatuteMiles ::= CHOICE

```
{
  oneSixteenth          [0] INTEGER (0..6),
  -- units = Statute Mile, range (0..3/8), resolution = 1/16th
  oneEighth             [1] INTEGER (3..16),
  -- units = Statute Mile, range (3/8..2), resolution = 1/8th
  oneFourth             [2] INTEGER (8..12),
  -- units = Statute Mile, range (2..3), resolution = 1/4th
  one                   [3] INTEGER (3..15),
  -- units = Statute Mile, range (3..15), resolution = 1
  five                  [4] INTEGER (3..10),
  -- units = Statute Mile, range (15..50), resolution = 5
  moreThanFifty         [5] NULL
}
```

Visibility ::= CHOICE

```
{
  lowMeter              [0] INTEGER (0..10),
  -- units = meters, range (0..500), resolution = 50
  highMeter             [1] INTEGER (6..49),
  -- units = meters, range (600..4900), resolution = 100
  kms                   [2] INTEGER (5..10),
  -- units = kms, range (5..10), resolution = 1
  statuteMiles          [3] VisibilityStatuteMiles
}
```

Year ::= INTEGER (1996..2095)

-- unit = year, range (1996..2095), resolution = 1

END

2.4.5 PROTOCOL DEFINITION

2.4.5.1 Sequence Rules

2.4.5.1.1 With the exception of abort primitives, only the sequence of primitives illustrated in figures 2.4.5-1 to 2.4.5-17 shall be permitted.

Note 1.— The following figures define the valid sequences of primitives that is possible to invoke during the operation of the FIS application. They show the relationship in time between the service request and the resulting indication, and if applicable, the subsequent response and the resulting confirmation.

Note 2.— All abort primitives may interrupt and terminate any of the normal message sequences outline below.

Note 3.— Primitives are processed in the order in which they are received .

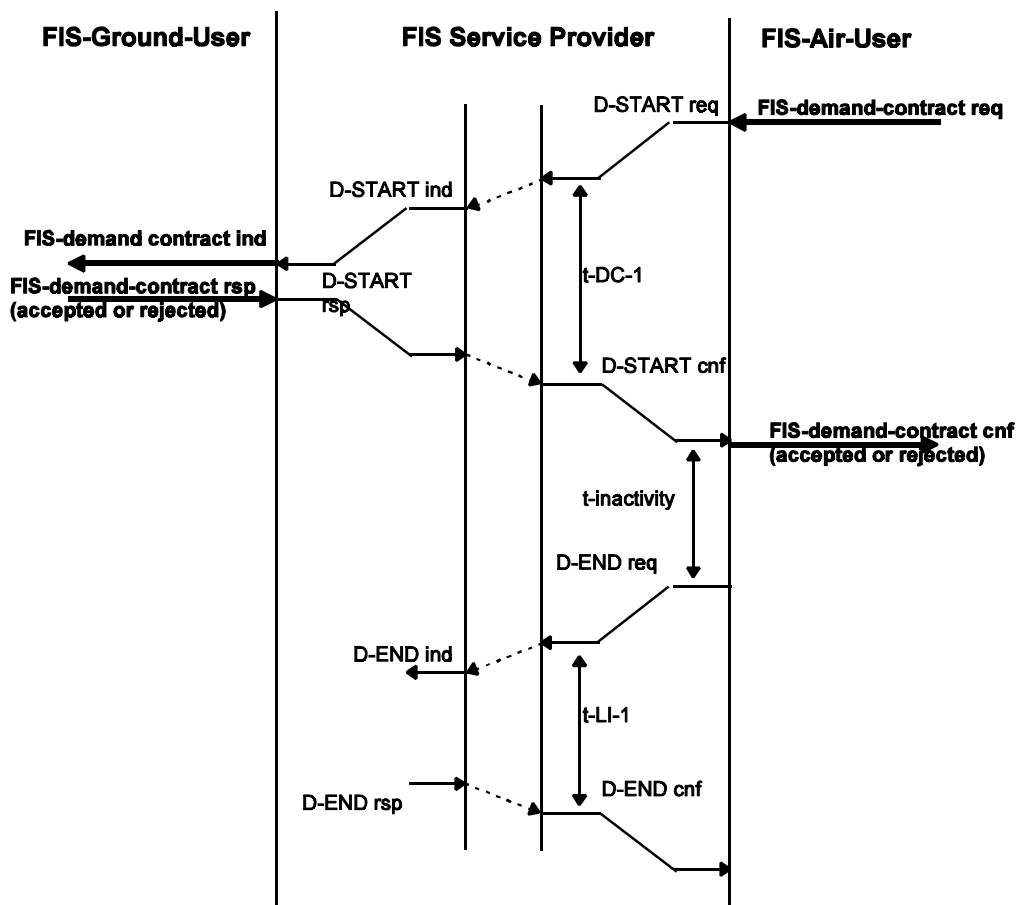


Figure 2.4.5-1. Use of FIS-demand-contract service with no dialogue existing with contract accept or contract reject

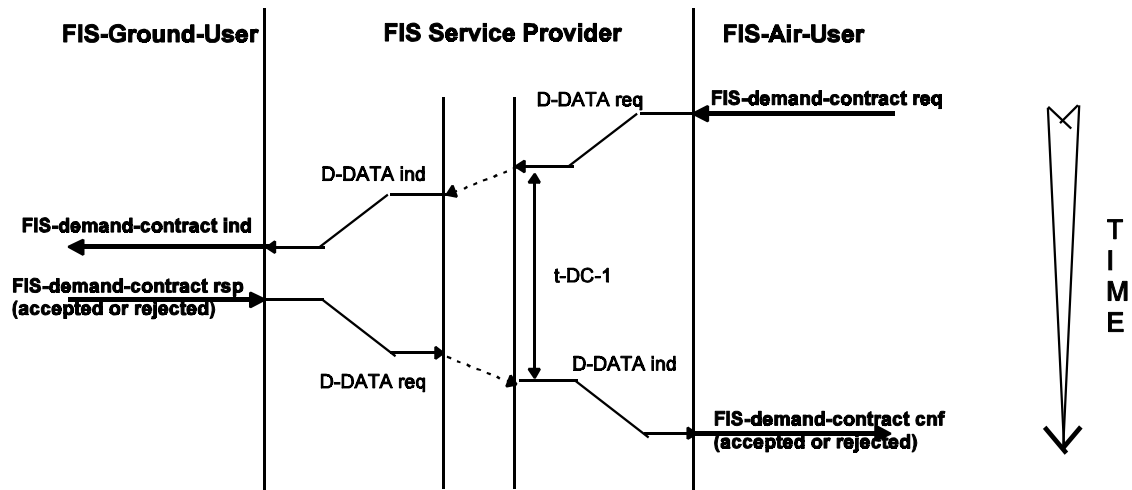


Figure 2.4.5-2. Use of FIS-demand-contract service with dialogue existing with contract accept or reject

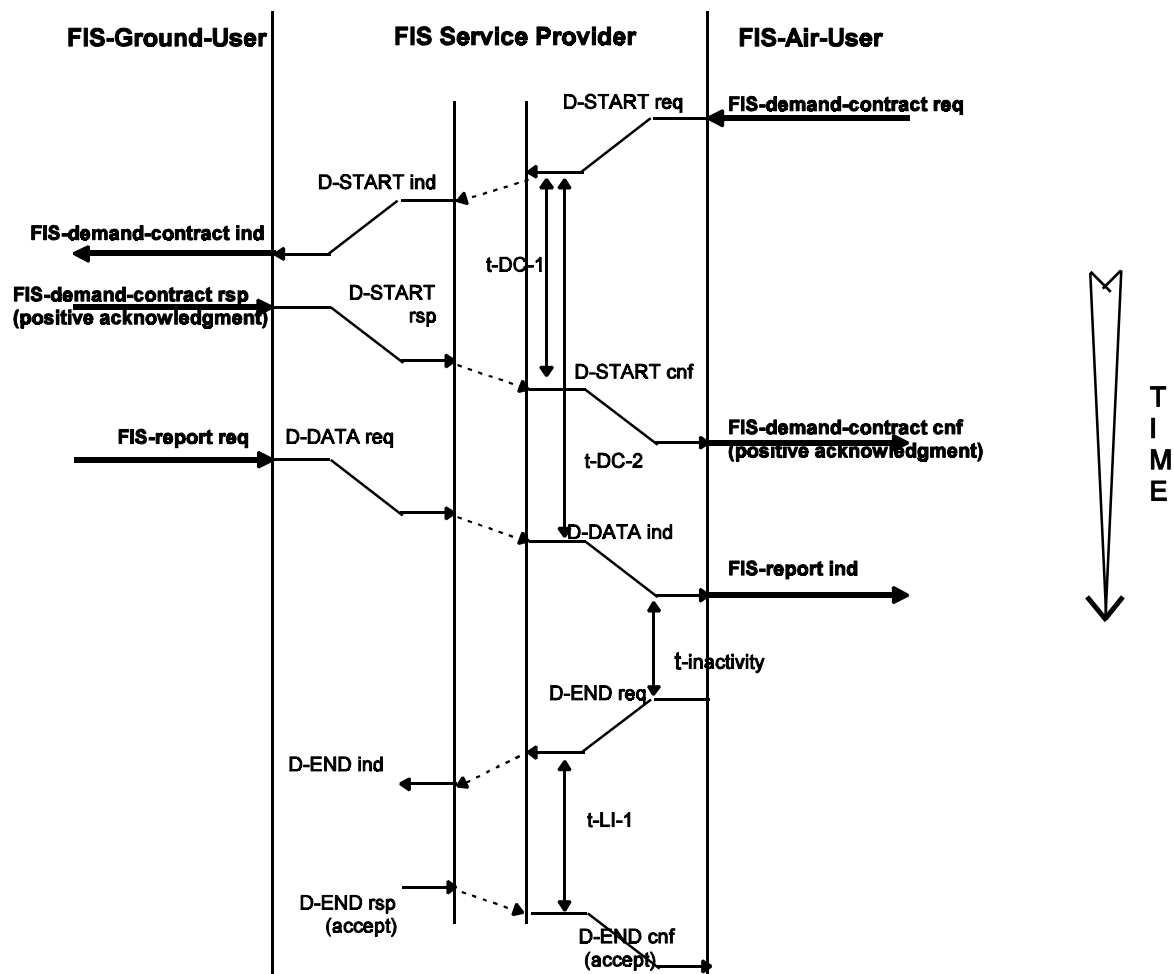


Figure 2.4.5-3. Use of FIS-demand-contract service with no existing dialogue and with a positive acknowledgement

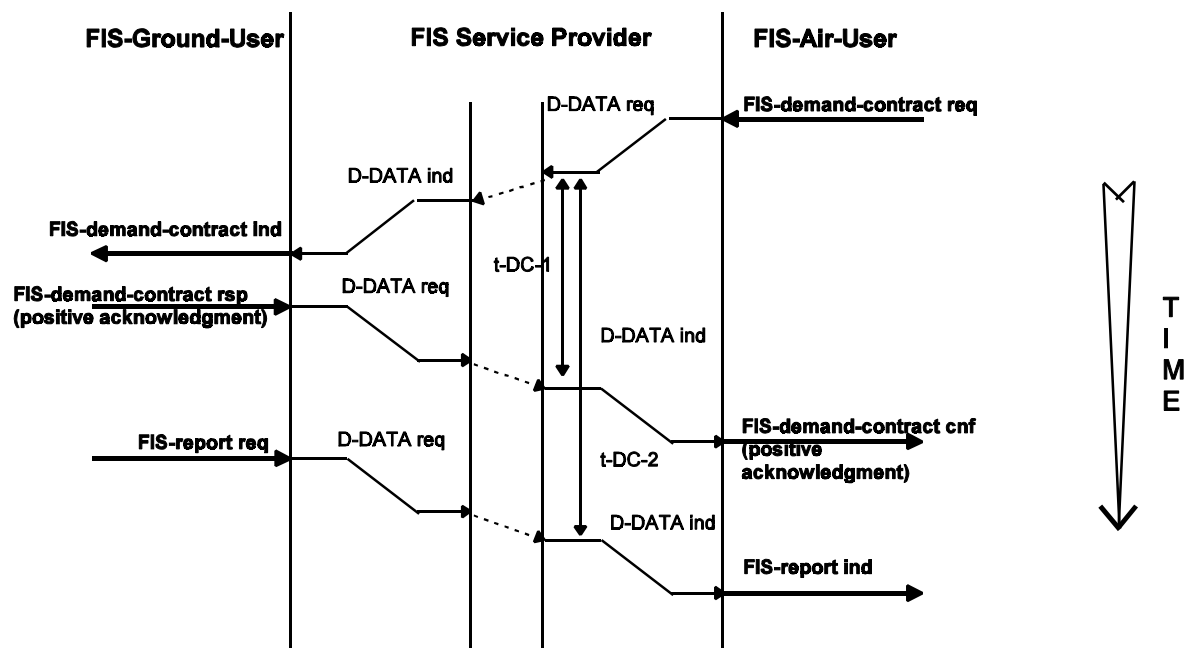


Figure 2.4.5-4. Use of FIS-demand-contract with existing dialogue and positive acknowledgement

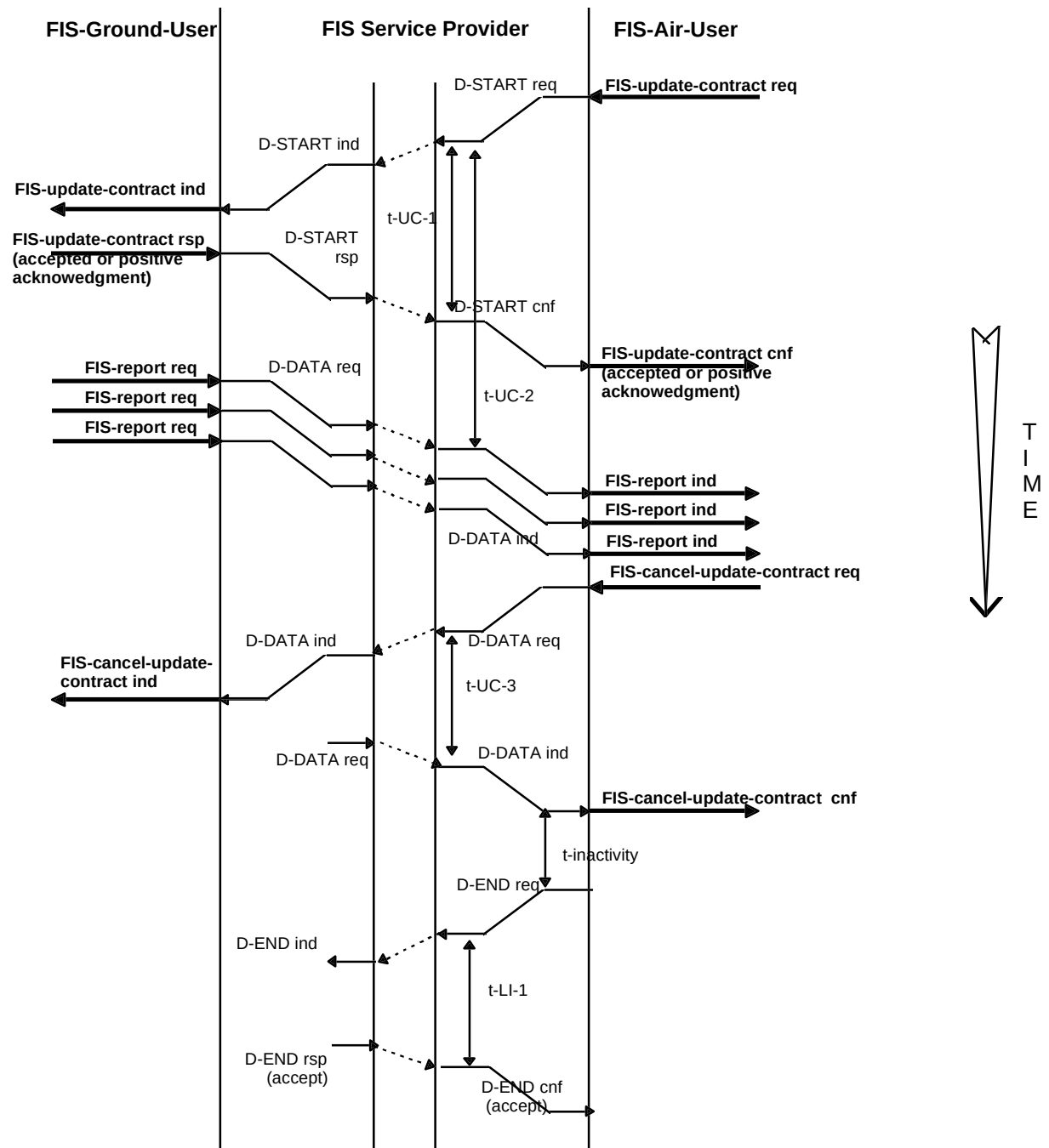


Figure 2.4.5-5. Use of FIS-update-contract service with no existing dialogue and with contract accept and with air-initiated cancellation

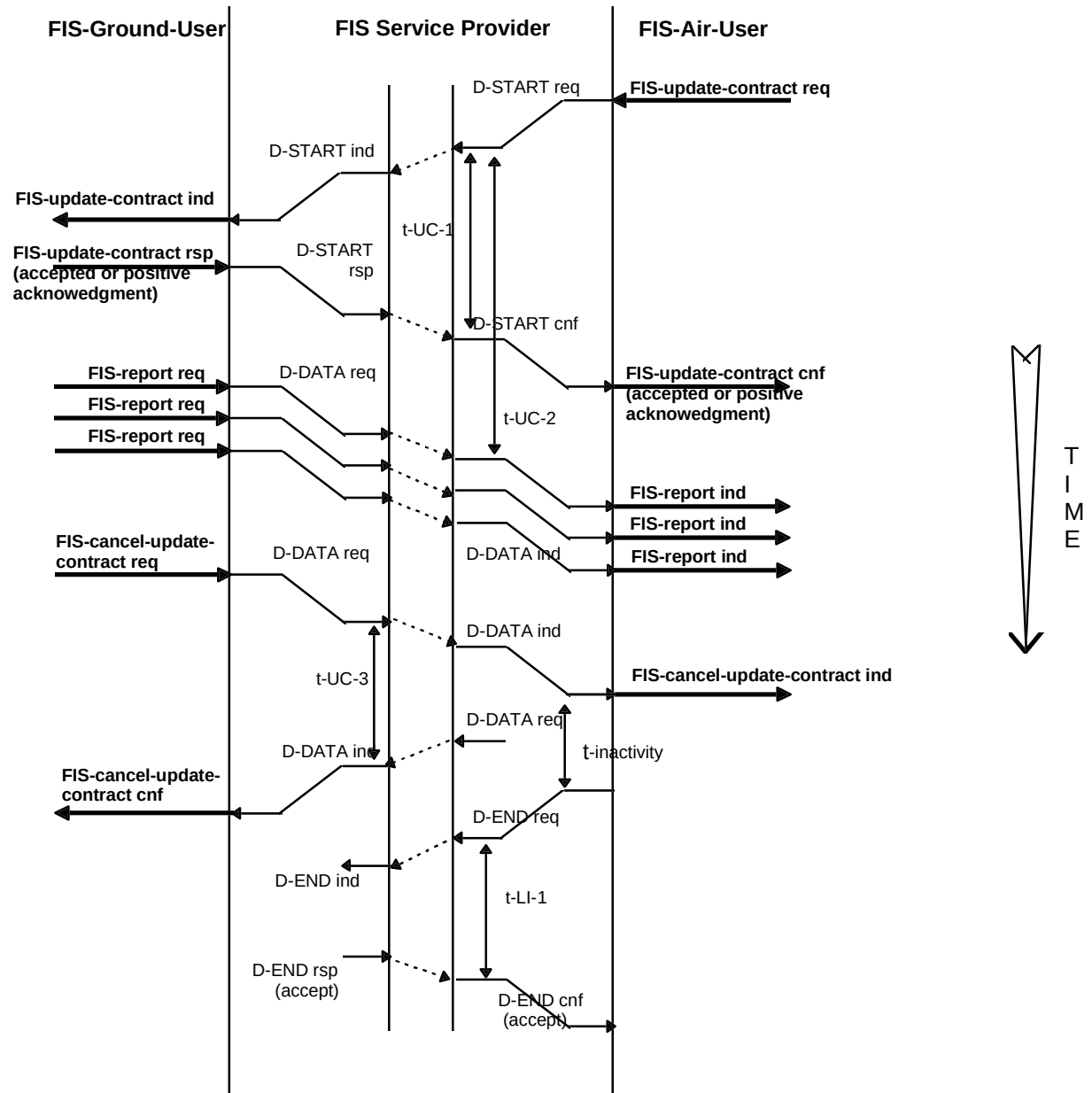


Figure 2.4.5-6. Use of FIS-update-contract service with no existing dialogue and with contract accept and with ground-initiated cancellation

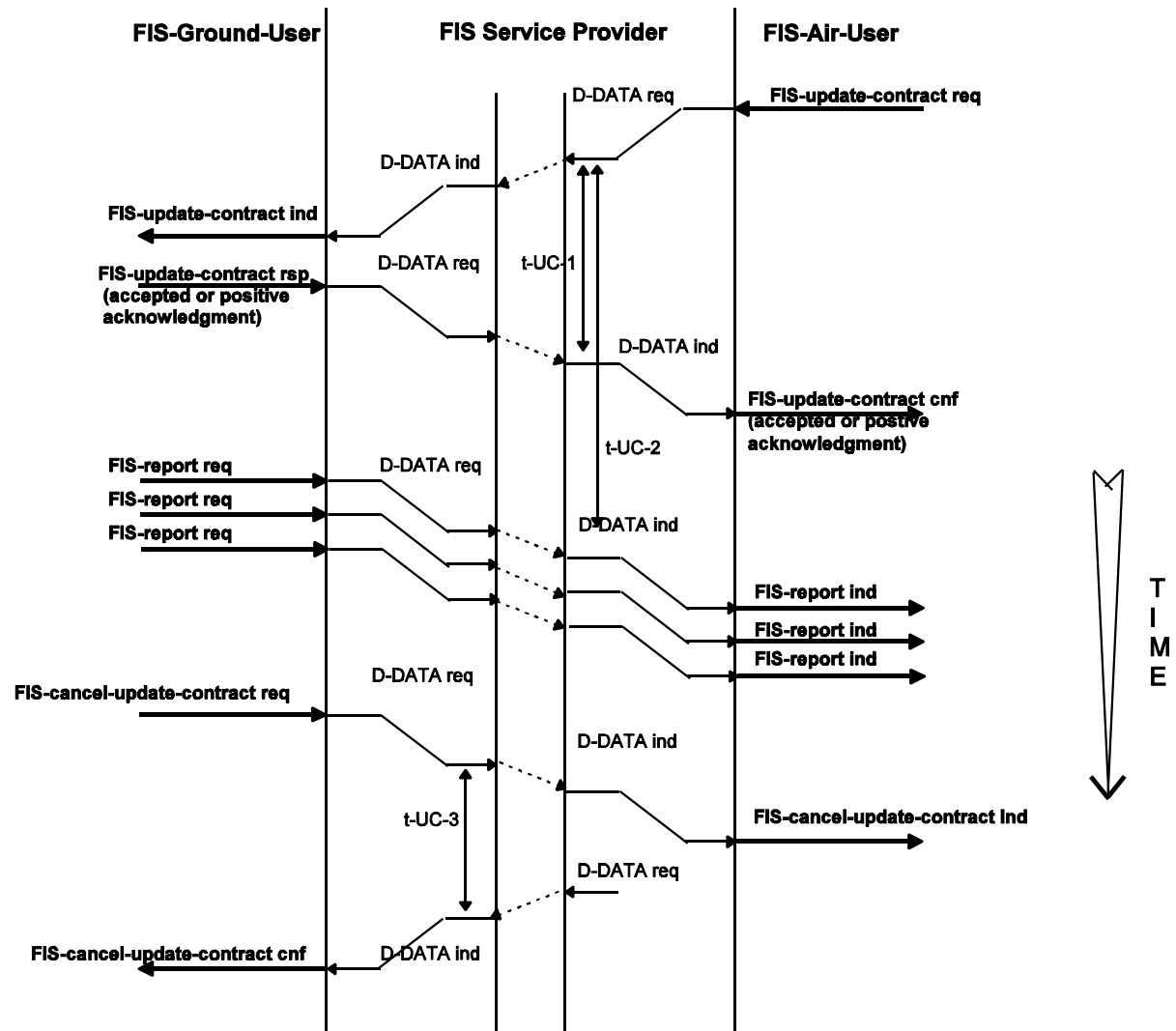


Figure 2.4.5-7. Use of FIS-update-contract service with existing dialogue and with contract accept and with ground-initiated cancellation

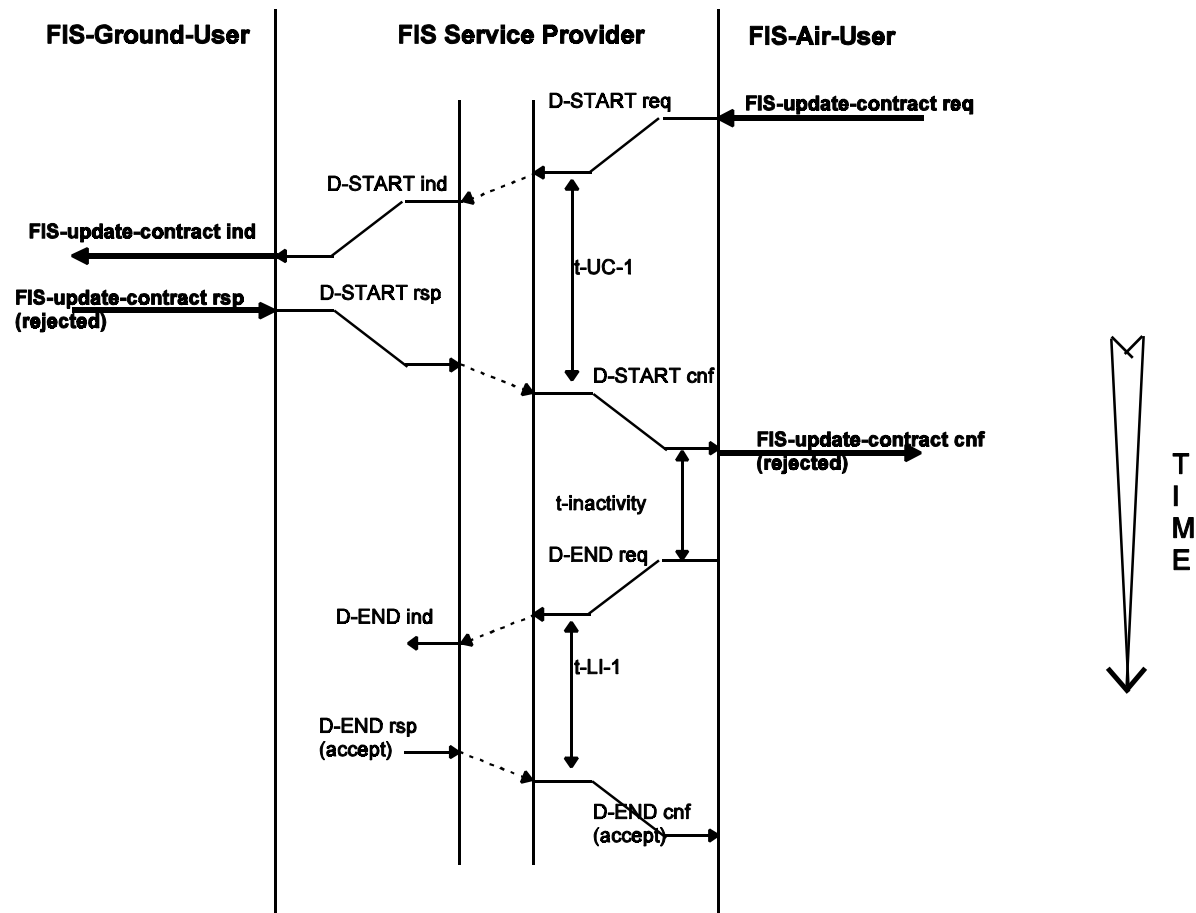


Figure 2.4.5-8. Use of the FIS-update-contract service with no existing dialogue and with contract reject

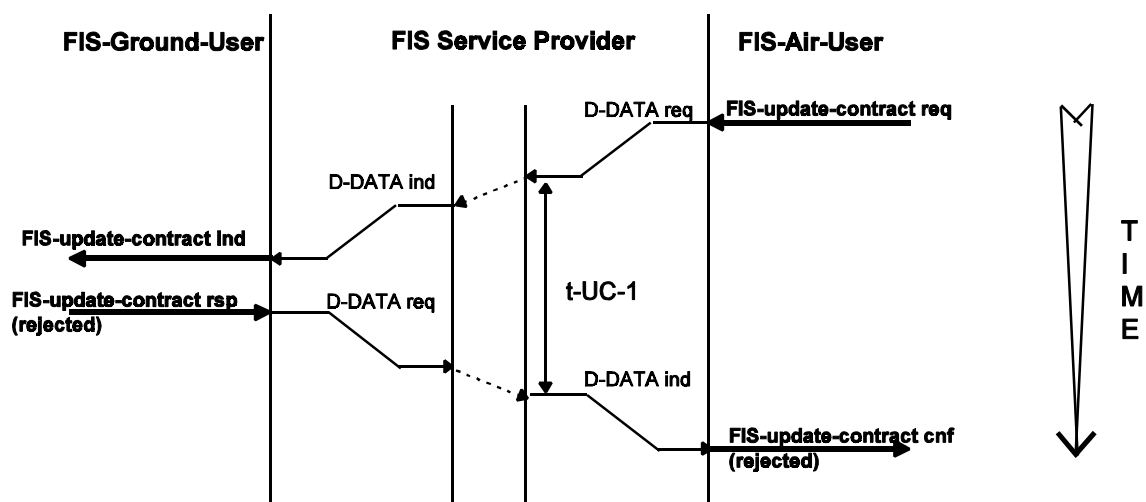


Figure 2.4.5-9. Use of FIS-update-contract service with existing dialogue and with contract reject

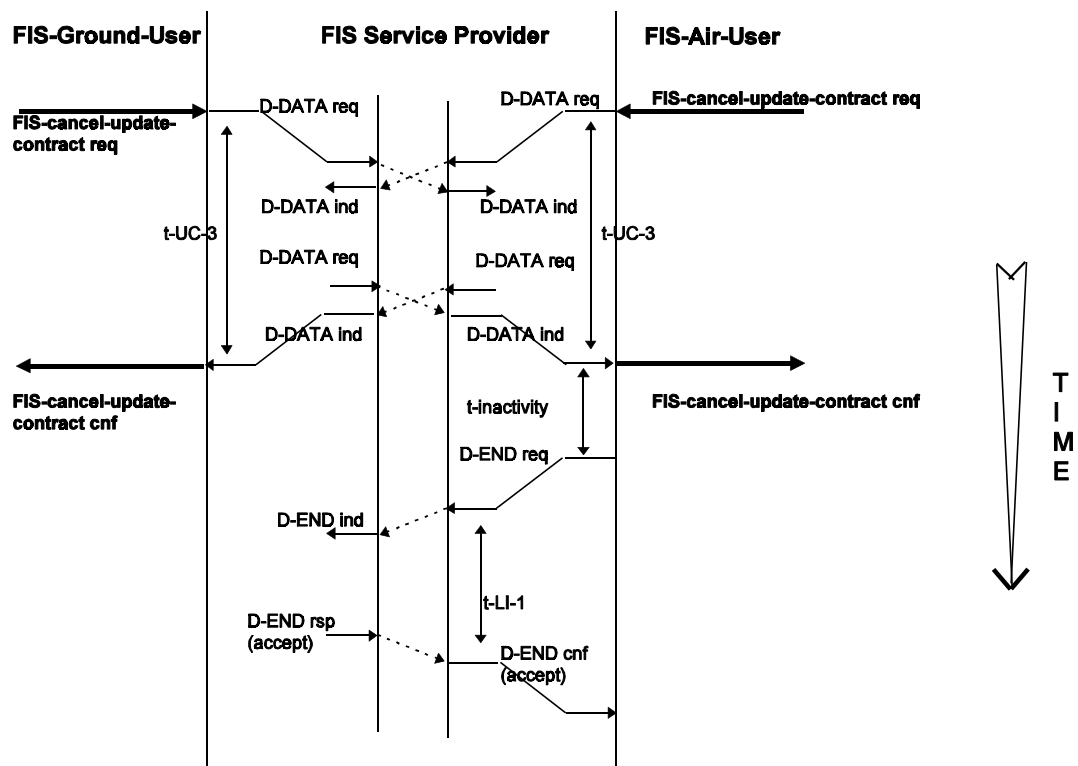


Figure 2.4.5-10. Use of FIS-cancel-update service by both FIS-users without other FIS Contract in place

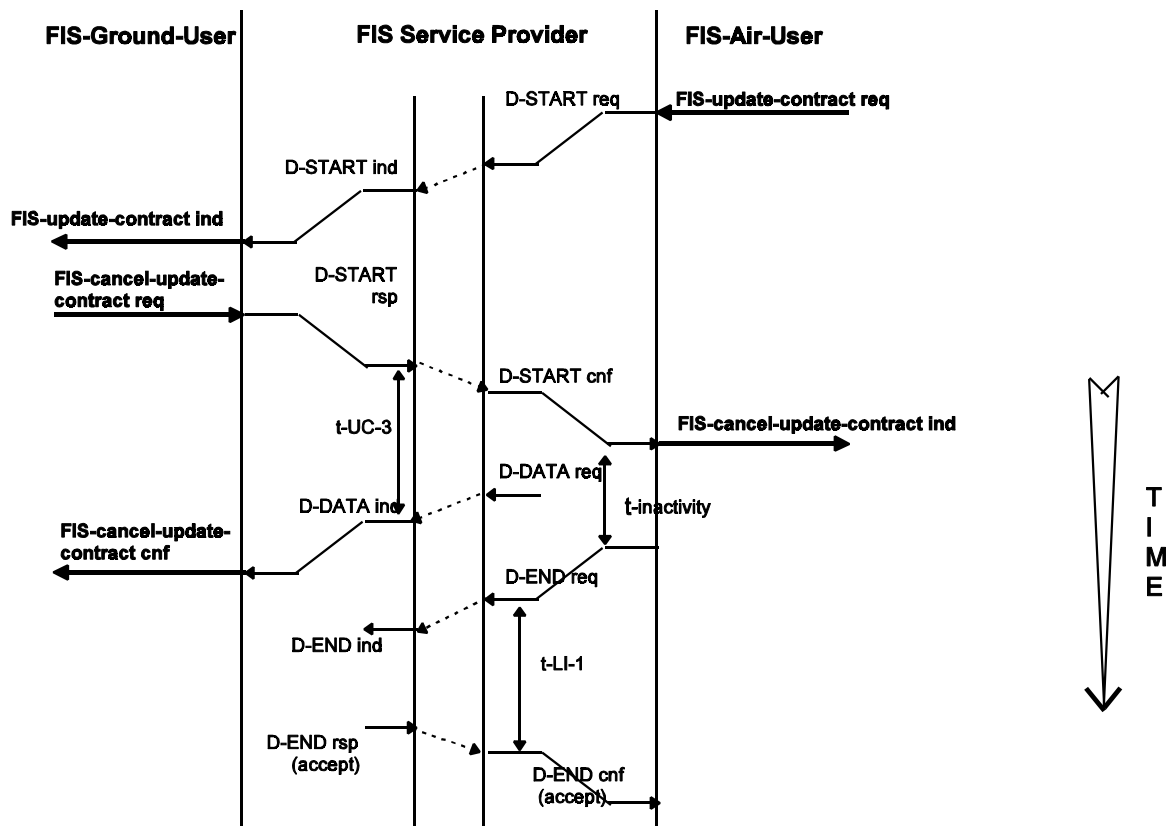
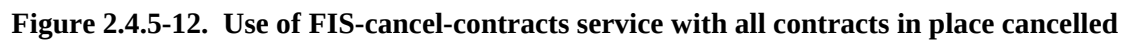


Figure 2.4.5-11. Use of FIS-update-contract service with no existing dialogue and with ground-initiated cancellation during contract establishment phase



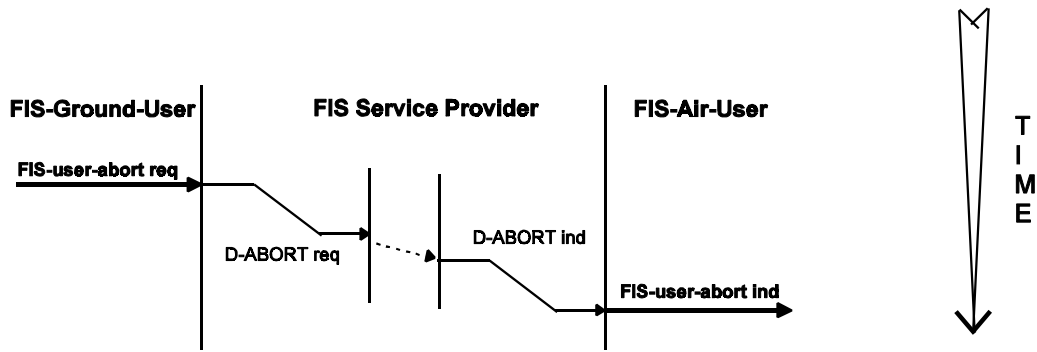


Figure 2.4.5-14. Use of FIS-user-abort service (ground-initiated)

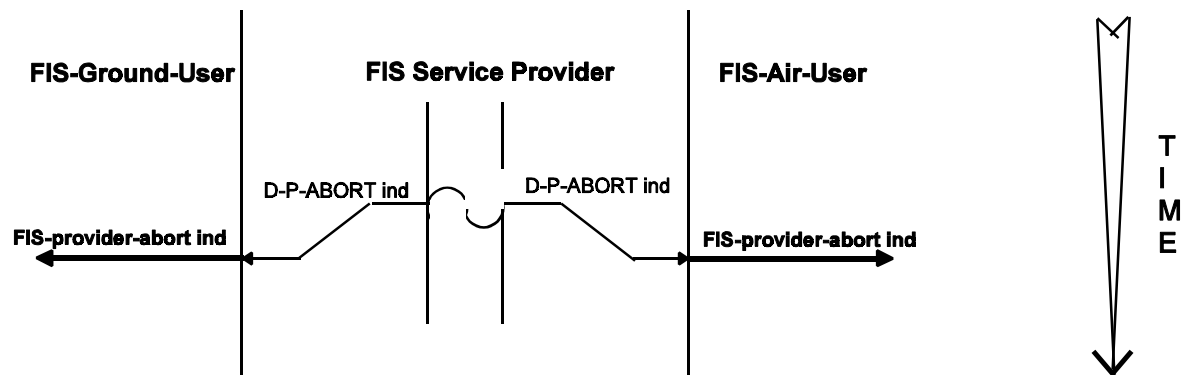


Figure 2.4.5-15. Use of FIS-provider-abort service (dialogue service provider aborts)

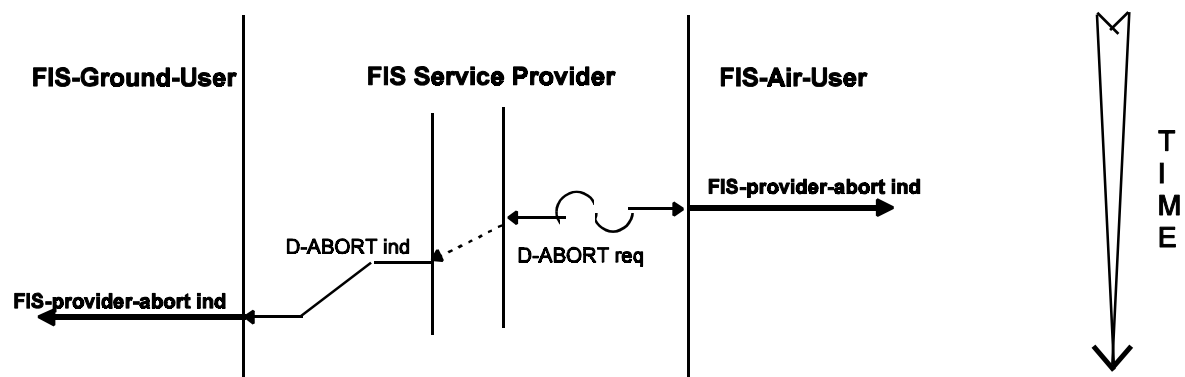


Figure 2.4.5-16. Use of FIS-provider-abort service (FIS-air-ASE aborts)

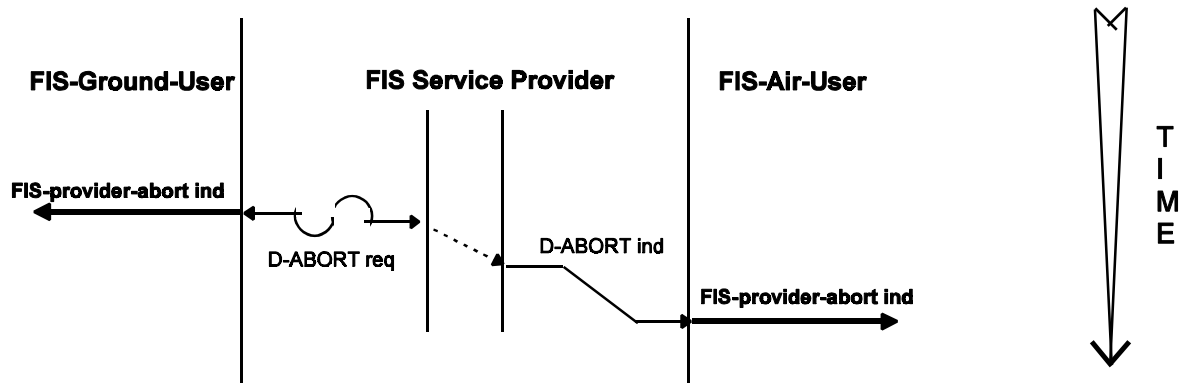


Figure 2.4.5-17. Use of FIS-provider-abort service (FIS-ground-ASE aborts)

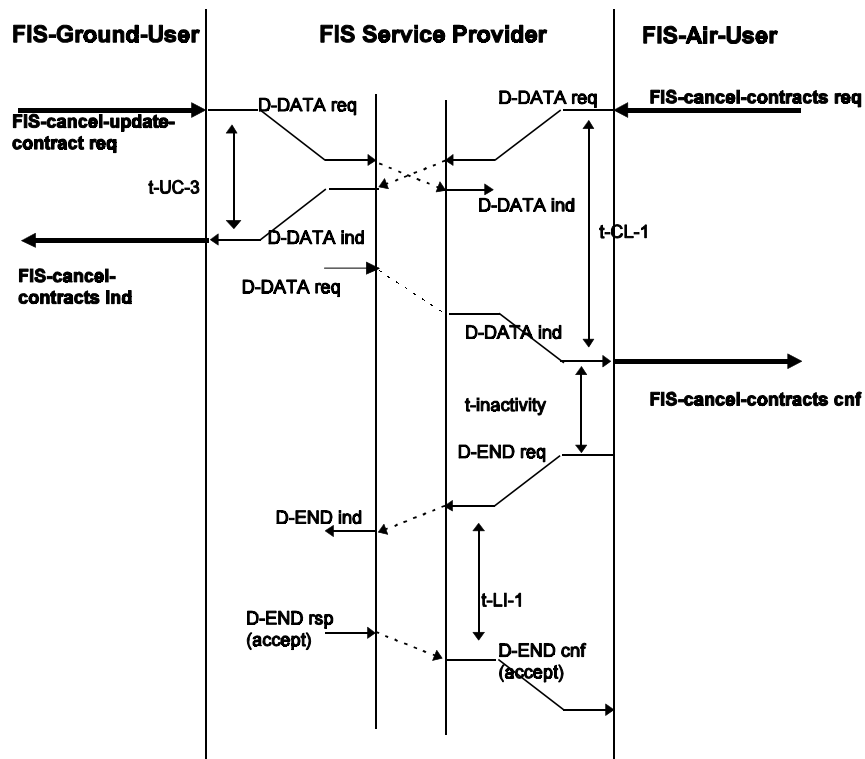


Figure 2.4.5-18. Use of FIS-cancel-update-contract service and FIS-cancel-contracts service simultaneously

2.4.5.2 FIS Service Provider Timers

2.4.5.2.1 The FIS-ASEs shall be capable of detecting when a timer expires.

Note 1.— Table 2.4.5-1 lists the time constraints related to the FIS application. Each time constraint requires a timer to be set in the FIS protocol machine.

Note 2.— If the timer expires before the final event has occurred:

- a) *for all timers but the t-inactivity timer, the FIS-ASEs takes the appropriate action as defined in 2.4.5.3.*
- b) *for the t-inactivity timer, the FIS-air-ASE closes the dialogue by using the D-END service.*

2.4.5.2.2 **Recommendation.** — *The timer values should be as indicated in Table 2.4.5-1.*

Table 2.4.5-1. FIS Service-Provider Timers

FIS Service	Timer Name	Timer Value	Timer Start Event	Timer Stop Event
FIS-demand-contract	t-DC-1	6 min	FIS-demand-contract req	FIS-demand-contract cnf
	t-DC-2	9 min	FIS-demand-contract req	FIS-report ind
FIS-update-contract	t-UC-1	6 min	FIS-update-contract req	FIS-update-contract cnf
	t-UC-2	9 min	FIS-update-contract req	FIS-report ind
FIS-cancel-update-contract	t-UC- 3	9 min	FIS-cancel-update-contract req	FIS-cancel-update-contract cnf
FIS-cancel-contracts	t-CL-1	6 min	FIS-cancel-contracts req	FIS-cancel-contracts cnf
General	t-LI-1	6 min	D-END req	D-END cnf
	t-inactivity	(see note)	last primitive of the last contract received send by the FIS-air-ASE to the FIS-air-user	FIS-demand-contract req or FIS-update-contract req

Note 1.— The t-inactivity timer value is set on configuration basis.

Note 2.— The receipt of FIS-user-abort request, D-ABORT indication and D-P-ABORT indication are also timer stop events.

2.4.5.3 FIS-ASE Protocol Description

2.4.5.3.1 Functional Model

Note 1.— The FIS-ASE is functionally made of 6 modules, as shown in Figure 2.4.5-19:

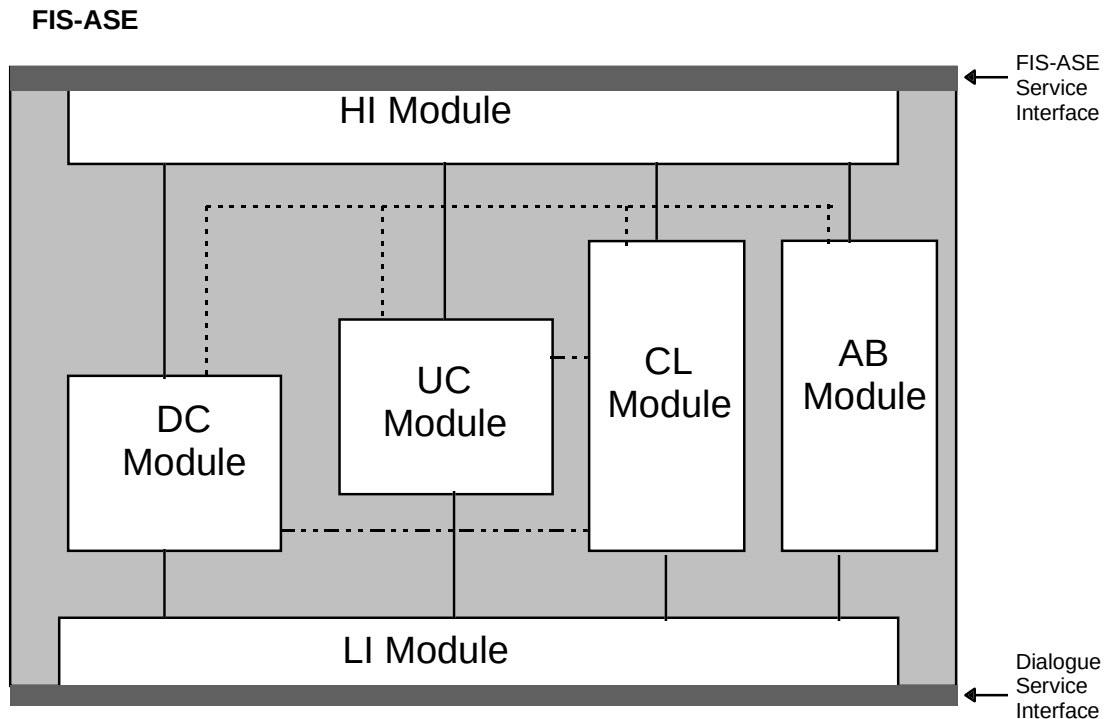


Figure 2.4.5-19. Functional Architecture of the FIS-air-ASE and the FIS-ground-ASE

- a) *the **FIS High Interface module** (HI module). This module interfaces with the ASE-user through the abstract service interface as defined in 2.4.3.*
- b) *the **FIS Demand Contract module** (DC module). This module manages a single FIS Demand Contract.*
- c) *the **FIS Update Contract module** (UC module). This module manages a single FIS Update Contract.*
- d) *the **FIS Cancel contracts module** (CL module). This module processes the termination of all contracts of the same type (i.e. ATIS) still in operation.*
- e) *the **FIS Abort module** (AB module). This module handles aborts in case of unrecoverable error.*

- f) the **FIS Low Interface module** (LI module). This module interfaces the Dialogue Service Provider on behalf of the DC, UC, CL and AB modules. It performs the multiplexing of FIS Contracts on a single dialogue.

Note 2.— This functional architecture allows simplification of the description of the protocol handled by the FIS-ASE. It does not constrain the implementation architecture.

Note 3.— The following subsections describe the actions of the individual modules in both the air and ground ASEs. 2.4.5.5 contains state tables for the individual modules.

Note 4.— The FIS-air-user is considered an active user from the time at which it invokes the first FIS-demand-contract request or FIS-update-contract request until such time that:

- a) the FIS-air-ASE invokes a FIS-demand-contract confirmation (accepted or rejected) and there are no contracts in place,
- b) the FIS-air-ASE invokes a FIS-report indication following a FIS-demand-contract confirmation (positive acknowledgment),
- c) the FIS-air ASE invokes a FIS-update-contract confirmation (rejected) and there is no contract in place,
- d) the FIS-air-ASE invokes a FIS-cancel-update-contract confirmation and there is no contract in place,
- e) the FIS-air-ASE invokes a FIS-cancel-update-contract indication and there is no contract in place,
- f) the FIS-air-ASE invokes a FIS-cancel-contracts confirmation,
- g) the FIS-air-ASE receives a FIS-user-abort request,
- h) the FIS-air-user invokes a FIS-user-abort indication, or
- i) the FIS-air-user receives a FIS-user-abort indication.

Note 5.— The FIS-ground-user is considered an active user from the time at which it receives the first FIS-demand-contract indication or FIS-update-contract indication until such time that:

- a) the FIS-ground-ASE receives a FIS-demand-contract response (accepted or rejected) and there is no contract in place,
- b) the FIS-ground-ASE receives a FIS-report request following a FIS-demand-contract response (positive acknowledgment),
- c) the FIS-ground-ASE receives a FIS-update-contract response (rejected) and there is no contract in place,

- d) *the FIS-ground-ASE invokes a FIS-cancel-update-contract indication and there is no contract in place,*
- e) *the FIS-ground-ASE invokes a FIS-cancel-update-contract confirmation and there is no contract in place,*
- f) *the FIS-ground-ASE invokes a FIS-cancel-contracts indication,*
- g) *the FIS-ground-ASE receives a FIS-user-abort request,*
- h) *the FIS-ground-user invokes a FIS-user-abort indication, or*
- i) *the FIS-ground-user receives a FIS-user-abort indication.*

2.4.5.3.2 In the following sections, if no actions are described for a FIS-service primitive in a particular state, then the invocation of that service primitive shall be prohibited in that state.

2.4.5.3.3 Possible Errors Arising from Receipt of an APDU

2.4.5.3.3.1 Upon receipt of an APDU, if no actions are described for the arrival of that APDU when in a particular state, then exception handling procedures as described in 2.4.5.4.3 shall apply.

2.4.5.3.3.2 If an APDU is not received when one is required, then exception handling procedures as described in 2.4.5.4.2 shall apply.

2.4.5.3.3.3 Upon receipt of an APDU that cannot be decoded, exception handling procedures as described in 2.4.5.4.4 shall apply.

2.4.5.3.4 Ground and Air FIS HI Module

Note.— All statements in 2.4.5.3.4 apply to both the FIS ground HI module and the FIS air HI module.

2.4.5.3.4.1 Upon receipt of a FIS-ASE service request or response primitive, the FIS HI module shall:

- a) if the primitive is a FIS-demand-contract request or a FIS-update-contract request primitive,
 - 1) reject the primitive if the *FISContractNumber* parameter corresponds to an existing FIS contract,
 - 2) else stop timer t-inactivity if set,
- b) if the primitive is not a FIS-demand-contract request or a FIS-update-contract request primitive, reject the primitive if the *FISContractNumber* parameter does not correspond to an existing FIS contract,
- c) otherwise pass the primitive to the relevant DC, UC, CL or AB module, as shown in Tables 2.4.5-2/a and 2.4.5-2/b.

Table 2.4.5-2/a. Request and response primitive to FIS-ASE module mapping - Air ASE

FIS-ASE Service Primitive Name	FIS-ASE Module
FIS-demand-contract req	DC
FIS-update-contract req	UC
FIS-cancel-update-contract req	UC
FIS-user-abort req	AB
FIS-cancel-contracts req	CL

Table 2.4.5-2/b. Request and response primitive to FIS-ASE module mapping - Ground ASE

FIS-ASE Service Primitive Name	FIS-ASE Module
FIS-demand-contract rsp	DC
FIS-update-contract rsp	UC
FIS-cancel-update-contract req	UC
FIS-user-abort req	AB
FIS-report req	UC or DC based on the FISContractNumber

2.4.5.3.4.2 Upon receipt of a request to invoke a service indication or confirmation primitive from one of the modules in the FIS-ASE as shown in Tables 2.4.5-3/a and 2.4.5-3/b, and if the FIS-user is active, the FIS HI module shall do so.

Table 2.4.5-3/a: FIS-ASE module to indication and confirmation primitives mapping - Air ASE

FIS-ASE Module	FIS-ASE Service Primitive Name
DC	FIS-demand-contract cnf
UC	FIS-update-contract cnf

FIS-ASE Module	FIS-ASE Service Primitive Name
UC	FIS-cancel-update-contract ind
UC	FIS-cancel-update-contract cnf
DC, UC	FIS-report ind
AB	FIS-user-abort ind
AB	FIS-provider-abort ind
CL	FIS-cancel-contracts cnf

Table 2.4.5-3/b: FIS-ASE module to indication and confirmation primitives mapping - Ground ASE

FIS-ASE Module	FIS-ASE Service Primitive Name
DC	FIS-demand-contract ind
UC	FIS-update-contract ind
UC	FIS-cancel-update-contract ind
UC	FIS-cancel-update-contract cnf
AB	FIS-user-abort ind
AB	FIS-provider-abort ind
CL	FIS-cancel-contracts ind

2.4.5.3.4.3 The air HI module shall reject requests and responses, apart from FIS-user-abort requests, when the air LI module is in the LI-START-I state or the LI-END-I state.

2.4.5.3.5 Air FIS DC Module

Note.— The states defined for the air FIS DC module are the following:

- a) *DC-A-IDLE,*
- b) *DC-A-PENDING, and*
- c) *DC-A-ACTIVE.*

2.4.5.3.5.1 On initiation, the air FIS DC module shall be in the DC-A-IDLE state.

2.4.5.3.5.2 Upon receipt of a FIS-demand-contract request, then

2.4.5.3.5.2.1 If in the DC-A-IDLE state, the air FIS DC module shall:

- a) start timers t-DC-1 and t-DC-2,
- b) create a FISDownlinkAPDU [FISRequest] APDU with a *demandContract* APDU-element based on the value of the *FISContractNumber* and *FISContractDetails* parameters,
- c) request the LI module to send the APDU to the FIS-ground-ASE identified by the value of the received *ICAOFacilityDesignation* parameter, and with the *ClassOfCommunicationService* parameter value if specified by the user, and
- d) enter the DC-A-PENDING state.

2.4.5.3.5.3 Upon receipt of a [FISAccept] APDU with an *accept* APDU-element, then

2.4.5.3.5.3.1 If in the DC-A-PENDING state, the air FIS DC module shall:

- a) stop timers t-DC-1 and t-DC-2,
- b) if there is no other FIS contract in place, start t-inactivity timer,
- c) request the FIS HI module to invoke a FIS-demand-contract confirmation with the following parameters:
 - 1) the *FISContractNumber* parameter containing the information which has been received as the *ContractNumber* APDU-element,
 - 2) the *Result* parameter, containing the abstract value “accepted”, and
 - 3) the *FISInformation* parameter containing the information which has been received as the *FISReportData* APDU-element, and
- d) enter the DC-A-IDLE state.

2.4.5.3.5.4 Upon receipt of a [FISAccept] APDU containing a *positiveAcknowledgement* APDU-element, then

2.4.5.3.5.4.1 If in the DC-A-PENDING state, the air FIS DC module shall:

- a) stop timer t-DC-1,
- b) request the FIS HI module to invoke a FIS-demand-contract confirmation with the following parameters:
 - 1) the *FISContractNumber* parameter containing the information which has been received as the *ContractNumber* APDU-element, and

- 2) the *Result* parameter, containing the abstract value “positive acknowledgement”, and
 - c) enter the DC-A-ACTIVE state.
- 2.4.5.3.5.5 Upon receipt of a [FISReject] APDU, then
- 2.4.5.3.5.5.1 If in the DC-A-PENDING state, the air FIS DC module shall:
- a) stop timers t-DC-1 and t-DC-2 ,
 - b) if there is no other FIS contract in place, start t-inactivity timer,
 - c) request the FIS HI module to invoke a FIS-demand-contract confirmation with the following parameters:
 - 1) the *FISContractNumber* parameter containing the information which has been received as the *ContractNumber* APDU-element,
 - 2) the *Result* parameter, containing the abstract value “rejected”, and
 - 3) the *RejectReason* parameter containing the information which has been received as the *FISRejectReason* APDU-element, and
 - d) enter the DC-A-IDLE state.
- 2.4.5.3.5.6 Upon receipt of a [FISReport] APDU, then
- 2.4.5.3.5.6.1 If in the DC-A-ACTIVE state, the air FIS DC module shall:
- a) stop timer t-DC-2 ,
 - b) if there is no other FIS contract in place, start t-inactivity timer,
 - c) request the FIS HI module to invoke a FIS-report indication with the following parameters:
 - 1) the *FISContractNumber* parameter containing the information which has been received as the *ContractNumber* APDU-element, and
 - 2) the *FISInformation* parameter, containing the information which has been received as the *FISReportData* APDU-element, and
 - d) enter the DC-A-IDLE state.

2.4.5.3.5.7 Upon receipt of a request from the AB or CL modules to stop operation, then

2.4.5.3.5.7.1 The air FIS DC module shall:

- a) stop any timers, and
- b) enter the DC-A-IDLE state.

2.4.5.3.5.8 Upon expiration of the t-DC-1 timer, then

2.4.5.3.5.8.1 If in the DC-A-PENDING state, the air FIS DC module shall:

- a) stop timer t-DC-2,
- b) request the AB module to abort with reason “timer expiration”, and
- c) enter the DC-A-IDLE state.

2.4.5.3.5.9 Upon expiration of the t-DC-2 timer, then

2.4.5.3.5.9.1 If in the DC-A-ACTIVE state, the air FIS DC module shall:

- a) request the AB module to abort with reason “timer expiration”, and
- b) enter the DC-A-IDLE state.

2.4.5.3.6 Ground FIS DC Module

Note.— The states defined for the ground FIS DC module are the following:

- a) *DC-G-IDLE*,
- b) *DC-G-PENDING*, and
- c) *DC-G-ACTIVE*.

2.4.5.3.6.1 On initiation, the ground FIS DC module shall be in the DC-G-IDLE state.

2.4.5.3.6.2 Upon receipt of a [FISRequest] APDU with a *demandContract* APDU-element, then

2.4.5.3.6.2.1 If in the DC-G-IDLE state, the ground FIS DC module shall:

- a) request the FIS HI module to invoke a FIS-demand-contract indication with the following parameters:
 - 1) the *FISContractNumber* parameter containing the information which has been received as the *ContractNumber* APDU-element, and

- 2) the *FISContractDetails* parameter, containing the information which has been received as the *FISRequestData* APDU-element, and
 - b) enter the DC-G-PENDING state.
- 2.4.5.3.6.3 Upon receipt of a FIS-demand-contract response with the *Result* parameter containing the abstract value “accepted”, then
- 2.4.5.3.6.3.1 If in the DC-G-PENDING state, the ground FIS DC module shall:
- a) create a FISUplinkAPDU [FISAccept] APDU with an *accept* APDU-element based on the value of the *FISContractNumber* and *FISInformation* parameters,
 - b) request the LI module to send the APDU to the FIS-air-ASE, and
 - c) enter the DC-G-IDLE state.
- 2.4.5.3.6.4 Upon receipt of a FIS-demand-contract response with the *Result* parameter containing the abstract value “positive acknowledgement”, then:
- 2.4.5.3.6.4.1 If in the DC-G-PENDING state, the ground FIS DC module shall:
- a) create a FISUplinkAPDU [FISAccept] APDU with a *positiveAcknowledgement* APDU-element based on the value of the *FISContractNumber* parameter,
 - b) request the LI module to send the APDU to the FIS-air-ASE, and
 - c) enter the DC-G-ACTIVE state.
- 2.4.5.3.6.5 Upon receipt of a FIS-demand-contract response with the *Result* parameter containing the abstract value “rejected”, then:
- 2.4.5.3.6.5.1 If in the DC-G-PENDING state, the ground FIS DC module shall:
- a) create a FISUplinkAPDU [FISReject] APDU with an *otherReasons* APDU-element based on the value of the *FISContractNumber* and the *RejectReason* parameters,
 - b) request the LI module to send the APDU to the FIS-air-ASE, and
 - c) enter the DC-G-IDLE state.
- 2.4.5.3.6.6 Upon receipt of a FIS-report request, then
- 2.4.5.3.6.6.1 If in the DC-G-ACTIVE state, the ground FIS DC module shall:
- a) create a FISUplinkAPDU [FISReport] APDU based on the value of the *FISContractNumber* and the *FISInformation* parameters,
 - b) request the LI module to send the APDU to the FIS-air-ASE, and

- c) enter the DC-G-IDLE state.

2.4.5.3.6.7 Upon receipt of a request from the AB or CL modules to stop operation, then

2.4.5.3.6.7.1 The ground FIS DC module shall enter the DC-G-IDLE state.

2.4.5.3.7 Air FIS UC Module

Note.— The states defined for the air UC module are the following:

- a) *UC-A-IDLE*,
- b) *UC-A-PENDING*,
- c) *UC-A-ACTIVE*, and
- d) *UC-A-CANCEL*

2.4.5.3.7.1 On initiation, the air FIS UC module shall be in the UC-A-IDLE state.

Note.— The air FIS UC module has a boolean variable named CANCELFROMPENDING.

2.4.5.3.7.2 On initiation, CANCELFROMPENDING shall be set to FALSE.

2.4.5.3.7.3 Upon receipt of a FIS-update-contract request, then

2.4.5.3.7.3.1 If in the UC-A-IDLE state, the air FIS UC module shall:

- a) start timers t-UC-1 and t-UC-2,
- b) create a FISDownlinkAPDU [FISRequest] APDU with an *updateContract* APDU-element based on the value of the *FISContractNumber* and *FISContractDetails* parameters,
- c) request the LI module to send the APDU to the FIS-ground-ASE identified by the value of the received *ICAOFacilityDesignation* parameter, with the *ClassOfCommunicationService* parameter value if specified by the user, and
- d) enter the UC-A-PENDING state.

2.4.5.3.7.4 Upon receipt of a [FISAccept] APDU containing an *accept* APDU-element, then

2.4.5.3.7.4.1 If in the UC-A-PENDING state, the air FIS UC module shall:

- a) stop timers t-UC-1 and t-UC-2,
- b) request the FIS HI module to invoke a FIS-update-contract confirmation with the following parameters:

- 1) the *FISContractNumber* parameter containing the information which has been received as the *ContractNumber* APDU-element,
 - 2) the *Result* parameter, containing the abstract value “accepted”, and
 - 3) the *FIS Information* parameter, containing the information which has been received as the *FISReportData* APDU-element, and
- c) enter the UC-A-ACTIVE state.

2.4.5.3.7.4.2 If in the UC-A-CANCEL state and the CANCELFROMPENDING is set to TRUE, the air FIS UC module shall:

- a) stop timers t-UC-1 and t-UC-2,
- b) set CANCELFROMPENDING to FALSE, and
- c) remain in the UC-A-CANCEL state.

2.4.5.3.7.5 Upon receipt of a [FISAccept] APDU containing a *positiveAcknowledgement* APDU-element, then

2.4.5.3.7.5.1 If in the UC-A-PENDING state, the air FIS UC module shall:

- a) stop timer t-UC-1,
- b) request the FIS HI module to invoke a FIS-update-contract confirmation with the following parameters:
 - 1) the *FISContractNumber* parameter containing the information which has been received as the *ContractNumber* APDU-element, and
 - 2) the *Result* parameter, containing the abstract value “positive acknowledgement”, and
- c) enter the UC-A-ACTIVE state.

2.4.5.3.7.5.2 If in the UC-A-CANCEL state and the CANCELFROMPENDING is set to TRUE, the air FIS UC module shall:

- a) stop timers t-UC-1 and t-UC-2,
- b) set CANCELFROMPENDING to FALSE, and
- c) remain in the UC-A-CANCEL state.

2.4.5.3.7.6 Upon receipt of a [FISReject] APDU, then:

2.4.5.3.7.6.1 If in the UC-A-PENDING state, the air FIS UC module shall:

- a) stop timers t-UC-1 and t-UC-2,
- b) if there is no other FIS contract in place, start t-inactivity timer,
- c) request the FIS HI module to invoke a FIS-update-contract confirmation with the following parameters:
 - 1) the *FISContractNumber* parameter containing the information which has been received as the *ContractNumber* APDU-element,
 - 2) the *Result* parameter, containing the abstract value “rejected”,
 - 3) the *RejectReason* parameter containing:
 - i) the information which has been received as the *FISRejectReason* APDU-element, if the *FISRejectData* contains the *otherReasons* element, or
 - ii) the abstract value “update contract function not supported by the FIS-ground-user”, if the *FISRejectData* APDU-element contains the *updateFISContractNotSupported* or *updateFISContractNotSupportedWithReport* choice, and
 - 4) the *FISInformation* parameter containing the information which has been received as the *FISReportData* element, if the *FISRejectData* APDU-element contains the *updateFISContractNotSupportedWithReport* choice, and
- d) enter the UC-A-IDLE state.

2.4.5.3.7.6.2 If in the UC-A-CANCEL state and the CANCELFROMPENDING is set to TRUE, the air FIS UC module shall:

- a) stop timers t-UC-1 and t-UC-2,
- b) set CANCELFROMPENDING to FALSE,
- c) request the FIS HI module to invoke a FIS-cancel-update-contract confirmation with the *FISContractNumber* parameter containing the information which has been received as the *ContractNumber* APDU-element, and
- d) enter the UC-A-IDLE state.

2.4.5.3.7.7 Upon receipt of a [FISReport] APDU, then

2.4.5.3.7.7.1 If in the UC-A-ACTIVE state, the air FIS UC module shall:

- a) if timer t-UC-2 is set, stop timer t-UC-2 ,
- b) request the FIS HI module to invoke a FIS-report indication with the following parameters:
 - 1) the *FISContractNumber* parameter containing the information which has been received as the *ContractNumber* APDU-element, and
 - 2) the *FISInformation* parameter, containing the information which has been received as the *FISReportData* APDU-element, and
- c) remain in the UC-A-ACTIVE state.

2.4.5.3.7.7.2 If in the UC-A-CANCEL state and the CANCELFROMPENDING is set to FALSE, the air FIS UC module shall remain in the UC-A-CANCEL state.

2.4.5.3.7.8 Upon receipt of a FIS-cancel-update-contract request, then

2.4.5.3.7.8.1 If in the UC-A-ACTIVE state, the air FIS UC module shall:

- a) if timer t-UC-2 set, stop timer t-UC-2 ,
- b) create a FISDownlinkAPDU [FISCancelUpdateContract] APDU based on the value of the received *FISContractNumber* parameter,
- c) request the LI module to send the APDU to the FIS-ground-ASE,
- d) start the t-UC-3 timer, and
- e) enter the UC-A-CANCEL state.

2.4.5.3.7.8.2 If in the UC-A-PENDING state and a dialogue is fully established, the air FIS UC module shall:

- a) if timer t-UC-2 set, stop timer t-UC-2 ,
- b) create a FISDownlinkAPDU [FISCancelUpdateContract] APDU based on the value of the received *FISContractNumber* parameter,
- c) request the LI module to send the APDU to the FIS-ground-ASE,
- d) start the t-UC-3 timer,
- e) set CANCELFROMPENDING to TRUE, and

- f) enter the UC-A-CANCEL state.

2.4.5.3.7.9 Upon receipt of a [FISCancelUpdateContract] APDU, then

2.4.5.3.7.9.1 If in the UC-A-ACTIVE state, the air FIS UC module shall:

- a) if timer t-UC-2 is set, stop timer t-UC-2 ,
- b) request the FIS HI module to invoke a FIS-cancel-update-contract indication with the *FISContractNumber* parameter containing the information which has been received as the *ContractNumber* APDU-element,
- c) create a FISDownlinkAPDU [FISCancelUpdateAccept] APDU based on the value of the *ContractNumber* APDU-element as received in the APDU,
- d) request the LI module to send the APDU to the FIS-ground-ASE,
- e) if there is no other FIS contract in place, start the t-inactivity timer, and
- f) enter the UC-A-IDLE state.

2.4.5.3.7.9.2 If in the UC-A-PENDING state, the air FIS UC module shall:

- a) stop timers t-UC-1 and t-UC-2 ,
- b) request the FIS HI module to invoke a FIS-cancel-update-contract indication with the *FIS ContractNumber* parameter containing the information which has been received as the *ContractNumber* APDU-element,
- c) create a FISDownlinkAPDU [FISCancelUpdateAccept] APDU based on the value of the *ContractNumber* APDU-element as received in the APDU,
- d) request the LI module to send the APDU to the FIS-ground-ASE,
- e) if there is no other FIS contract in place, start the t-inactivity timer, and
- f) enter the UC-A-IDLE state.

2.4.5.3.7.9.3 If in the UC-A-CANCEL state (i.e. collision of cancel-update-contract requests), the air FIS UC module shall:

- a) create a FISDownlinkAPDU [FISCancelUpdateAccept] APDU based on the value of the *ContractNumber* APDU-element as received in the APDU,
- b) request the LI module to send the APDU to the FIS-ground-ASE, and
- c) remain in the UC-A-CANCEL state.

2.4.5.3.7.10 Upon receipt of a [FISCancelUpdateAccept] APDU, then

2.4.5.3.7.10.1 If in the UC-A-CANCEL state, the air FIS UC module shall:

- a) stop the t-UC-3 timer,
- b) request the FIS HI module to invoke a FIS-cancel-update-contract confirmation with the *FISContractNumber* parameter containing the information which has been received as the *ContractNumber* APDU-element,
- c) if there is no other FIS contract in place, start the t-inactivity timer, and
- d) enter the UC-A-IDLE state.

2.4.5.3.7.11 Upon receipt on a request from the AB or CL modules to stop operation, then

2.4.5.3.7.11.1 The air FIS UC module shall:

- a) stop any timers, and
- b) enter the UC-A-IDLE state.

2.4.5.3.7.12 Upon expiration of the t-UC-1 timer, then:

2.4.5.3.7.12.1 If in UC-A-PENDING state, the air FIS UC module shall:

- a) stop timer t-UC-2,
- b) request the AB module to abort with reason “timer expiration”, and
- c) enter the UC-A-IDLE state.

2.4.5.3.7.13 Upon expiration of the t-UC-2 timer, then:

2.4.5.3.7.13.1 If in UC-A-ACTIVE state, the air FIS UC module shall:

- a) request the AB module to abort with reason “timer expiration”, and
- b) enter the UC-A-IDLE state.

2.4.5.3.7.14 Upon expiration of the t-UC-3 timer, then:

2.4.5.3.7.14.1 If in UC-A-CANCEL state, the air FIS UC module shall:

- a) request the AB module to abort with reason “timer expiration”, and
- b) enter the UC-A-IDLE state.

2.4.5.3.8 Ground FIS UC Module

Note.— The states defined for the ground FIS UC module are the following:

- a) *UC-G-IDLE*,
- b) *UC-G-PENDING*,
- c) *UC-G-ACTIVE*, and
- d) *UC-G-CANCEL*.

2.4.5.3.8.1 On initiation, the ground FIS UC module shall be in the *UC-G-IDLE* state.

2.4.5.3.8.2 Upon receipt of a [FISRequest] APDU containing an *updateContract* APDU-element, then

2.4.5.3.8.2.1 If in the *UC-G-IDLE* state, the ground FIS UC module shall:

- a) request the FIS HI module to invoke a FIS-update-contract indication with the following parameters:
 - 1) the *FISContractNumber* parameter containing the information which has been received as the *ContractNumber* APDU-element, and
 - 2) the *FISContractDetails* parameter, containing the information which has been received as the *FISRequestData* APDU-element, and
- b) enter the *UC-G-PENDING* state.

2.4.5.3.8.3 Upon receipt of a FIS-update-contract response with a *Result* parameter containing the abstract value “accepted”, then

2.4.5.3.8.3.1 If in the *UC-G-PENDING* state, the ground FIS UC module shall:

- a) create a FISUplinkAPDU [FISAccept] APDU with an *accept* APDU-element based on the value of the *FISContractNumber* and *FISInformation* parameters,
- b) request the LI module to send the APDU to the FIS-air-ASE, and
- c) enter the *UC-G-ACTIVE* state.

2.4.5.3.8.4 Upon receipt of a FIS-update-contract response with a *Result* parameter containing the abstract value “positive acknowledgement”, then

2.4.5.3.8.4.1 If in the *UC-G-PENDING* state, the ground FIS UC module shall:

- a) create a FISUplinkAPDU [FISAccept] APDU with a *positiveAcknowledgement* APDU-element based on the value of the *FISContractNumber* parameter,

- b) request the LI module to send the APDU to the FIS-air-ASE, and
- c) enter the UC-G-ACTIVE state.

2.4.5.3.8.5 Upon receipt of a FIS-update-contract response with a *Result* parameter containing the abstract value “rejected”, then

2.4.5.3.8.5.1 If in the UC-G-PENDING state, the ground FIS UC module shall:

- a) create a FISUplinkAPDU [FISReject] APDU based on the value of:
 - 1) the *FISContractNumber* parameter and,
 - 2) the *FISInformation* parameter if provided and if the *RejectReason* parameter contains the abstract value “update contract function not supported”, or
 - 3) the *RejectReason* parameter if the *RejectReason* parameter does not contain the abstract value “update contract function not supported”,
- b) request the LI module to send the APDU to the FIS-air-ASE, and
- c) enter the UC-G-IDLE state.

2.4.5.3.8.6 Upon receipt of a FIS-report request, then

2.4.5.3.8.6.1 If in the UC-G-ACTIVE state, the ground FIS UC module shall:

- a) create a FISUplinkAPDU [FISReport] APDU based on the value of the *FISContractNumber* and *FISInformation* parameters,
- b) request the LI module to send the APDU to the FIS-air-ASE, and
- c) remain in the UC-G-ACTIVE state.

2.4.5.3.8.7 Upon receipt of a FIS-cancel-update-contract request, then

2.4.5.3.8.7.1 If in the UC-G-ACTIVE state, the ground FIS UC module shall:

- a) create a FISUplinkAPDU [FISCancelUpdateContract] APDU based on the value of the *FISContractNumber* parameter,
- b) start timer t-UC-3,
- c) request the LI module to send the APDU to the FIS-air-ASE, and
- d) enter the UC-G-CANCEL state.

2.4.5.3.8.7.2 If in the UC-G-PENDING state, the ground FIS UC module shall:

- a) create a FISUplinkAPDU [FISCancelUpdateContract] APDU based on the value of the *FISContractNumber* parameter,
- b) start timer t-UC-3,
- c) request the LI module to send the APDU to the FIS-air-ASE, and
- d) enter the UC-G-CANCEL state.

2.4.5.3.8.8 Upon receipt of a [FISCancelUpdateContract] APDU, then

2.4.5.3.8.8.1 If in the UC-G-ACTIVE state, the ground FIS UC module shall:

- a) request the FIS HI module to invoke a FIS-cancel-update-contract indication with the *ContractNumber* APDU-element as received in the APDU,
- b) create a FISUplinkAPDU [FISCancelUpdateAccept] APDU based on the value of the *ContractNumber* APDU-element as received in the APDU,
- c) request the LI module to send the APDU to the FIS-air-ASE, and
- d) enter the UC-G-IDLE state.

2.4.5.3.8.8.2 If in the UC-G-PENDING state, the ground FIS UC module shall:

- a) request the FIS HI module to invoke a FIS-cancel-update-contract indication with the *ContractNumber* APDU-element as received in the APDU,
- b) create a FISUplinkAPDU [FISCancelUpdateAccept] APDU based on the value of the *ContractNumber* APDU-element as received in the APDU,
- c) request the LI module to send the APDU to the FIS-air-ASE, and
- d) enter the UC-G-IDLE state.

2.4.5.3.8.8.3 If in the UC-G-CANCEL state (i.e. collision of cancel-contract-update requests), the ground FIS UC module shall:

- a) create a FISUplinkAPDU [FISCancelUpdateAccept] APDU based on the *ContractNumber* APDU-element as received in the APDU,
- b) request the LI module to send the APDU to the FIS-air-ASE, and
- c) enter the UC-G-CANCEL state.

- 2.4.5.3.8.9 Upon receipt of a [FISCancelUpdateAccept] APDU, then
- 2.4.5.3.8.9.1 If in the UC-G-CANCEL state, the ground FIS UC module shall:
- a) stop timer t-UC-3,
 - b) request the FIS HI module to invoke a FIS-cancel-update-contract confirmation with the *FISContractNumber* parameter containing the information which has been received as the *ContractNumber* APDU-element, and
 - c) enter the UC-G-IDLE state.
- 2.4.5.3.8.10 Upon receipt of a request from the AB or CL modules to stop operation, then
- 2.4.5.3.8.10.1 The ground FIS UC module shall enter the UC-G-IDLE state.
- 2.4.5.3.8.11 Upon expiration of the t-UC-3 timer, then:
- 2.4.5.3.8.11.1 If in the UC-G-CANCEL state, the ground FIS UC module shall:
- a) request the AB module to abort with reason “timer expiration”, and
 - b) enter the UC-G-IDLE state.
- 2.4.5.3.9 Ground and Air AB Module

Note.— All statements in 2.4.5.3.9 apply to both the FIS ground AB module and the FIS air AB module.

- 2.4.5.3.9.1 Upon receipt of a request to abort from the HI, LI, DC, UC or CL modules, the AB module shall:
- a) request DC and UC modules to stop operation,
 - b) request the FIS HI module to invoke a FIS-provider-abort indication primitive with the *Reason* parameter set to value supplied by the module requesting the abort,
 - c) if the AB module is an air FIS module, create a FISDownlinkAPDU [FISAbort] APDU, with the *FISProtocolErrorDiag* APDU-element set to the value supplied by the module requesting the abort,
 - d) if the AB module is a ground FIS module, create a FISUplinkAPDU [FISAbort] APDU, with the *FISProtocolErrorDiag* APDU-element set to the value supplied by the module requesting the abort, and
 - e) request the LI module to send a D-ABORT request with the *originator* parameter set to the abstract value “provider”.

2.4.5.3.9.2 Upon receipt of a HI request to send a FIS-user-abort request, the AB module shall:

- a) request DC and UC modules to stop operation, and
- b) request the LI module to send a D-ABORT request with the *originator* parameter set to the abstract value “user”.

2.4.5.3.9.3 Upon receipt of a LI request to deliver a D-ABORT indication, the AB module shall:

- a) request DC and UC modules to stop operation,
- b) if the *originator* parameter contains the abstract value “user”, request the FIS HI module to invoke a FIS-user-abort indication primitive, and
- c) if the *originator* parameter contains the abstract value “provider”, request the FIS HI module to invoke a FIS-provider-abort indication primitive with the *FISProtocolErrorDiag* APDU-element of the received APDU as *Reason* parameter.

2.4.5.3.9.4 Upon receipt of a LI request to deliver a D-P-ABORT indication, the AB module shall:

- a) request DC and UC modules to stop operation, and
- b) request the FIS HI module to invoke a FIS-provider-abort indication with the abstract value “communication system failure” as *Reason* parameter.

2.4.5.3.10 Air Cancel (CL) Module

2.4.5.3.10.1 Upon receipt of a HI request to send a FIS-cancel-contracts request, if there are DC and UC modules handling contracts of the types specified in the *FISServiceType* parameter, the CL module shall:

- a) request these DC and UC modules to stop operation for the types of contracts concerned by the cancellation as specified in the *FISServiceType* parameter,
- b) create a FISDownlinkAPDU [FISCancelContracts] APDU based on the value of the *FISServiceType* parameter,
- c) request the LI module to send the APDU to the peer FIS-ASE, and
- d) start the t-CL-1 timer.

2.4.5.3.10.2 Upon receipt of [FISCancelContractsAccept] APDU and the *FISServiceType* APDU-element matches exactly the *FISServiceType* APDU-element of the [FISCancelContracts] APDU sent previously, the CL module shall:

- a) stop the t-CL-1 timer,
- b) request the FIS HI module to invoke a FIS-cancel-contracts confirmation primitive with the types of contracts concerned by the cancellation specified in the [FISCancelContractsAccept] APDU as *FISServiceType* parameter, and

- c) start the t-inactivity timer if there is no other FIS contract in place.

2.4.5.3.10.3 Upon expiration of the timer t-CL-1, the CL module shall request the AB module to abort with reason “timer expiration”.

2.4.5.3.11 Ground Cancel (CL) Module

2.4.5.3.11.1 Upon receipt of [FISCancelContracts] APDU, the CL module shall:

- a) request DC and UC modules to stop operation for the types of contracts concerned by the cancellation as specified in the [FISCancelContracts] APDU,
- b) request the FIS HI module to invoke a FIS-cancel-contracts indication primitive with the type(s) of contracts concerned by the cancellation specified in the [FISCancelContracts] APDU as *FISServiceType* parameter,
- c) create a FISUplinkAPDU [FISCancelContractsAccept] APDU based on the value of the *FISServiceType* parameter, and
- d) request the LI module to send the APDU to the peer FIS-ASE.

2.4.5.3.12 Ground and Air Low Interface Module

Note 1.— Except when explicitly indicated, the statements in 2.4.5.3.12 apply to both the FIS ground LI module and the FIS air LI module.

Note 2.— Table 2.4.5-4 specifies the type of the APDUs the LI module can send to the peer FIS-ASE.

Table 2.4.5-4. Types of the FIS-ASE APDUs

APDU	Type
FISRequest	initial
FISAccept	normal
FISReject	normal
FISReport	normal
FISCancelUpdateContract	normal
FISCancelUpdateAccept	normal
FISCancelContracts	normal
FISCancelContractsAccept	normal

Note 3.— The states defined for the LI module are the following:

- a) *LI-IDLE*,
- b) *LI-START-I (initiator of the D-START req)*,

- c) *LI-START-R* (receptor of the *D-START* ind),
- d) *LI-DIALOGUE*, and
- e) *LI-END-I* (initiator of the *D-END* req).

2.4.5.3.12.1 On initiation, the LI module shall be in the LI-IDLE state.

2.4.5.3.12.2 On receipt of a request to send an APDU, the FIS LI module shall determine the APDU type based on the Table 2.4.5-4.

2.4.5.3.12.3 When requested by a DC or UC module to send an initial APDU to the peer ASE, then

2.4.5.3.12.3.1 If in the LI-IDLE state, the LI module shall:

- a) invoke the *D-START* request primitive with the following parameters:
 - 1) the ICAO Facility Designation provided by the DC or UC module as the *CalledPeerId* parameter,
 - 2) the *QualityOfService* parameter composed as follows:
 - i) the Routing Class based on the value of the *ClassOfCommunicationService* parameter if provided by the DC or UC module,
 - ii) the Application Service Priority as defined in 2.4.6.2.2.1, and
 - iii) the RER as defined in 2.4.6.2.2.2, and
 - 3) the APDU as *UserData* parameter, and
- b) enter the LI-START-I state.

2.4.5.3.12.3.2 If in the LI-DIALOGUE state, the LI module shall:

- a) invoke the *D-DATA* request primitive with the APDU as *UserData* parameter, and
- b) remain in the LI-DIALOGUE state.

2.4.5.3.12.4 When requested by a DC or UC module to send a normal APDU to the peer ASE, then

2.4.5.3.12.4.1 If in the LI-DIALOGUE state, the LI module shall:

- a) invoke the *D-DATA* request primitive with the APDU as *UserData* parameter, and
- b) remain in the LI-DIALOGUE state.

2.4.5.3.12.4.2 If in the LI-START-R state, the LI module shall:

- a) invoke the D-START response primitive with the following parameters:
 - 1) the abstract value “accepted” as the *Result* parameter, and
 - 2) the APDU as the *UserData* parameter, and
- b) enter the LI-DIALOGUE state.

2.4.5.3.12.5 When requested by the AB module to send a D-ABORT request, then

2.4.5.3.12.5.1 The LI module shall:

- a) if a dialogue exists, invoke the D-ABORT request primitive with the APDU as *UserData* parameter and the value supplied by the AB module as *Originator* parameter, and
- b) enter the LI-IDLE state.

2.4.5.3.12.6 Upon receipt of a D-START indication with the *UserData* parameter containing a FISDownlinkAPDU [FISRequest] APDU and the D-START QOS *Priority* parameter has the abstract value “Aeronautical Information Service messages” and the D-START QOS *Residual Error Rate* parameter has the abstract value “low” and the D-START QOS *Routing Class* parameter identifies the traffic category “Air Traffic Service Communications (ATSC)” and the D-START *Calling Peer Id* parameter is a valid 24 bit address, then :

2.4.5.3.12.6.1 If in the LI-IDLE state, the LI ground module shall:

- a) pass the APDU to the DC module if the APDU contains a *demandContract* APDU-element,
- b) pass the APDU to the UC module if the APDU contains an *updateContract* APDU-element, and
- c) enter the LI-START-R state.

2.4.5.3.12.7 Upon receipt of a D-START confirmation with a *Result* parameter containing the abstract value “accepted” and with a *UserData* parameter containing a FISUplinkAPDU [FISAccept] or [FISReject] APDU, then

2.4.5.3.12.7.1 If in the LI-START-I state, the LI air module shall:

- a) identify the module from the *ContractNumber* parameter of the received APDU,
- b) pass the APDU to the relevant DC or UC module, and
- c) enter the LI-DIALOGUE state.

2.4.5.3.12.8 Upon receipt of a D-START confirmation with the *RejectSource* parameter containing the abstract value “DS provider”, then:

2.4.5.3.12.8.1 If in the LI-START-I state, the LI module shall:

- a) request the AB module to abort with reason “cannot establish contact with the peer”, and
- b) enter the LI-IDLE state.

2.4.5.3.12.9 Upon receipt of a D-START confirmation with the *RejectSource* parameter containing the abstract value “DS user”, then:

2.4.5.3.12.9.1 If in the LI-START-I state, the LI module shall:

- a) request the AB module to abort with the reason “contact refused by the peer”, and
- b) enter the LI-IDLE state.

2.4.5.3.12.10 Upon receipt of a D-DATA indication with a *UserData* parameter containing a valid APDU (i.e. any APDU except [FISAbort] APDU), then

2.4.5.3.12.10.1 If in the LI-DIALOGUE or LI-END-I state, the LI module shall:

- a) identify the module:
 - 1) the DC module if the [FISRequest] APDU contains an *demandContract* APDU-element,
 - 2) the UC module if the [FISRequest] APDU contains an *updateContract* APDU-element,
 - 3) the DC or UC module based on the *ContractNumber* APDU-element of the received APDU, if the APDU is one of the following: [FISAccept], [FISReject], [FISCancelUpdateContract], [FISCancelUpdateAccept] or [FISReport], or
 - 4) the CL module if the APDU is [FISCancelContracts] or [FISCancelContractsAccept],
- b) if the APDU is not a [FISCancelUpdateContract] APDU, pass the APDU to that module,
- c) if the APDU is a [FISCancelUpdateContract] APDU, then:
 - 1) if the FIS-air-user has not initiated a global cancellation (FIS-cancel-contracts) for the type of FIS contracts identified in the APDU-element FISCancelUpdateData, pass the APDU to that module, or
 - 2) otherwise, discard the APDU, and

- d) remain in the same state.

2.4.5.3.12.11 Upon receipt of a D-END indication, then

2.4.5.3.12.11.1 If in the LI-DIALOGUE state and if there is no FIS contract in place, the LI ground module shall:

- a) invoke the D-END response primitive with the abstract value “accepted” as *Result* parameter, and
- b) enter the LI-IDLE state.

2.4.5.3.12.12 Upon receipt of a D-END confirmation with a *Result* parameter containing the abstract value “accepted”, then

2.4.5.3.12.12.1 If in the LI-END-I state, the LI air module shall:

- a) stop the t-LI-1 timer, and
- b) enter the LI-IDLE state.

2.4.5.3.12.13 Upon receipt of a D-END confirmation with a *Result* parameter containing the abstract value “rejected”, then:

2.4.5.3.12.13.1 If in the LI-END-I state, the LI air module shall:

- a) stop the t-LI-1 timer,
- b) request the AB module to abort with reason “dialogue-end-not-supported”, and
- c) remain in the LI-END-I state.

2.4.5.3.12.14 Upon receipt of a D-ABORT indication with the *UserData* parameter, the LI module shall:

- a) stop the t-LI-1 timer, if set,
- b) forward the primitive to the AB Module, and
- c) enter the LI-IDLE state.

2.4.5.3.12.15 Upon receipt of a D-P-ABORT indication, the LI module shall:

- a) stop the t-LI-1 timer, if set,
- b) forward the primitive to the AB module, and
- c) enter the LI-IDLE state.

2.4.5.3.12.16 Upon receipt of an indication that the t-inactivity timer has expired, then

2.4.5.3.12.16.1 If in LI-DIALOGUE state, the LI air module shall:

- a) start the timer t-LI-1,
- b) invoke the D-END request, and
- c) enter the LI-END-I state.

2.4.5.3.12.17 Upon receipt of an indication that the t-LI-1 timer has expired, then

2.4.5.3.12.17.1 If in LI-END-I state, the LI air module shall:

- a) request the AB module to abort with the reason “timer expiration”, and
- b) remain in the LI-END-I state.

2.4.5.4 Exception Handling

2.4.5.4.1 Unrecoverable Internal Error

2.4.5.4.1.1 **Recommendation.**— *When any module has an unrecoverable system error, the module should:*

- a) *request the AB module to abort with reason “unrecoverable internal error”, and*
- b) *remains in its current state.*

2.4.5.4.2 Protocol Error

2.4.5.4.2.1 When the LI module detects that the *UserData* parameter of a dialogue service indication or confirmation does not contain an APDU and the service is not D-END, the LI module shall:

- a) request the AB module to abort with the reason “protocol error”, and
- b) remain in its current state.

2.4.5.4.2.2 Upon receipt of a Dialogue service primitive for which there are no instruction in 2.4.5.3 (i.e. the primitive was not expected, or was expected under other conditions or with other parameter values), the LI module shall:

- a) request the AB module to abort with the reason “protocol error”, and
- b) remains in its current state.

2.4.5.4.3 Sequence Error

2.4.5.4.3.1 When an APDU is passed to any module except the LI and the HI modules for which there are no instructions in 2.4.5.3 (i.e. the PDU has arrived out of sequence), that module shall:

- a) request the AB module to abort with the reason “sequence error”, and
- b) remains in its current state.

2.4.5.4.4 Decoding Error

2.4.5.4.4.1 When the LI module fails to decode an APDU, the LI module shall:

- a) request the AB module to abort with the reason “decoding error”, and
- b) remain in its current state.

2.4.5.4.5 Invalid FIS Contract Number

2.4.5.4.5.1 When the LI module detects that the *ContractNumber* APDU-element in the decoded APDU is invalid (i.e. the *ContractNumber* APDU-element in a non initial APDU is not already in use or the *ContractNumber* APDU-element in an initial APDU is already in use), the LI module shall:

- a) request the AB module to abort with the reason “invalid contract number”, and
- b) remain in its current state.

2.4.5.4.6 D-START Indication Quality of Service Parameter Not As Expected

2.4.5.4.6.1 When the LI module detects that the *QualityOfService* parameter of a D-START indication primitive does not contain the abstract value “Aeronautical Information Service messages” as application service priority or does not contain the abstract value “low” as RER or does not identify the traffic category “Air Traffic Service Service Communications (ATSC)”, the LI module shall:

- a) request the AB module to abort with the reason “invalid QOS parameter”, and
- b) remain in its current state.

2.4.5.5 State Tables

2.4.5.5.1 Priority

2.4.5.5.1.1 If the state tables for the modules of the FIS-air-ASE and the FIS-ground-ASE shown below conflict with textual statements made elsewhere in this document, the textual statements shall take precedence.

Note 1.— The following notation conventions apply for the following state tables :

- a) *Module States are identified as follows : <module name>-<location(A/G)>-<state>.*
- b) *Events between a source module and a recipient module are identified as shown in the following figure:*

Note 2.— In the following state tables, the statement “cannot occur” means that if the implementation conforms to the SARPs, it is impossible for this event to occur. If the event does occur, this implies that there is an error in the implementation. If such a situation is detected, it is suggested that the ASE aborts with the error “unrecoverable internal error”.

Note 3.— In the following state tables, the statement “not permitted” means that the implementation must prevent this event from occurring through some local means. If the event does occur this implies that there is an error in the implementation. If such a situation is detected, it is suggested that the ASE performs a local rejection of the request rather than aborting the dialogue.

Note 4.— As the HI Module provides the ASE-users with a straightforward pass-through service to the DC, UC, CL and AB Modules, state tables related to the HI Modules are not described. FIS-xxx request and response primitives are mapped onto DC-xxx, UC-xxx, CL-xxx or AB-xxx request and response primitives; DC-xxx, UC-xxx, CL-xxx or AB-xxx indication and confirmation primitives are mapped onto FIS-xxx indication and confirmation primitives, with the following exception: when AB-user-abort indication or AB-provider-abort indication are received by the HI Module from the AB Module, the corresponding FIS-user-abort indication or FIS-provider-abort indication is only generated to the FIS-user when the FIS-user is active.

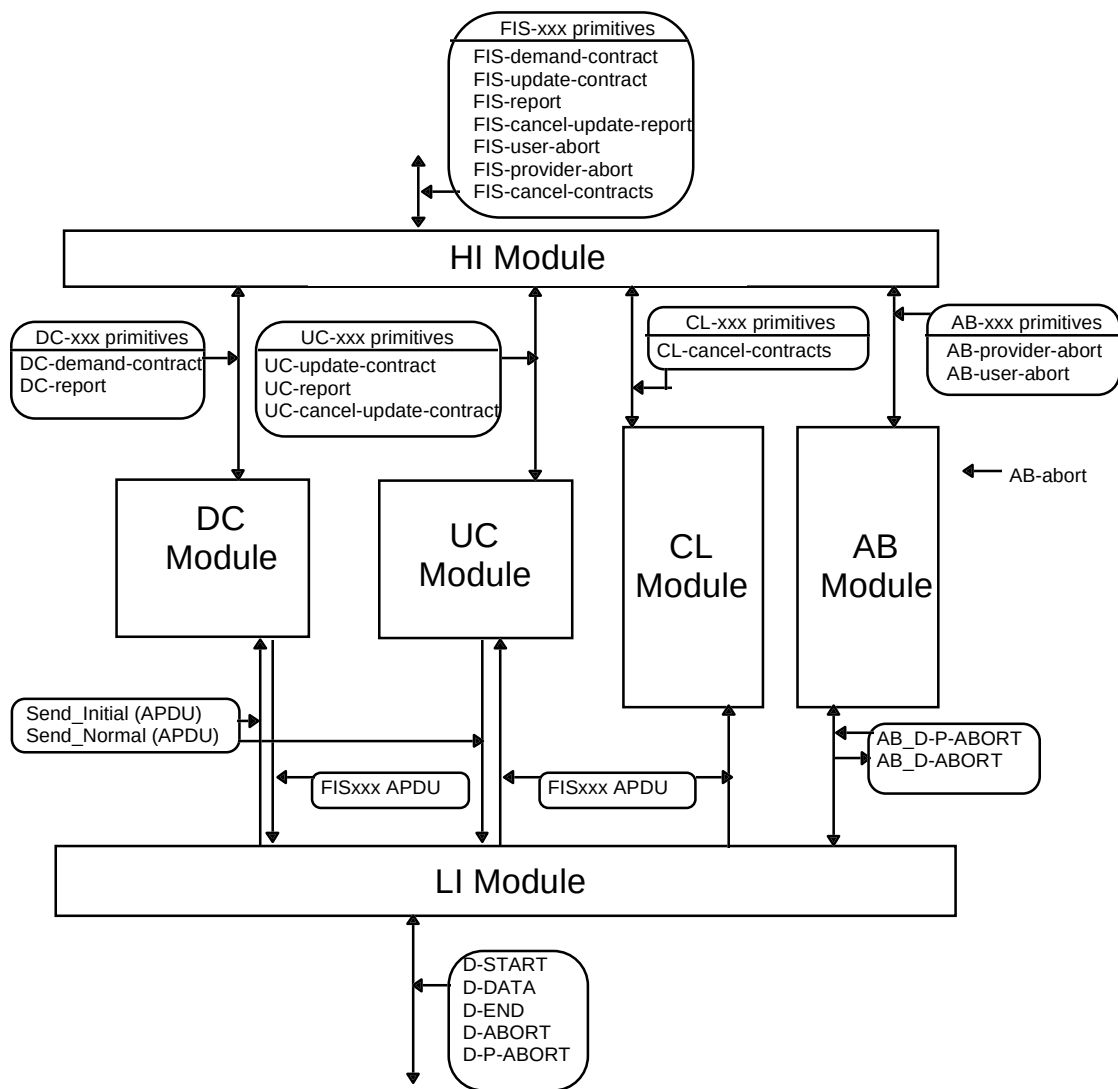
**Figure 2.4.5-20. Functional Model for FIS ASE State Tables**

Table 2.4.5-5/a. Air FIS DC Module State Table

State ⇒	DC-A-IDLE (Initial State)	DC-A-PENDING	DC-A-ACTIVE
Event ↓			
Primitive Requests and Responses from FIS HI module			
DC-demand-contract req	LI-Send-Initial [FISRequest-DemandContract] APDU Start t-DC-1, t-DC-2 →DC-A-PENDING	not permitted	not permitted
FIS APDUs from LI module			
[FISReport] APDU	cannot occur	AB-abort (“sequence error”) →DC-A-IDLE	stop t-DC-2 if last FIScontract start t-INACTIVITY DC-report ind →DC-A-IDLE
[FISAccept-accept] APDU	cannot occur	stop t-DC-1, t-DC-2 if last FIScontract start t-INACTIVITY DC-demand-contract cnf →DC-A-IDLE	AB-abort (“sequence error”) →DC-A-IDLE
[FISAccept-positive acknowledgement] APDU	cannot occur	stop t-DC-1 DC-demand-contract cnf →DC-A-ACTIVE	AB-abort (“sequence error”) →DC-A-IDLE
[FISReject] APDU	cannot occur	stop t-DC-1, t-DC-2 if last FIScontract start t-INACTIVITY DC-demand-contract cnf →DC-A-IDLE	AB-abort (“sequence error”) →DC-A-IDLE
Requests from AB module			
DC-stop-operation	→DC-A-IDLE	stop t-DC-1, t-DC-2 →DC-A-IDLE	stop t-DC-2 →DC-A-IDLE

Table 2.4.5-5/b. Air FIS DC Module State Table

State ⇒	DC-A-IDLE (Initial State)	DC-A-PENDING	DC-A-ACTIVE
Event ↓			
Timer expiration			
t-DC-2	cannot occur	cannot occur	AB-abort (“timer expiration”) →DC-A-IDLE
t-DC-1	cannot occur	Stop t-DC-2 AB-abort (“timer expiration”) →DC-A-IDLE	cannot occur

Table 2.4.5-6. Ground FIS DC Module State Table

State ⇒	DC-G-IDLE (Initial State)	DC-G-PENDING	DC-G-ACTIVE
Event ↓			
Primitive Requests and Responses from HI Module			
DC-demand-contract rsp (accepted)	not permitted	LI-Send-Normal [FISAccept-accept] APDU →DC-G-IDLE	not permitted
DC-demand-contract rsp (rejected)	not permitted	LI-Send-Normal [[FISReject] APDU →DC-G-IDLE	not permitted
DC-demand-contract rsp (positive acknowledgement)	not permitted	LI-Send-Normal [FISAccept-positive-ack] APDU →DC-G-ACTIVE	not permitted
DC-report req	not permitted	not permitted	LI-Send-Normal [FISReport] APDU →DC-G-IDLE
APDU from LI Module			
[FISRequest-DemandContract] APDU	DC-demand-contract ind →DC-G-PENDING	AB-abort (“sequence error”) →DC-G-IDLE	AB-abort (“sequence error”) →DC-G-IDLE
APDU from AB Module			
DC-stop-operation	→DC-G-IDLE	→DC-G-IDLE	→DC-G-IDLE

Table 2.4.5-7/a. Air FIS UC Module State Table

State →	UC-A-IDLE (Initial State)	UC-A-PENDING	UC-A-ACTIVE	UC-A-CANCEL
Event ↓				
Primitive Requests and Responses from HI				
UC-update-contract req	LI-Send-Initial [FIS-Request-Update Contract] APDU Start t-UC-1, t-UC-2 →UC-A-PENDING	not permitted	not permitted	not permitted
UC-cancel-update-contract req	not permitted	if dialogue not established not permitted else stop t-UC-1, t-UC-2 LI-Send-Normal [FISCANCELUpdate Contract] APDU start T-UC-3 CANCELFROM PENDING = TRUE →UC-A-CANCEL	if set, stop t-UC-2 LI-Send-Normal [FISCANCELUpdate Contract] APDU start t-UC-3 →UC-A-CANCEL	not permitted
FIS uplink PDUs from LI				
[FISAccept-accept] APDU	cannot occur	stop t-UC-1, t-UC-2 UC-update-contract cnf →UC-A-ACTIVE	AB-abort ("sequence error") →UC-A-IDLE	If CANCELFROM PENDING = TRUE stop t-UC-1, t-UC-2 CANCELFROM PENDING = FALSE →UC-A-CANCEL
[FISAccept-positive acknowledgement] APDU	cannot occur	stop t-UC-1 UC-update-contract cnf →UC-A-ACTIVE	AB-abort ("sequence error") →UC-A-IDLE	If CANCELFROM PENDING = TRUE stop t-UC-1, t-UC-2 CANCELFROM PENDING = FALSE →UC-A-CANCEL

Table 2.4.5-7/b. Air FIS UC Module State Table

State →	UC-A-IDLE (Initial State)	UC-A-PENDING	UC-A-ACTIVE	UC-A-CANCEL
Event ↓				
[FISReject] APDU	cannot occur	stop t-UC-1, t-UC-2 if last FIScontract start t-INACTIVITY UC-update-contract cnf →UC-A-IDLE	AB-abort (“sequence error”) →UC-A-IDLE	If CANCELFROM PENDING = TRUE stop t-UC-1, t-UC-2 CANCELFROM PENDING = FALSE UC-cancel-update- contract cnf →UC-A-IDLE else AB-abort (“sequence error”) →UC-A-IDLE
[FISReport] APDU	cannot occur	AB-abort (“sequence error”) →UC-A-IDLE	If set, stop t-UC-2 UC-report ind →UC-A-ACTIVE	If CANCELFROM PENDING = FALSE →UC-A-CANCEL else AB-abort (“sequence error”) →UC-A-IDLE
[FISCancelUpdateContract] APDU	cannot occur	UC-cancel-update- contract ind LI-Send-Normal [FISCancelUpdate Accept] APDU if last FIS Contract start t-inactivity →UC-A-IDLE	If set, stop t-UC-2 UC-cancel-update- contract ind LI-Send-Normal [FISCancelUpdate Accept] APDU if last FIS Contract start t-inactivity →UC-A-IDLE	/* collision */ LI-Send-Normal [FISCancelUpdateAcc ept] APDU →UC-A-CANCEL
[FISCancelUpdateAccept] APDU	cannot occur	AB-abort (“sequence error”) →UC-A-IDLE	AB-abort (“sequence error”) →UC-A-IDLE	stop t-UC-3 UC-cancel-update- contract cnf if last FIS Contract start t-inactivity →UC-A-IDLE
Request from AB module				
UC-stop-operation	→UC-A-IDLE	stop t-UC-1, t-UC-2 →UC-A-IDLE	stop t-UC-2 →UC-A-IDLE	→UC-A-IDLE
Timer Expiration				
t-UC-1	cannot occur	Stop t-UC-2 AB-abort (“timer expiration”) →UC-A-IDLE	cannot occur	cannot occur
t-UC-2	cannot occur	cannot occur	AB-abort (“timer expiration”) →UC-A-IDLE	cannot occur

State →	UC-A-IDLE (Initial State)	UC-A-PENDING	UC-A-ACTIVE	UC-A-CANCEL
Event ↓				
t-UC-3	cannot occur	cannot occur	cannot occur	AB-abort (“timer expiration”) →UC-A-IDLE

Table 2.4.5-8/a. Ground FIS UC Module State Table

State ⇒	UC-G-IDLE (Initial State)	UC-G-PENDING	UC-G-ACTIVE	UC-G-CANCEL
Event ↓				
Primitive Requests and Responses from HI				
UC-update-contract rsp (accepted)	not permitted	LI-Send-Normal [FISAccept-accept] APDU →UC-G-ACTIVE	not permitted	not permitted
UC-update-contract rsp (rejected)	not permitted	LI-Send-Normal [FISReject] APDU →UC-G-IDLE	not permitted	not permitted
UC-update-contract rsp (processing)	not permitted	LI-Send-Normal [FISAccept- processing] APDU →UC-G-ACTIVE	not permitted	not permitted
UC-report req	not permitted	not permitted	LI-Send-Normal [FISReport] APDU →UC-G-ACTIVE	not permitted
UC-cancel-update- contract req	not permitted	LI-Send-Normal [FISCancelUpdate Contract] APDU start t-UC-3 →UC-G-CANCEL	LI-Send-Normal [FISCancelUpdate Contract] APDU start t-UC-3 →UC-G-CANCEL	not permitted

Table 2.4.5-8/b. Ground FIS UC Module State Table

State →	UC-G-IDLE (Initial State)	UC-G-PENDING	UC-G-ACTIVE	UC-G-CANCEL
Event ↓				
FIS downlink PDUs from LI				
[FISRequest-UpdateContract] APDU	UC-update-contract ind → UC-G-PENDING	AB-abort (“protocol error”) → UC-G-IDLE	AB-abort (“protocol error”) → UC-G-IDLE	AB-abort (“protocol error”) → UC-G-IDLE
[FISCANCELUpdateContract] APDU	cannot occur	UC-cancel-update-contract ind LI-Send-Normal [FISCANCELUpdateAccept] APDU → UC-G-IDLE	UC-cancel-update-contract ind LI-Send-Normal [FISCANCELUpdateAccept] APDU → UC-G-IDLE	/* collision */ LI-Send-Normal [FISCANCELUpdateAccept] APDU → UC-G-CANCEL
[FISCANCELUpdateAccept] APDU	cannot occur	AB-abort (“protocol error”) → UC-G-IDLE	AB-abort (“protocol error”) → UC-G-IDLE	stop t-UC-3 FIS-cancel-update-contract cnf → UC-G-IDLE
Request from AB module				
UC-stop-operation	→ UC-G-IDLE	→ UC-G-IDLE	→ UC-G-IDLE	→ UC-G-IDLE
Timer Expiration				
t-UC-3	cannot occur	cannot occur	cannot occur	AB-abort (“timer expiration”) → UC-G-IDLE

Table 2.4.5-9/a. Air and ground FIS LI Modules State Table

State →	LI-IDLE (Initial State)	LI-START-I	LI-DIALOGUE	LI-START-R	LI-END-I (air LI)
Event ↓					
Primitive Requests and Responses from DC, UC or AB Module					
LI-Send-Initial (APDU)	D-START req → LI-START- I	not permitted	D-DATA req	not permitted /* only air- initiated */	not permitted
LI-Send-Normal (APDU)	not permitted	not permitted	D-DATA req	D-START rsp → LI- DIALOGUE	not permitted
Primitive Requests from AB Module					
AB_D-ABORT req	if dialog exists D-ABORT req → LI-IDLE	if dialog exists D-ABORT req → LI-IDLE	if dialog exists D-ABORT req → LI-IDLE	if dialog exists D-ABORT req → LI-IDLE	if dialog exists D-ABORT req → LI-IDLE
Timer Expiration					
t-LI-1 (air LI)	cannot occur	cannot occur	cannot occur	cannot occur	AB-abort ("timer expiration") → LI-END-I
t-INACTIVITY (air LI)	cannot occur	cannot occur	start t-LI-1 D-END req → LI-END-I	cannot occur	cannot occur

Table 2.4.5-9/b. Air and ground FIS LI Modules State Table

State →	LI-IDLE (Initial State)	LI-START-I	LI-DIALOGUE	LI-START-R	LI-END-I (air LI)
Event ↓					
Dialogue Service primitives from DSP					
D-START ind	Pass APDU → LI-START- R	cannot occur	cannot occur	cannot occur	cannot occur
D-START cnf positive (result=accepted)	cannot occur	Pass APDU → LI- DIALOGUE	cannot occur	cannot occur	cannot occur
D-START cnf negative (result=rejectedtran sient & source=DS provider)	cannot occur	AB-abort ("cannot establish contact with the peer") → LI-IDLE	cannot occur	cannot occur	cannot occur
D-START cnf negative (result=rejectedtran sient & source=DS user)	cannot occur	AB-abort ("contact refused by the peer") → LI-IDLE	cannot occur	cannot occur	cannot occur
D-DATA ind	cannot occur	cannot occur	Pass APDU	cannot occur	Pass APDU
D-END ind (ground LI)	cannot occur	cannot occur	if no FIS contract D-END rsp positive → LI-DLE else AB-abort ("protocol error") → LI-DIALOGUE	cannot occur	cannot occur
D-END cnf positive (air LI)	cannot occur	cannot occur	cannot occur	cannot occur	stop t-LI-1 → LI-IDLE
D-END cnf negative (air LI)	cannot occur	cannot occur	cannot occur	cannot occur	stop t-LI-1 AB-abort ("dialogue end not supported") → LI-END-I
D-ABORT ind	cannot occur	AB_D-ABORT ind → LI-IDLE	AB_D-ABORT ind → LI-IDLE	AB_D-ABORT ind → LI-IDLE	stop t-LI-1 AB_D-ABORT ind → LI-IDLE
D-P-ABORT ind	cannot occur	AB_D-P-ABORT ind → LI-DLE	AB_D-P-ABORT ind → LI-DLE	AB_D-P- ABORT ind → LI-DLE	stop t-LI-1 AB_D-P- ABORT ind → LI-DLE

Table 2.4.5-10. Air and Ground FIS AB Modules State Table

AB module state →	AB-IDLE
Event ↓	
Primitive Requests from DC and UC Module	
AB-abort (reason)	UC-stop-operation (all FIS services) DC-stop-operation AB-provider-abort ind (reason) AB-D-ABORT req (“provider ” [FISAbort] APDU)
Primitive Indications and APDUs from LI Module	
AB_D-P-ABORT ind (reason)	UC-stop-operation (all FIS services) DC-stop-operation (all FIS services) AB-provider-abort ind (reason)
AB_D-ABORT ind (« provider »,APDU)	UC-stop-operation (all FIS services) DC-stop-operation (all FIS services) AB-provider-abort ind (APDU.AbortReason)
AB_D-ABORT ind (« user »)	UC-stop-operation (all FIS services) DC-stop-operation (all FIS services) AB-user-abort ind ()
Primitive Requests from HI Module	
AB-user-abort req	UC-stop-operation (all FIS services) DC-stop-operation (all FIS services) AB-D-ABORT req (« user »)

Table 2.4.5-11. Air FIS CL Module State Table

CL module state ⇒	CL-IDLE
Event ↓	
Primitive Request from the FIS HI module	
CL-cancel-contracts req (types of FIS service)	if UC or DC exist with these types of FIS service UC-stop-operation (type of FIS service) DC-stop-operation (type of FIS service) LI-Send-Normal [FISCcancelContracts] APDU start t-CL-1
APDUs from LI Module	
[FISCcancelContractsAc cep t] APDU	stop t-CL-1 CL-cancel-contracts cnf if last FIS Contract, start t-inactivity
Timer Expiration	
t-CL-1	AB-abort (« timer expiration »)

Table 2.4.5-12. Ground FIS CL Module State Table

CL module state ⇒	CL-IDLE
Event ↓	
APDUs from LI Module	
[FISCancelContracts] APDU	UC-stop-operation (type of FIS service) DC-stop-operation (type of FIS service) CL-cancel-contracts ind LI-Send-Normal [FISCancelContractsAccept] APDU

2.4.6 COMMUNICATION REQUIREMENTS

2.4.6.1 Encoding Rules

2.4.6.1.1 The FIS application shall use PER as defined in ISO/IEC 8825-2, using the Basic Unaligned variant to encode/decode the ASN.1 message structure and content specified in 2.4.4.

2.4.6.2 Dialogue Service Requirements

2.4.6.2.1 Primitives Requirements

2.4.6.2.1.1 Where dialogue service primitives, that is D-START, D-END, D-ABORT, D-P-ABORT, and D-DATA are described as being invoked in 2.4.5, the FIS-ground-ASE and the FIS-air-ASE shall exhibit external behavior consistent with the dialogue service, as described in 4.2, having been implemented and its primitives invoked.

2.4.6.2.2 Quality Of Service Requirements

2.4.6.2.2.1 The application service priority for ATIS shall have the abstract value of “Aeronautical Information Service messages”.

2.4.6.2.2.2 The Residual Error Rate (RER) Quality Of Service parameter of the D-START request shall be set to the abstract value of “low”.

2.4.6.2.2.3 The FIS-ASE shall map the FIS-service Class of Communication Service abstract value to the ATSC routing class abstract value part of the D-START *Quality Of Service* parameter as presented in Table 4.2.6-1.

Table 4.2.6-1. Mapping between Class of Communication and Routing Class Abstract Values

Class of Communication Abstract Value	Routing Class Abstract Value
A	Traffic follows Class A ATSC route(s)
B	Traffic follows Class B ATSC route(s)
C	Traffic follows Class C ATSC route(s)
D	Traffic follows Class D ATSC route(s)
E	Traffic follows Class E ATSC route(s)
F	Traffic follows Class F ATSC route(s)
G	Traffic follows Class G ATSC route(s)
H	Traffic follows Class H ATSC route(s)

Note.— ATSC values are defined in 1.3.

2.4.7 FIS USER REQUIREMENTS

2.4.7.1 Introduction

Note.— Requirements imposed on FIS-users interfacing with the FIS-ASEs are presented in 2.4.7.

2.4.7.2 General

2.4.7.2.1 General Requirements

Note 1.— When a FIS-air-user invokes FIS-demand-contract request or FIS-update-contract request and requires a particular class of communication service, it sets the ClassOfCommunicationService parameter to be the class of communication service it requires.

Note 2.— When a FIS-air-user invokes FIS-demand-contract request or FIS-update-contract request and does not provide the ClassOfCommunicationService parameter, this indicates no routing preference.

Note 3.— When a FIS-air-user specifies the ClassOfCommunicationService parameter and there is a FIS contract in place, the ClassOfCommunicationService parameter is ignored.

2.4.7.2.2 Response Time Requirements

2.4.7.2.2.1 Recommendation.—*Upon receipt of a FIS-ASE service indication that requires a response, the FIS-user should invoke the FIS-ASE service response within 1 second.*

2.4.7.2.3 Error Handling Requirements

2.4.7.2.3.1 If a FIS-user has an unrecoverable system error it shall:

- a) cease the operation of all contracts with peer systems which are affected by the error, and
- b) for each affected peer system, invoke FIS-user-abort request.

2.4.7.2.3.2 If a FIS-user receives a FIS-provider-abort indication or a FIS-user-abort indication, it shall cease operation of all FIS contracts with the peer system to which the indication is related.

2.4.7.2.4 Miscellaneous Air and Ground FIS Systems Requirements

2.4.7.2.4.1 With the exception of the FIS-user-abort and FIS-provider-abort, the FIS-user shall respond to FIS-ASE service indications and confirmations from the FIS-ASE in the order in which they are received.

2.4.7.2.5 Miscellaneous Ground FIS System Requirements

2.4.7.2.5.1 The FIS-ground-user shall be capable of detecting that the content of the FIS request is not valid or that the requested FIS information is not available.

2.4.7.2.5.2 The FIS-ground-user shall be capable of detecting that it can not continue to provide updates of the requested FIS information.

2.4.7.2.5.3 The FIS-ground-user shall be able to support multiple concurrent FIS contracts with a given aircraft.

2.4.7.2.6 ATIS Service Requirements

2.4.7.2.6.1 If only the arrival ATIS or only the departure ATIS is requested, and if the FIS-ground-user only provides combined ATIS, the FIS-ground-user shall send combined ATIS with a 'combined' indication.

2.4.7.2.6.2 If both arrival and departure ATIS are requested, the FIS-ground-user shall:

- a) send a combined ATIS with a 'combined' indication, if the combined ATIS is available, or
- b) send an arrival ATIS and a departure ATIS if both information are available but the combined ATIS is not available, or
- c) send the available ATIS with an indication that the other is not available, if only one is available.

2.4.7.2.6.3 **Recommendation.**— *The ATIS fields should be presented in the following order:*

- a) *Airport identification,*
- b) *Departure or arrival indicator,*
- c) *ATIS Code,*
- d) *Time of Observation, if present,*
- e) *Approach Type, if present,*
- f) *Runways In Use,*
- g) *RunwayArresting System, if present,*
- h) *RunwaySurfaceConditions, if present,*
- i) *RunwayBraking Action, if present,*
- j) *Holding Delay, if present,*
- k) *Transition Level, if present,*
- l) *Other Operational Information, if present,*
- m) *Surface Wind,*
- n) *Visibility,*

- o) *RVR, if present,*
- p) *Present Weather,*
- q) *Cloud Sky and Cover,*
- r) *Air Temperature,*
- s) *Dew Point Temperature,*
- t) *Altimeter Setting,*
- u) *Significant Met Information,*
- v) *Trend Type of Landing Forecast, if present, and*
- w) *Specific ATIS Instruction, if present.*

2.4.7.3 Establishment And Operation Of a FIS Demand Contract

Note 1.— When a FIS-air-user requires to establish a FIS demand contract with a FIS-ground-user it invokes FIS-demand-contract request.

Note 2.— Only the FIS-air-user is capable of initiating the FIS-demand-contract service.

Note 3.— If the FIS-air-user uses a value for the FISContractNumber parameter already in use, the FIS Demand Contract request will be rejected.

Note 4.— If the FIS-air-user invokes a second FIS-demand-contract request before the very first request (either a FIS-demand-contract or a FIS-update-contract request) has been confirmed, the FIS-demand-contract will be rejected.

2.4.7.3.1 The FIS-ground-user shall use the value received in the *FISContractNumber* parameter of the FIS-demand-contract indication for the *FISContractNumber* parameter of any ground-initiated request or response related to that FIS demand contract.

2.4.7.3.2 The same type of FIS information (i.e. “ATIS” for version 1 of the FIS-ASEs) shall be identified by the FIS-air-user and FIS-ground-user when invoking the FIS-demand-contract request and response primitives.

2.4.7.3.3 Upon receipt of a FIS-demand-contract indication, when the FIS-ground-user is able to accept the contract in full, then:

2.4.7.3.3.1 If the FIS-ground-user is not able to send the FIS report within the response time specified in 2.4.7.2.2., it shall:

- a) invoke the FIS-demand-contract response with the *Result* parameter set to abstract value “positive acknowledgment”, and

- b) invoke the FIS-report request later.

2.4.7.3.3.2 **Recommendation.**— *If the FIS-ground-user is not able to send the FIS report within the response time, it should invoke the FIS-demand-contract response within the response time and then invoke the FIS-report request within 10 seconds.*

2.4.7.3.3.3 If the FIS-ground-user is able to send the FIS report within the response time specified in 2.4.7.2.2., it shall invoke the FIS-demand-contract response with the following parameters:

- a) the Result parameter set to the abstract value “accepted”, and
- b) the FIS report in the *FIS Information* parameter.

2.4.7.3.4 Upon receipt of a FIS-demand-contract indication, if the FIS-ground-user cannot comply with the demand contract request, it shall reject the contract by invoking a FIS-demand-contract response with the following parameters:

- a) the abstract value “rejected” as *Result* parameter, and
- b) one of the following abstract values “cannot comply”, “FIS service not available”, “error detected in the FIS request” or “undefined”, as *RejectReason* parameter.

2.4.7.3.5 The FIS-air-user shall be allowed to reuse the value of a *FISContractNumber* used in a FIS-demand-contract request for a new FIS contract:

- a) upon receipt of a FIS-demand-contract confirmation with a *Result* parameter containing the abstract value “rejected” or “accepted”,
- b) upon receipt of a FIS-report indication, a FIS-user-abort indication or a FIS-provider-abort indication, or
- c) after invocation of a FIS-user-abort request.

2.4.7.4 Establishment and Operation of a FIS Update Contract

Note 1.— When an FIS-air-user requires to establish a FIS update contract with an FIS-ground-user it invokes the FIS-update-contract request.

Note 2.— Only the FIS-air-user is capable of initiating the FIS-update-contract service.

Note 3.— If the FIS-air-user uses a value for the FISContractNumber parameter already in use, the FIS Update Contract request will be rejected.

Note 4.— If the FIS-air-user invokes a second FIS-update-contract request before the very first request (either a FIS-demand-contract or a FIS-update-contract request) has been confirmed, the FIS-update-contract will be rejected.

2.4.7.4.1 The FIS-ground-user shall use the value received in the *FISContractNumber* parameter of the FIS-update-contract indication for the *FISContractNumber* parameter of any ground-initiated request or response related to that FIS update contract.

2.4.7.4.2 The same type of FIS information (i.e. “ATIS” for version 1 of the FIS-ASEs) shall be identified by the FIS-air-user and FIS-ground-user when invoking the FIS-update-contract request and response primitives.

2.4.7.4.3 Upon receipt of a FIS-update-contract indication, if the FIS-ground-user is able to accept the contract in full then:

2.4.7.4.3.1 If the FIS-ground-user is not able to send the first FIS-report within the response time specified in 2.4.7.2.2., it shall:

- a) invoke the FIS-update-contract response with the *Result* parameter set to the abstract value “positive acknowledgment”, and
- b) invoke the FIS-report request later.

2.4.7.4.3.2 **Recommendation.**— *If the FIS-ground-user is not able to send the FIS report within the response time, it should invoke the FIS-update-contract response within the response time and then invoke the FIS-report request within 10 seconds.*

2.4.7.4.3.3 If the FIS-ground-user is able to send the FIS-report report within the response time specified in 2.4.7.2.2., it shall invoke the FIS-update-contract response with the following parameters:

- a) the *Result* parameter set to the abstract value “accepted”, and
- b) the FIS report in the *FIS Information* parameter.

2.4.7.4.3.4 If the FIS-ground-user does not support the update-contract function but the requested FIS report is available, it shall invoke the FIS-update-contract response with the following parameters:

- a) the *Result* parameter set to the abstract value “rejected”,
- b) the *RejectReason* parameter set to the abstract value “update contract function not supported by the FIS-ground-user”, and
- c) the FIS report in the *FISInformation* parameter.

2.4.7.4.4 Upon receipt of a FIS-update-contract indication, if the FIS-ground-user cannot comply with the update contract request, it shall reject the contract by invoking a FIS-update-contract response with the following parameters:

- a) the abstract value “rejected” as *Result* parameter, and
- b) one of the following abstract values “cannot comply”, “FIS service not available”, “error detected in the FIS request” or “undefined”, as *RejectReason* parameter.

2.4.7.4.5 The FIS-ground-user shall invoke the FIS-report request each time the FIS information requested in the FIS-update-contract indication is updated until such time as the contract is canceled or aborted.

2.4.7.4.6 The FIS-air-user shall be allowed to reuse the value of a *FISContractNumber* used in a FIS-update-contract request for a new FIS contract:

- a) upon receipt of a FIS-update-contract confirmation with a *Result* parameter containing the abstract value “rejected”,
- b) upon receipt of a FIS-user-abort indication or a FIS-provider-abort indication,
- c) upon receipt of a FIS-cancel-update confirmation or a FIS-cancel-contracts confirmation,
- d) upon receipt of a FIS-cancel-update indication, or
- e) after invocation of a FIS-user-abort request.

2.4.7.5 Air And Ground Cancellation Of a Single FIS Update Contract

Note 1.— There is no provision for the cancellation of a single FIS demand contract.

Note 2.— Both FIS-air-user and FIS-ground-user are capable of initiating the FIS-cancel-update-contract service.

Note 3.— When invoking the FIS-cancel-update-contract request, if the FIS-air-user does not provide in the FISContractNumber parameter a value identifying an active FIS update contract, the request is rejected.

Note 4.— If the FIS-air-user invokes a FIS-cancel-update-contract request before the very first request (either a FIS-demand-contract or a FIS-update-contract-request) has been confirmed, the FIS-cancel-update-contract request will be rejected.

2.4.7.5.1 Upon receipt of a FIS-cancel-update-contract indication, the FIS-user shall cancel the FIS update contract identified by the *FISContractNumber* parameter.

2.4.7.6 Air cancellation of FIS contracts of the same type

Note 1.— Only the FIS-air-user is capable of initiating the FIS-cancel-contracts service.

Note 2.— If the FIS-air-user invokes a FIS-cancel-contracts request before the very first request (either a FIS-demand-contract or a FIS-update-contract-request) has been confirmed, the FIS-cancel-contracts request will be rejected.

2.4.7.6.1 When the FIS-ground-user receives a FIS-cancel-contracts indication, it shall cancel any FIS demand contract and FIS update contract for the type(s) identified in the indication (e.g. ATIS) in force with that aircraft.

2.4.7.7 FIS-user-abort

Note 1.— Both FIS-air-user and FIS-ground-user are capable of initiating the FIS-user-abort service.

Note 2.— If an active FIS-user requires to stop brutally any activity of the current FIS-ASE with the peer FIS-ASE with a possible loss of FIS messages currently in transit in the communication system, it invokes the FIS-user-abort request.

Note 3.— After the FIS-user-abort request is invoked, the FIS-user is no more considered by the FIS-AE as an active user.

2.4.7.8 Parameter Value, Unit, Range and Resolution

2.4.7.8.1 A FIS-user shall interpret variable unit, range and resolution as defined in 2.4.4.

2.4.8 SUBSETTING RULES

2.4.8.1 General

Note.— 2.4.8 specifies conformance requirements which all implementations of the FIS protocol obey.

2.4.8.1.1 An implementation of either the FIS ground based service or the FIS air based service claiming conformance to 2.4 shall support the FIS protocol features as shown in the tables below.

Note.— The ‘status’ column indicates the level of support required for conformance to the FIS-ASE protocol described in this part. The values are as follows:

- a) **‘M’** mandatory support is required,
- b) **‘O’** optional support is permitted for conformance to the FIS protocol,
- c) **‘N/A’** the item is not applicable, and
- d) **‘C.n’** the item is conditional where *n* is the number which identifies the condition which is applicable.

Table 2.4.8-1. FIS Protocol Versions Implemented

	Status	Associated Predicate
Version 1	M	none

Table 2.4.8-2. FIS Protocol Options

	Status	Associated Predicate
FIS-air-ASE	C.1	FIS/air
FIS-ground-ASE	C.1	FIS/ground
FIS Update Contract Function supported by the FIS-ground-user	if (FIS/ground) O else N/A	FIS-user/UC
FIS Update Contract supported by the FIS-air-ASE	if (FIS/air) O, else N/A	UC-FU

C.1: a conforming implementation will support one and only one of these two options.

Table 2.4.8-3/a. FIS-air-ASE conformant configurations

	List of Predicates	Functionality Description
I	FIS/air	a FIS-air-ASE supporting the FIS demand contract (core functions)
II	FIS/air + UC-FU	a FIS-air-ASE supporting core functions and the FIS update contract

Note.— A FIS air system may or may not support the cancel contracts feature.

Table 2.4.8-3/b. FIS-ground-ASE conformant configurations

	List of Predicates	Functionality Description
I	FIS/ground	a FIS-ground-ASE fully supporting the FIS demand contract and functionality for receiving and indicating that the FIS Update Contract is not supported
II	FIS/ground + FIS-user/UC	a FIS-ground-ASE fully supporting both types of FIS contract

Note.— A FIS ground system must support the cancel contracts feature.

Sub-Volume III

Ground-Ground Applications

NOTE ON THE SECOND EDITION

The list below shows the parts of this sub-volume that are different from similar parts of the first edition. It also shows the parts of the first edition that have been deleted and thus no longer appear in this edition.

Reference	Nature of change
3.1.2.2.1.1.3	Addition
3.1.2.2.2.1.3	Addition
3.1.2.3.4.3.2.4 a) and b)	Modification
3.2.5.3.2.17.1.1	Modification
3.2.5.3.2.17.2.1	Modification
3.2.5.3.4.3.2.1 a)	Addition
3.2.5.3.4.3.2.1 b)	Modification
3.2.5.3.5.2.1.2	Modification
3.2.5.3.5.2.2.1	Modification
3.2.5.3.5.2.2.2	Addition
3.2.6.1.10.2.1 g)	Modification
3.2.6.1.14.2.1 e)	Modification
Table 3.2.6-4	Modification
3.2.7.1.1 (Frequency, FrequencyVHF)	Modification

3.1 ATS MESSAGE HANDLING SERVICES (ATSMHS)

3.1.1 INTRODUCTION

The ATS (Air Traffic Services) Message Handling Services (ATSMHS) applications allow ATS Messages to be exchanged between service users. The ATS Message Handling Services are the ATS Message Service and the ATN Pass-Through Service.

Note 1.— These ATS Message Handling Services aim at providing generic message services over the Aeronautical Telecommunication Network (ATN) Internet. They may in turn be used as a communication system by user-applications communicating over the ATN. This may be achieved e.g. by means of application program interfaces to either the ATS Message Service or to the ATN Pass-Through Service.

Note 2.— ATS Message Service

- a) *The ATS Message Service is provided by the implementation over the ATN Internet Communication Services of the Message Handling Systems specified in ISO/IEC (International Organization for Standardization/ International Electrotechnical Commission) 10021 and CCITT (Consultative Committee of International Telegraph and Telephone) or ITU-T (International Telecommunication Union - Telecommunications Standards) X.400, and complemented with the additional requirements specified in 3.1. The two sets of documents, the ISO/IEC MOTIS (Message-Oriented Text Interchange System) International Standards and the CCITT X.400 Series of Recommendations (1988 or later) are in principle aligned to each other. However there are a small number of differences. In 3.1 reference is made to the relevant ISO International Standards, and International Standardized Profiles (ISP) where applicable. Where necessary, e.g. for reasons of interworking or to point out differences, reference is also made to the relevant X.400 Recommendations.*
- b) *Two levels of service are intended to be defined within the ATS Message Service:*
 - i) *the Basic ATS Message Service.*
 - ii) *the Extended ATS Message Service.*
- c) *This specification of the ATS Message Service supports only the Basic ATS Message Service. The Extended ATS Message Service could be incorporated in future packages.*

Note 3.— The ATN Pass-Through Service is the ATS Message Handling Service offered over the ATN Internet Communication Services by the use of the Dialogue Service and of the associated ATN upper layer architecture to exchange AFTN (Aeronautical Fixed Telecommunication Network) Messages formatted in IA-5 (International Alphabet No 5) in compliance with the provisions of Annex 10, Volume II.

Note 4.— End systems performing ATS Message Handling Services

- a) *Four types of ATN End Systems are defined in 3.1:*
 - 1) *an ATS Message Server,*
 - 2) *an ATS Message User Agent,*
 - 3) *an AFTN/AMHS Gateway (Aeronautical Fixed Telecommunication Network / ATS Message Handling System), and*
 - 4) *an AFTN/ATN Type A Gateway.*
- b) *Connections may be established over the Internet Communications Service between any pair constituted of these ATN End Systems and listed in Table 3.1.1-1. Although included in Table 3.1.1-1, the communication between an ATS Message Server and an ATS Message User Agent is not specified in 3.1.*

**Table 3.1.1-1. Communications between ATN End Systems implementing
ATS Message Handling Services**

<i>ATN End System 1</i>	<i>ATN End System 2</i>
<i>ATS Message Server</i>	<i>ATS Message Server</i>
<i>ATS Message Server</i>	<i>AFTN/AMHS Gateway</i>
<i>ATS Message Server</i>	<i>ATS Message User Agent</i>
<i>AFTN/AMHS Gateway</i>	<i>AFTN/AMHS Gateway</i>
<i>AFTN/ATN Type A Gateway</i>	<i>AFTN/ATN Type A Gateway</i>

Note 5.— Structure of 3.1

- a) *3.1.1: INTRODUCTION contains the purpose and structure, and a summary of the functionalities offered by the ATS Message Handling Services.*
- b) *3.1.2: ATS MESSAGE SERVICE contains three sections as follows:*
 - 1) *3.1.2.1: System Level Provisions, provides a high level specification of the application and of the environment in which it operates;*

- 2) 3.1.2.2: *ATS Message Service Specification*, provides the detailed specification of the service and protocol requirements for each type of ATN End System (ATS Message User Agent and ATS Message Server) implementing the ATS Message Service;
 - 3) 3.1.2.3: *AFTN/AMHS Gateway Specification*, provides the detailed specification of an AFTN/AMHS Gateway and of the related functional requirements such as conversion.
- c) 3.1.3: *ATN PASS-THROUGH SERVICE* contains three sections as follows:
- 1) 3.1.3.1: *System Level Provisions*, provides a high level specification of the application and of the environment in which it operates;
 - 2) 3.1.3.2: *ATN Pass-Through Service Specification*, provides the detailed specification of the protocol requirements between two AFTN/ATN Type A Gateways implementing the ATN Pass-Through Service;
 - 3) 3.1.3.3: *AFTN/ATN Type A Gateway Specification*, provides the detailed specification of an AFTN/ATN Type A Gateway and of the related functional requirements.

Note 6.— The following terminology applies in 3.1:

- a) **AFTN acknowledgement message:** an AFTN service message acknowledging the receipt of an AFTN message which priority indicator has the value “SS”.
- b) **direct AMHS user:** an ATS Message Service user who engages in the ATS Message Service at an ATS Message User Agent. A direct AMHS user may belong to two subgroups as follows:
 - 1) human users who interact with the ATS Message Service by means of an ATS Message User Agent connected to an ATS Message Server; and
 - 2) host users which are computer applications running on ATN end systems and interacting with the ATS Message Service by means of application programme interfaces.
- c) **indirect AMHS user:** an ATS Message Service user at an AFTN station using an AFTN/AMHS Gateway to communicate with other ATS Message Service users.
- d) **subject AFTN message:** an AFTN message which causes an AFTN service message or an AMHS report to be generated.
- e) **subject AMHS message:** an AMHS message which causes an AFTN service message or an AMHS report to be generated.
- f) **subject IPM:** the IPM which is the content of an AMHS message and which causes an AMHS Receipt Notification to be generated.

- g) **unknown address AFTN service message:** an AFTN service message requesting correction by the originator of a message received with an unknown addressee indicator.

Note 7.— The classifications defined in the ISPs apply for expressing conformance requirements - i.e. static capability - in 3.1. The ISP classifications refine the ISO/IEC 9646-7 classification to include different levels of mandatory support, depending on the level of functionality to be supported by the considered Message Handling System. These classifications include the following elements, of which the complete definition may be found in each referenced ISP:

- a) **mandatory full support (M).**
- b) **mandatory minimal support (M-).**
- c) **mandatory O/R name minimal support (M1)** (see ISO/IEC ISP 12062-2).
- d) **optional support (O).**
- e) **conditional support (C).**
- f) **out of scope (I).**
- g) **not applicable (-).**

Note 8.— The following classification applies for expressing dynamic behaviour requirements - i.e. the action performed by the ATN end system - related to parameters or elements in the Profile Requirement Lists (PRLs) included in 3.1.2.3, for the specification of the AFTN/AMHS Gateway:

- a) **generated (G):** used to describe the generation of an AMHS or AFTN information object. It means that the element is generated by the AFTN/AMHS Gateway, and that its value does not depend on the value of an element of the information object received by the AFTN/AMHS Gateway which caused the current generation of an information object, but that the value of the element is based on parameters related to the AFTN/AMHS Gateway itself or takes a pre-determined value. If an element comprises several components, then the element is classified as generated if at least one of its components is generated, and the others are either generated or excluded;
- b) **optionally generated (G1):** used with the same meaning as “generated”, with the exception that the generation of the element is optional, the decision being a matter of policy local to the Management Domain operating the AFTN/AMHS Gateway;
- c) **conditionally generated (G2):** used only to describe the generation of an AMHS report or RN (Receipt Notification) element. It means, for a report generation, that the element is generated in the report or RN based on some condition related to the subject AMHS message being true. If the element is generated, it takes a value derived from elements present in the received AMHS information object which caused the generation of the report or RN;

- d) **translated (T)**: used to describe either the generation of an AMHS or AFTN information object or the use of a received information object. It means that the element is translated by the AFTN/AMHS Gateway, using a dependence relationship between the value of an element of the received information object and the value of the translated element in the generated information object. If an element comprises several components, then the element is classified as translated if at least one of its components is translated, and the others are either generated or excluded in generation, discarded or out of scope in reception;
- e) **conditionally translated (T1)**: used with the same meaning as “translated”, with the exception that the translation of the element is subject to some condition being true, e.g. the presence of an optional element in the received information object;
- f) **discarded (D)**: used to describe the use of a received AMHS or AFTN information object. It means that the value of the element is not used by the Message Transfer and Control Unit when generating the elements of the information object converted from the received information object, and that the semantic information conveyed in the element is discarded during the process of conversion in the AFTN/AMHS Gateway. However the presence or value of the element may be used by the Message Transfer and Control Unit for purposes other than conversion, such as report generation and logging;
- g) **excluded (X)**: used to describe either the generation of an AMHS or AFTN information object or the use of a received information object. Upon generation of an information object, it means that the element is not used nor present in the generated information object. Upon reception of an AMHS information object, it means that the presence of the element causes rejection of the information object, and generation of an AMHS non-delivery report as appropriate;
- h) **out of scope or not-applicable (-)**: used to describe the use of a received information object, when the element is either a format element which cannot be processed in any way or an element which is not in the scope of the section, but which presence is included in the ISPICS (ISP Implementation Conformance Statement) serving as a basis for the mapping specification.

Note 9.— Application Functionalities

- a) The Basic ATS Message Service meets the basic requirements of the Message Handling Systems Profiles published by ISO/IEC as ISPs (International Standardized Profiles), and it incorporates additional features to support the service offered by the AFTN. The Basic ATS Message Service is further specified in 3.1.2.2. This includes the specification of which ISPs apply in this context.
- b) The ATN Pass-Through Service encapsulates and decapsulates AFTN messages at an AFTN/ATN type A Gateway, using the Dialogue Service and the associated upper layer protocol architecture. The ATN Pass-Through Service is further specified in 3.1.3.2.

Note 10.— Applicability

- a) *The implementation of the ATS Message Service is mandatory for conformance with 3.1. However, as a matter of organisations' policy, interim conformance may be achieved with the implementation of the ATN Pass-Through Service. The choice to implement the ATN Pass-Through Service as an interim solution does not replace the requirement to implement the ATS Message Service at the earliest possible date.*
- b) *The interoperability between the ATS Message service and the ATN Pass-Through Service is a local implementation matter, provided that such an implementation has an external behaviour identical to that of an AFTN/AMHS Gateway and of an AFTN/ATN Type A Gateway, as appropriate. The choice to implement the ATN Pass-Through Service implies the requirement to provide the interoperability facilities to the ATS Message Service implementations.*

3.1.2 ATS MESSAGE SERVICE

3.1.2.1 System level provisions

The ATS Message Service shall be implemented for conformance with 3.1.

3.1.2.1.1 ATS Message Service Users

3.1.2.1.1.1 Direct AMHS users shall use the Basic ATS Message Service at an ATS Message User Agent.

3.1.2.1.1.2 Indirect AMHS users shall use only that part of the Basic ATS Message Service which corresponds to AFTN functionalities, by using the interworking capability provided by an AFTN/AMHS Gateway as specified in 3.1.2.3.

3.1.2.1.2 AMHS Model

3.1.2.1.2.1 AMHS functional model

3.1.2.1.2.1.1 Model components

The systems comprising the AMHS shall themselves be comprised of the following functional objects, the general role of which is described in ISO/IEC 10021-2:

- a) message transfer agent(s) (MTA),
- b) user agent(s) (UA),
- c) message store(s) (MS), and
- d) access unit(s) (AU).

Note.— The ISO/IEC 10021 Elements of Service and Protocols used by these functional objects are specified in 3.1.2.2 and 3.1.2.3.

3.1.2.1.2.1.2 ATS Message Server

An ATS Message Server shall include a MTA and optionally one or several MSs, as specified in 3.1.2.2.2.

3.1.2.1.2.1.3 ATS Message User Agent

An ATS Message User Agent shall include a UA as specified in 3.1.2.2.1.

3.1.2.1.2.1.4 AFTN/AMHS Gateway

An AFTN/AMHS Gateway shall include a MTA, which is part of the ATN Component of the AFTN/AMHS Gateway, and an AU, as specified in 3.1.2.3.

Note.— The AU is the Message Transfer and Control Unit of the AFTN/AMHS Gateway.

3.1.2.1.2.2 AMHS information model

The following three categories of AMHS information objects shall be used:

- a) messages;
- b) probes; and
- c) reports.

3.1.2.1.2.2.1 Messages

Note.— The provisions in 3.1.2 concerning ISO/IEC 10021 envelopes apply to Transfer Envelopes only.

In the Basic ATS Message Service, each AMHS message shall correspond unequivocally to an ATS Message.

3.1.2.1.2.2.2 Probes

Only direct AMHS users shall be able to submit AMHS probes.

3.1.2.1.2.2.3 Reports

AMHS reports shall be delivered only to direct AMHS users.

3.1.2.1.2.3 Security and management models

Recommendation.— *In the Basic ATS Message Service, security should be obtained by procedural means rather than by technical features inherent to the AMHS.*

Note 1.— In the Basic ATS Message Service, the security at each ATS Message Server or AFTN/AMHS Gateway is deemed a local issue to be addressed by the authority in charge of the system.

Note 2.— In the Basic ATS Message Service, management is limited to the logging provisions which are defined for the ATS Message User Agent, for the ATS Message Server and for the AFTN/AMHS Gateway. No provision is made for retrieval or exchange of this information, which is deemed a local issue to be addressed by the authority in charge of the system.

3.1.2.1.3 Organization of the AMHS

The AMHS shall be organisationally composed of AMHS Management Domains.

Note 1.— An AMHS Management Domain may elect to operate as either an ADMD (Administration Management Domain) or a PRMD (Private Management Domain), depending on the national telecommunications regulation in force in the country(ies) where it operates and on its relationships with other Management Domains.

Note 2.— A PRMD which is subordinate to one or several AMHS ADMDs may qualify as AMHS Management Domain if it satisfies the provisions of 3.1.2.

3.1.2.1.4 AMHS Management Domain configurations

3.1.2.1.4.1 Minimal set of systems

The minimal set of systems implemented and operated by an AMHS Management Domain shall be one of the following:

- a) an ATS Message Server and one or several ATS Message User Agents;
- b) an AFTN/AMHS Gateway; or
- c) any combination of a) and b).

3.1.2.1.4.2 Interconnection between two AMHS Management Domains

An interconnection between two AMHS Management Domains shall be implemented as one of the following:

- a) a connection between two ATS Message Servers;
- b) a connection between an ATS Message Server and an AFTN/AMHS Gateway; or
- c) a connection between two AFTN/AMHS Gateways.

3.1.2.1.5 Naming and addressing principles

3.1.2.1.5.1 AMHS Naming and Addressing

3.1.2.1.5.1.1 AMHS O/R Names

For the support of the Basic ATS Message Service, the O/R (originator/recipient) name of an AMHS user shall comprise:

- a) the O/R address of the AMHS user, called an MF-address; and
- b) optionally the directory name of the AMHS user, if the policy of the AMHS Management Domain, to which the AMHS user belongs, includes the local support of directory-names.

Note.— As a matter of policy local to an AMHS Management Domain, the directory name component of an O/R name may be used by the implementation of the Optional DIR (Use of Directory) FG (Functional Group).

3.1.2.1.5.1.2 Structure of a MF-Address

The MF-Address (MHS-form address) of an AMHS user shall comprise:

- a) a set of attributes as specified in 3.1.2.1.5.1.3, identifying the AMHS Management Domain of which the AMHS user, either direct or indirect, is a service-user; and
- b) a set of attributes as specified in 3.1.2.1.5.1.4, identifying uniquely the AMHS user within the AMHS Management Domain, in compliance with the AMHS addressing scheme implemented by the AMHS Management Domain.

Note.— The attributes present in the identifier defined in item b) may include any standard or domain-defined attribute as specified in section 18 of ISO/IEC 10021-2, other than country-name, administration-domain-name and private-domain-name.

3.1.2.1.5.1.3 AMHS Management Domain identifier

The attributes identifying an AMHS Management Domain shall include the following standard attributes as specified in ISO/IEC 10021-2, section 18.3, depending on the status under which the AMHS Management Domain has elected to operate:

- a) *country-name*,
- b) *administration-domain-name*,
- c) *private-domain-name*, if the AMHS Management Domain has elected to operate as a PRMD.

3.1.2.1.5.1.4 AMHS Addressing Schemes

3.1.2.1.5.1.4.1 General provisions

Note 1.— It is a matter of policy local to each AMHS Management Domain to implement either a locally defined AMHS Addressing Scheme, or a Common AMHS Addressing Scheme, or a combination of these. The single Common ICAO AMHS Addressing Scheme defined in the present version of this document is the XF-Addressing Scheme. Aeronautical Industry X.400 Addressing Schemes are defined in appropriate Aeronautical Industry Standards.

Note 2.— Each AMHS Addressing Scheme includes the set of attributes identifying the AMHS Management Domain as specified in 3.1.2.1.5.1.3.

3.1.2.1.5.1.4.2 XF-Addressing Scheme

The XF-Address (translated address) of a direct or indirect AMHS user shall be composed exclusively of the following:

- a) an AMHS Management Domain identifier as specified in 3.1.2.1.5.1.3;

- b) an *organization-name* attribute:
 - 1) as specified in ISO/IEC 10021-2, Section 18.5,
 - 2) taking the 4-character value “AFTN”, and
 - 3) encoded as a Printable String; and
- c) an *organizational-unit-names* attribute:
 - 1) as specified in ISO/IEC 10021-2, Section 18.5,
 - 2) comprising a sequence of one single element, which takes the 8-character alphabetical value of the AF-Address (AFTN-form address) of the user, and
 - 3) encoded as a Printable String.

Note 1.— An XF-Address is a particular MF-Address of which the attributes identifying the user within an AMHS Management Domain (i.e. those attributes other than country-name, administration-domain-name and private-domain-name) may be converted by an algorithmic method to and from an AF-Address. The algorithmic method requires the additional use of look-up tables which are limited, i.e. which include only a list of AMHS Management Domains rather than a list of individual users, to determine the full MF-address of the user.

Note 2.— No distinction is made between upper case and lower case.

3.1.2.1.5.2 Upper Layer Naming and Addressing

3.1.2.1.5.2.1 Application Process Titles

3.1.2.1.5.2.1.1 **Recommendation.**— *The Application Process Title of an ATS Message Server should be as specified in 4.3.3.2.*

3.1.2.1.5.2.1.2 **Recommendation.**— *The Application Process Title of an AFTN/AMHS Gateway should be as specified in 4.3.3.2.*

3.1.2.1.5.2.1.3 **Recommendation.**— *The Application Process Title of an ATS Message User Agent should be as specified in 4.3.3.2.*

3.1.2.1.5.2.2 Application Entity Qualifiers

3.1.2.1.5.2.2.1 **Recommendation.**— *The Application Entity Qualifier of an ATS Message Server should be AMS (7).*

3.1.2.1.5.2.2.2 **Recommendation.**— *The Application Entity Qualifier of an AFTN/AMHS Gateway should be GWB (8).*

3.1.2.1.5.2.2.3 **Recommendation.**— *The Application Entity Qualifier of an ATS Message User Agent should be AUA (9).*

3.1.2.1.5.2.3 Transport, Session and Presentation Addresses

The TSAP (Transport Service Access Point) of an ATS Message Server or of an ATS Message User Agent shall comply with the provisions of 5.4.

Note 1.— The assignment of a transport selector value is a matter local to an AMHS Management Domain.

Note 2.— The format and encoding of a session selector in the AMHS is specified in ISO/IEC ISP 11188-1, section 9.3.

Note 3.— The assignment and administration of session selectors is a matter local to an AMHS Management Domain.

Note 4.— The format and encoding of a presentation selector in the AMHS is specified in ISO/IEC ISP 11188-1, section 7.2.

Note 5.— The assignment and administration of presentation selectors is a matter local to an AMHS Management Domain.

3.1.2.1.6 AMHS Routing and rerouting

3.1.2.1.6.1 The definition of AMHS routing shall be subject to multilateral agreements.

3.1.2.1.6.2 The MTAs implemented by an AMHS Management Domain shall be collectively able to route on *country-name*, *ADMD-name*, *PRMD-name*, *organization-name* and *organizational-units-name* attributes.

3.1.2.1.7 AMHS Traffic logging upon origination

An AMHS Management Domain shall be responsible for long-term logging of all messages in their entirety which are originated by its direct AMHS users, for a period of at least thirty days.

Note.— This requirement implies the logging of the entire BER-encoded ASN.1 messages.

3.1.2.2 ATS Message Service Specification

3.1.2.2.1 ATS Message User Agent Specification

Note.— For the support of the Basic ATS Message Service, an ATS Message User Agent complies with:

- a) *profile AMH21 as specified in ISO/IEC ISP 12062-2;*
- b) *the requirements of Repertoire Group A, for messages including a body part whose type is an Extended Body Part Type of general-text-body-part type;*

- c) *the additional provisions relating to parameters generated at an ATS Message User Agent, as specified in 3.1.2.2.1.1; and*
- d) *the provisions related to traffic logging as specified in 3.1.2.2.1.2.*

3.1.2.2.1.1 Additional provisions on parameters

3.1.2.2.1.1.1 Message Content Profile Specification

In an ATS Message User Agent, the content of the Inter-Personal Messages conveyed in support of the Basic ATS Message Service shall conform to the basic requirements of AMH21 as specified in Clause A.1 of ISO/IEC ISP 12062-2, Annex A and to the additional requirements described in Table 3.1.2-1 which are specific to the Basic ATS Message Service.

Note 1.— Table 3.1.2-1 specifies the additional requirements in the form of a PRL (Profile Requirement List) expressing restrictions to a set of rows of the AMH21 profile, which are referred to using their reference in ISO/IEC ISP 12062-2.

Note 2.— There is no profile specification for the ATS Message User Agent at the level of the access protocol, i.e. at the level of the communication with the associated ATS Message Server, as this is considered to be a matter local to each AMHS Management Domain. If it is desired to use standard ISO/IEC 10021 protocols for this communication, then profile AMH23 (for P3) or profile AMH24 (for P7) as specified in ISO/IEC ISP 12062-4 or ISO/IEC ISP 12062-5, respectively, may be implemented.

Note 3.— The use of the ia5-text body part as specified in Table 3.1.2-1/AMH21/A1.3/1 ensures operability with both 1984 and 1988 IPM (Inter-Personal Message) UAs for the exchange of unstructured character data.

Table 3.1.2-1. Requirements specific to the Basic ATS Message Service in addition to profile AMH21

Ref	Element	Origination		Reception		Basic ATS Message Service Support	ATN reference	ISP 12062-2 Notes/References
		Base	ISP	Base	ISP			
Part 1 : AMH21/A.1.3 IPM body								
1	ia5-text	O	O	O	M	O/M		
1.2	data	M	M	M	M	M/M	3.1.2.2.3.2	

Ref	Element	Origination		Reception		Basic ATS Message Service Support	ATN reference	ISP 12062-2 Notes/References
		Base	ISP	Base	ISP			
Part 2 : AMH21/A.1.3.1Extended body part support								
1	ia5-text-body-part	O	O	O	M	O/M		see AMH21/ A.1.3/1
11	general-text-body-part	O	M	O	M	M/M	3.1.2.2.3.2 and Table 3.1.2-1 Part 4	
Part 3 : AMH21/A.1.5Common data types								
1	RecipientSpecifier							
1.2	notification-requests	O	O	M	M	M/M	3.1.2.2.3.3	
1.2.1	rn	O	O	O	O	M/M	3.1.2.2.3.3	
1.2.2	nrn	O	O	M	M	M/M		
2	ORDescriptor							
2.1	formal-name	M	M1	M	M1	M1/M1	3.1.2.2.3.1	
Part 4 : AMH21/A.1.3.2General text repertoire support								
1	Basic (ISO 646) (repertoire identifiers {1, 6})	M	M	M	M	M/M		Repertoire Group A
2	Basic-1 (ISO 8859-1) (repertoire identifiers {1, 6, 100})	O	M	O	M	O/O		Repertoire Group B

Legend : see 3.1.1

M = mandatory support

M1 = mandatory O/R name minimal support

O = optional support

3.1.2.2.1.1.2 Additional requirements upon MT-Elements of Service at an ATS Message User Agent

For the support of the Basic ATS Message Service, the *priority* element of an AMHS Message generated at an ATS Message User Agent shall take the value “urgent” if, and only if, the value of the priority-indicator in the ATS-Message-Priority as specified in 3.1.2.2.3.2.1 is “SS”.

3.1.2.2.1.1.3 Interpretation of UTC Time values

When generating and interpreting UTC Time values, an ATS Message User Agent shall associate dates up to ten years prior to the current time and up to forty years ahead of the current time with the corresponding century, with the interpretation of the remaining 49 values being implementation dependent.

Note.— This requirement is aligned on the convention used in ISO 10021-4:1997/Cor. 1:1998 and in ISO 10021-7:1997/Cor. 1:1998 for equivalent purposes.

3.1.2.2.1.2 Traffic logging requirements at an ATS Message User Agent

Note.— The requirement expressed in 3.1.2.1.7 may be implemented in the ATS Message User Agent.

3.1.2.2.2 ATS Message Server Specification

Note.— For the support of the Basic ATS Message Service, an ATS Message Server complies with:

- a) *the profile specification expressed in 3.1.2.2.2.1; and*
- b) *the provisions related to traffic logging as specified in 3.1.2.2.2.2.*

3.1.2.2.2.1 Profile Specification

3.1.2.2.2.1.1 Upper Layer Requirements

In an ATS Message Server, the Message Transfer (P1) implementation of the IPM Service in support of the Basic ATS Message Service shall conform to:

- a) *the basic requirements of AMH22 as specified in Clause B.1 of ISO/IEC ISP 12062-3, Annex B; and*
- b) *the additional requirements described in Clause B.2.2. for the support of the IPM Distribution List Functional Group.*

Note 1.— This in turn places no requirements concerning the P1 implementation other than:

- a) *the basic requirements of AMH11 specified for Common Messaging in annex A.1 of ISO/IEC ISP 10611-3, implying the mandatory support of the AMH111 Profile implementing the mts-transfer application context; and*
- b) *the additional requirements specified for the Common Messaging DL (Distribution List) Functional Group in annex A.2.2 of ISO/IEC ISP 10611-3.*

Note 2.— As a consequence of Note 2 in 3.1.2.2.1.1.1, the optional implementation of Message Stores (MS) in an ATS Message Server, being related to the access protocol from an ATS Message User Agent to an ATS Message Server, is a matter local to each AMHS Management Domain.

Note 3.— The additional support by an ATS Message Server of the AMH112 Profile as specified in ISO/IEC ISP 10611-3, for conformance to CCITT X.400 in order to interconnect with public ADMDs is a matter of policy local to each AMHS Management Domain.

Note 4.— For the use of the Association Control Service Element (ACSE) by an AMHS application, the application-context name which is used as a parameter in an A-ASSOCIATE is defined in the base standards (see ISO/IEC 10021-6).

Note 5.— The specification in 3.1.2.2.2.1.1 places no requirements for the Reliable Transfer Service Element (RTSE) and for ACSE other than conformance with ISO/IEC ISP 10611-2 in accordance with the P1 application-context(s) for which conformance is claimed.

Note 6.— The specification in 3.1.2.2.2.1.1 places no requirements for the Presentation and Session Layers other than conformance with ISO/IEC ISP 10611-2 in accordance with the P1 application-context(s) for which conformance is claimed.

3.1.2.2.2.1.2 Use of the Transport Service

3.1.2.2.2.1.2.1 The Basic ATS Message Service shall make use of the Connection Mode Transport Service as specified in 5.5.

Note.— For the support of the Basic ATS Message Service, the use of the expedited data option at the establishment of the transport connection is a local matter which may depend on the implemented application-context.

3.1.2.2.2.1.2.2 For the support of the Basic ATS Message Service, transport connections shall be established over the ATN Transport Service between systems belonging to the AMHS using the Residual Error Rate (RER) abstract-value “high”.

3.1.2.2.2.1.2.3 For the support of the Basic ATS Message Service, transport connections shall be established over the ATN Transport Service between systems belonging to the AMHS using the Transport Connection Priority abstract-value “6”, which corresponds to the message category “flight regularity communications”.

3.1.2.2.2.1.2.4 For the support of the Basic ATS Message Service, transport connections shall be established over the ATN Transport Service between systems belonging to the AMHS using the value of the ATN Security Label as specified in 5.6, which corresponds to:

- a) the ATN Traffic Type “ATN Operational Communications”;
- b) the Sub-Type “Air Traffic Services Communications” (ATSC); and
- c) “No Traffic Type Policy Preference”.

3.1.2.2.1.3 Interpretation of UTC Time values

When generating and interpreting UTC Time values, an ATS Message Server shall associate dates up to ten years prior to the current time and up to forty years ahead of the current time with the corresponding century, with the interpretation of the remaining 49 values being implementation dependent.

Note.— This requirement is aligned on the convention used in ISO 10021-4:1997/Cor. 1:1998 for equivalent purposes.

3.1.2.2.2 Traffic logging requirements at an ATS Message Server

3.1.2.2.2.1 The ATS Message Server shall perform a long-term logging, for a period of at least thirty days, of the actions taken with respect to every message received at the ATS Message Server, whether from an ATS Message User Agent or from another ATS Message Server, and to every report received or generated at the ATS Message Server.

3.1.2.2.2.2 For the long-term logging of information related to a message submitted to or received by an ATS Message Server, the following parameters related to the message shall be logged:

- a) *message-identifier*;
- b) *priority*;
- c) *content-type*;
- d) *originator-name*;
- e) *recipient-name* elements on responsibility list;
- f) *message-content-size*;
- g) last element of the *trace-information* (if any);
- h) *arrival-time* or *submission-time*;
- i) *transfer destination* (if any);
- j) *transfer time* (if any);
- k) *this-recipient-name* (if message delivery is performed by the ATS Message Server);
- l) *delivery-time* (if any);
- m) *delivery* and/or *non-delivery* reports generated (if any); and
- n) *event date/time*.

Note.— The responsibility list identifies recipients whose perRecipientIndicator responsibility bit has the abstract-value “responsible”.

3.1.2.2.2.3 For the long-term logging of information related to a report generated or received by an ATS Message Server, the following parameters related to the report shall be logged:

- a) *report-identifier*;
- b) *subject-identifier*;
- c) *actual-recipient-name* elements;
- d) *report-type* elements;
- e) *report-destination-name*;
- f) last element of the *trace-information* (if any);
- g) *arrival-time* in the ATS Message Server or generation time;
- h) transfer destination (if any);
- i) transfer time (if any);
- j) *OR-name* of the report recipient (if report delivery is performed by the ATS Message Server);
- k) *delivery-time* (if any); and
- l) event date/time.

3.1.2.2.3 Parameters

3.1.2.2.3.1 AMHS Addresses

In the AMHS, the O/R address of a direct AMHS user belonging to an AMHS Management Domain shall be a MF-Address.

3.1.2.2.3.2 Text

The body of an Inter-Personal Message (IPM) shall comprise a single body part carrying IA-5 characters and structured as depicted in Table 3.1.2-2.

Note 1.— This body part structure and its components which are described in the subsequent clauses are specific to the Basic ATS Message Service.

Note 2.— This clause places no constraint on its implementation, which may take place at the level of the user-interface.

Table 3.1.2-2. Structure of an IPM in the Basic ATS Message Service

Ref	Element	Basic ATS Message Service Support		Value	IA-5 Encoding
		Orig	Rec		
1	ATS-Message-Header	M	M		
1.1	start-of-heading	M	M	(SOH)	(0/1)
1.2	ATS-Message-Priority	M	M		
1.2.1	priority-prompt	M	M	PRI:(single space)	(5/0)(5/2)(4/9)(3/10)(2/0)
1.2.2	priority-indicator	M	M	see 3.1.2.2.3.2.1	see 3.1.2.2.3.2.1
1.2.3	priority-separator	M	M	(CR)(LF)	(0/13)(0/10)
1.3	ATS-Message-Filing-Time	M	M		
1.3.1	filing-time-prompt	M	M	FT:(single space)	(4/6)(5/4)(3/10)(2/0)
1.3.2	filing-time	M	M	see 3.1.2.2.3.2.2	see 3.1.2.2.3.2.2
1.3.3	filing-time-separator	M	M	(CR)(LF)	(0/13)(0/10)
1.4	ATS-Message-Optional-Heading-Info	O	M		
1.4.1	OHI-prompt	M	M	OHI:(single space)	(4/15)(4/8)(4/9)(3/10)(2/0)
1.4.2	optional-heading-information	M	M	see 3.1.2.2.3.2.3	see 3.1.2.2.3.2.3
1.4.3	OHI-separator	M	M	(CR)(LF)	(0/13)(0/10)
1.5	end-of-heading-blank-line	M	M	(LF)	(0/10)
1.6	start-of-text	M	M	(STX)	(0/2)
2	ATS-Message-Text	M	M	see 3.1.2.2.3.2.4	see 3.1.2.2.3.2.4

Legend (see 3.1.1):

M = mandatory support
O = optional support

3.1.2.2.3.2.1 ATS Message Priority

Each message shall be assigned to one of five priority groups which are designated, and have the value of, the priority indicators SS, DD, FF, GG and KK.

3.1.2.2.3.2.2 ATS Message Filing Time

Each message shall include a filing-time element, designated as a date-time group consisting of six numerical characters, the first two digits representing the date of the month and the last four digits the hours and minutes in UTC.

3.1.2.2.3.2.3 ATS Message Optional Heading Info

3.1.2.2.3.2.3.1 It shall be possible to associate an optional heading information with each message.

3.1.2.2.3.2.3.2 The value of the optional-heading-information element shall comprise a character string with a maximum length of 54 characters.

3.1.2.2.3.2.4 ATS Message Text

The ATS-Message-Text element shall be composed of IA-5 characters with no further restriction.

3.1.2.2.3.3 Notification requests

The *notification-requests* element in a RecipientSpecifier in an IPM Heading shall take the abstract-value “rn” if, and only if, the value of the priority-indicator is “SS”.

Note.— This clause places no constraint on its implementation, which takes place at the level of the user-interface.

3.1.2.3 AFTN/AMHS Gateway Specification

3.1.2.3.1 General

3.1.2.3.1.1 An AFTN/AMHS Gateway shall provide for an interworking between the AFTN and the ATN such that communication with other AFTN/AMHS Gateways and with ATS Message Servers is possible.

3.1.2.3.1.2 An AFTN/AMHS Gateway shall consist of the four following logical components:

- a) AFTN Component;
- b) ATN Component;
- c) Message Transfer and Control Unit; and
- d) Control Position.

Note.— This division into logical components is a convenient way of specifying functions of a gateway. There is no requirement for an AFTN/AMHS Gateway to be implemented according to this structure.

3.1.2.3.1.3 An AFTN/AMHS Gateway shall be able to perform actions upon receipt of any category of AMHS information object by its ATN Component.

3.1.2.3.1.4 An AFTN/AMHS Gateway shall be able to perform actions upon receipt of any type of AFTN message by its AFTN Component.

3.1.2.3.2 AFTN/AMHS Gateway components

3.1.2.3.2.1 AFTN component

3.1.2.3.2.1.1 The AFTN component shall handle the interface to the AFTN and provide an interface to the Message Transfer and Control Unit, implementing:

- a) all the applicable requirements of Annex 10, Volume II in a manner so as to be indistinguishable from an operational AFTN station by the AFTN centre to which the gateway is connected; and
- b) additional requirements which are necessary due to the AFTN Component pertaining to an AFTN/AMHS Gateway.

3.1.2.3.2.1.2 If an AFTN/AMHS Gateway is connected to an AFTN centre which is capable of using only ITA-2 (International Telegraph Alphabet No 2) format, the AFTN component shall convert messages to/from the IA-5 format.

Note.— This allows the Message Transfer and Control Unit to use IA-5 characters internally, as specified in 3.1.2.3.2.3.2.

3.1.2.3.2.1.3 The AFTN Component shall incorporate an AFTN procedure handler providing for all AFTN functions prescribed for the interface to the AFTN.

3.1.2.3.2.1.4 When received by the AFTN Component, AFTN service messages as generally specified in Annex 10, Volume II, 4.4.1.1.9 and subclauses, shall be handled by the AFTN Component of the Gateway in one of four mutually exclusive manners, depending on the category of the service message:

- a) transfer to the Message Transfer and Control Unit to be processed as specified in 3.1.2.3.4 if the service message is an AFTN acknowledgement message, as specified in Annex 10, Volume II, 4.4.10.1.6.1 and 4.4.15.6;
- b) transfer to the Message Transfer and Control Unit to be processed as specified in 3.1.2.3.4 if the service message is an AFTN service message requesting correction of a message received with an unknown addressee indicator as specified in Annex 10, Volume II, 4.4.11.13.3;

- c) processing as specified in 3.1.2.3.2.1.12 if the service message is an AFTN service message requesting from the originator repetition of an incorrectly received message when it is detected that a message has been mutilated, as specified in Annex 10, Volume II, 4.4.11.1 and 4.4.16.2.2; or
- d) processing in compliance with the provisions of Annex 10, Volume II, without being passed to the Message Transfer and Control Unit, if the service message belongs to any other category of AFTN service message.

3.1.2.3.2.1.5 When received by an AFTN/AMHS Gateway, AFTN channel-check transmissions as specified in Annex 10, Volume II, 4.4.9.3 and 4.4.15.5 shall:

- a) be handled by the AFTN Component in compliance with the provisions of Annex 10, Volume II; and
- b) be prevented from being passed to the Message Transfer and Control Unit.

3.1.2.3.2.1.6 The AFTN Component shall pass all messages, other than those referred to in 3.1.2.3.2.1.4 c) and d), and in 3.1.2.3.2.1.5, received from the AFTN to the Message Transfer and Control Unit for processing as specified in 3.1.2.3.4, and provided that the conditions of 3.1.2.3.2.1.7 are met.

3.1.2.3.2.1.7 The processing by the AFTN Component shall ensure that all messages and service messages received from the AFTN and passed to the Message Transfer and Control Unit for further processing by the AFTN/AMHS Gateway are constructed in strict accordance with the provisions of Annex 10, Volume II, paragraphs 4.4.15.1 through 4.4.15.3.12 and 4.4.15.6.

3.1.2.3.2.1.8 The AFTN Component shall perform short-term retention of all messages transmitted towards the AFTN in a manner equivalent to that specified for an AFTN communication centre in Annex 10, Volume II, 4.4.1.7.

3.1.2.3.2.1.9 The AFTN Component shall perform long-term retention of the heading, address and origin parts of all messages received from the AFTN, with the message receipt-time and the action taken thereon, for a period of at least thirty days.

3.1.2.3.2.1.10 The AFTN Component shall perform long-term retention of all AFTN messages, in their entirety, that it generates, for a period of at least thirty days.

3.1.2.3.2.1.11 The AFTN Component shall perform long-term retention of the heading, address and origin parts of all messages received from the Message Transfer and Control Unit and the action taken thereon, for a period of at least thirty days.

3.1.2.3.2.1.12 Upon reception by an AFTN/AMHS Gateway of an AFTN service message requesting repetition by the originator of an incorrectly received message as specified in Annex 10, Volume II, 4.4.11.1 or 4.4.16.2.2, the AFTN Component shall perform one of the following actions:

- a) terminate the procedure and report an error situation to a control position if the referenced subject AFTN message did not pass through the gateway or if the AFTN Component is not in possession of an unmutated copy of the subject AFTN message; or

- b) reassume responsibility for the mutilated message and repeat the message in compliance with the provisions of Annex 10, Volume II, 4.4.11.3, if the mutilated message is detected as having passed through the gateway and if the AFTN Component is in possession of an unmutated copy of the message.

Note.— The determination whether the AFTN Component is in possession of an unmutated copy of the message, as mentioned in items a) and b) above, may require the assistance of a control position.

3.1.2.3.2.1.13 If, for any reason, the Message Transfer and Control Unit is unable to accept AFTN messages passed by the AFTN Component, then the AFTN Component shall handle this situation in compliance with the provisions of Annex 10, Volume II, 4.4.1.5.2.3.

Note.— Such a condition may be caused by the inability of the Message Transfer and Control Unit to pass AMHS messages to the ATN Component.

3.1.2.3.2.1.14 The AFTN Component shall ensure that all information objects constructed by the Message Transfer and Control Unit for transmission over the AFTN are handled in accordance with the AFTN procedure, in application of 3.1.2.3.2.1.3 above.

3.1.2.3.2.1.15 If the AFTN Component is unable to handle an AFTN service message or an AFTN channel-check transmission in compliance with the provisions of Annex 10, Volume II, as specified in 3.1.2.3.2.1.4 d) or 3.1.2.3.2.1.5, then the error condition shall be logged and reported to a control position.

3.1.2.3.2.1.16 An AFTN address shall be allocated to the AFTN Component.

3.1.2.3.2.2 ATN Component

3.1.2.3.2.2.1 The ATN Component shall allow the AFTN/AMHS Gateway to function as an end system on the ATN.

3.1.2.3.2.2.2 The ATN Component shall handle the interface to the AMHS, and provide an interface to the Message Transfer and Control Unit as specified in 3.1.2.3.2.4, implementing a MTA complying with the profile specification included in 3.1.2.2.2.1 so as to be externally indistinguishable from an ATS Message Server by the ATS Message Server(s) or other AFTN/AMHS Gateway(s) to which it is connected.

3.1.2.3.2.2.3 If, for any reason, the Message Transfer and Control Unit is unable to accept messages or probes passed by the ATN Component, then the ATN Component shall behave as follows:

- a) attempt to reroute the message or probe as specified in ISO/IEC 10021-4, 14.3.4.4;
- b) if no alternate route is available in the MTA-routing tables or all such routes cannot be successfully used, reject the message for all the message recipients, whose *responsibility* element in the *per-recipient-indicators* has the abstract-value “responsible” in the received message, with the *non-delivery-reason-code* and *non-delivery-diagnostic-code* elements of the non-delivery report taking the abstract-values specified in the base standards (ISO/IEC 10021-4, 14.3.4.4., item 1).

Note.— Such a condition may be caused by the inability of the Message Transfer and Control Unit to pass AFTN messages to the AFTN Component.

3.1.2.3.2.2.4 If the AMHS Management Domain operating an AFTN/AMHS Gateway desires to implement Message Handling System optional functional groups in addition to the specification of 3.1.2.3.2.2.2 above, this shall be performed in the ATN Component.

Note.— This applies in particular to the Redirection Functional Group. If implemented, redirection may be performed by the ATN Component, caused by a failure situation as envisaged in 3.1.2.3.2.2.3 above for example.

3.1.2.3.2.2.5 The ATN Component shall ensure that all information objects constructed by the Message Transfer and Control Unit for transfer in the AMHS are handled in accordance with the procedures specified in the base standards for a relaying MTA implementing the profile specified in 3.1.2.2.2.1, in application of 3.1.2.3.2.2.2 above.

3.1.2.3.2.2.6 The ATN Component shall implement a traffic logging function identical to that of the MTA included in an ATS Message Server as specified in 3.1.2.2.2.2.

3.1.2.3.2.2.7 The ATN Component shall ensure that all AMHS information objects passed to the Message Transfer and Control Unit comply with the base standards.

3.1.2.3.2.3 Message Transfer and Control Unit

3.1.2.3.2.3.1 The Message Transfer and Control Unit in an AFTN/AMHS Gateway shall provide a bi-directional conversion facility between the AFTN component and the ATN component, consisting of:

- a) a set of general functions as specified in 3.1.2.3.3; and
- b) AFTN/AMHS conversion functions as respectively specified in 3.1.2.3.4 for the AFTN to AMHS conversion and in 3.1.2.3.5 for the AMHS to AFTN conversion.

3.1.2.3.2.3.2 The Message Transfer and Control Unit shall use IA-5 characters internally.

3.1.2.3.2.3.3 The Message Transfer and Control Unit in an AFTN/AMHS Gateway shall pass all the AMHS information objects which it constructs in application of 3.1.2.3.4 and 3.1.2.3.5.6 to the ATN Component of the gateway, for further conveyance in the AMHS.

3.1.2.3.2.3.4 For the generation of AMHS messages and reports, and for the processing of received AMHS messages, probes and reports, the Message Transfer and Control Unit shall have the capability to interpret the semantics and to perform actions related to the ISO/IEC 10021 Elements of Service which are part of the basic requirements of the MT service as specified in ISO/IEC ISP 12062-3.

3.1.2.3.2.3.5 The Message Transfer and Control Unit in an AFTN/AMHS Gateway shall pass all the AFTN messages which it constructs in application of 3.1.2.3.5 and 3.1.2.3.4.2.1.4.2 to the AFTN Component of the AFTN/AMHS Gateway, for further conveyance in the AFTN.

3.1.2.3.2.3.6 The Message Transfer and Control Unit shall ensure that all the AMHS information objects which it constructs comply with section 7 (for IPMs) and section 8 (for RNs) of ISO/IEC 10021-7, complemented with the additional requirements included in Table 3.1.2-1, and with the section 12.2.1.1 of ISO/IEC 10021-4 (for messages) and section 12.2.1.3 of ISO/IEC 10021-4 (for reports).

3.1.2.3.2.3.7 The Message Transfer and Control Unit shall ensure that all the AFTN information objects which it constructs comply with Annex 10, Volume II, 4.4.15.

3.1.2.3.2.4 Interface between the ATN Component and the Message Transfer and Control Unit

3.1.2.3.2.4.1 The ATN Component shall exchange information objects with the Message Transfer and Control Unit via its MTA transfer-port as specified in ISO/IEC 10021-4, section 12.2.

3.1.2.3.2.4.2 The ATN Component shall invoke the Message-transfer, Report-transfer and Probe-transfer abstract operations, respectively, to pass AMHS messages, reports and probes to the Message Transfer and Control Unit.

3.1.2.3.2.4.3 The Message Transfer and Control Unit shall invoke the Message-transfer and Report-transfer abstract operations, respectively, to pass AMHS messages and reports to the ATN Component.

3.1.2.3.2.5 Interface between the AFTN Component and the Message Transfer and Control Unit

3.1.2.3.2.5.1 An AFTN message or service message passed by the AFTN Component to the Message Transfer and Control Unit in application of 3.1.2.3.2.1.4 items a) and b), 3.1.2.3.2.1.6 and 3.1.2.3.2.1.7 shall be:

- a) transferred according to the table of priorities as specified in Annex 10, Volume II, 4.4.1.2.1; and
- b) passed as received by the AFTN Component from the adjacent AFTN centre, with the possible exception of an ITA-2 to IA-5 conversion performed in application of 3.1.2.3.2.1.2, and including the unaltered AFTN heading if present in the received message.

3.1.2.3.2.5.2 An AFTN message or service message passed by the Message Transfer and Control Unit to the AFTN Component in application of 3.1.2.3.2.3.5 shall be:

- a) transferred according to the table of priorities as specified in Annex 10, Volume II, 4.4.1.2.1; and
- b) passed as constructed by the Message Transfer and Control Unit, and thus without message heading as specified in Annex 10, Volume II, 4.4.15.1.1.

3.1.2.3.2.5.3 The AFTN Component shall return to the Message Transfer and Control Unit, as the result of the transfer operation described in 3.1.2.3.2.5.2, the Transmission Identification, if any, constructed by the AFTN Component for the transmission of the message or service message over the AFTN.

3.1.2.3.2.6 AFTN/AMHS Gateway Control Position

3.1.2.3.2.6.1 The AFTN/AMHS Gateway Control Position shall be used as the place where errors which occurred in the AFTN/AMHS Gateway and certain non-deliveries which occurred in the AMHS are reported for appropriate action.

3.1.2.3.2.6.2 The appropriate action to be undertaken on reporting of an error or of a non-delivery to an AFTN/AMHS Gateway control position shall be either:

- a) a matter of policy which is local to the AMHS Management Domain operating the AFTN/AMHS Gateway; or
- b) subject to multilateral agreements.

Note.— For some categories of error situations, 3.1.2 specify the actions to be taken, e.g. message rejection and generation of an appropriate service message (to the AFTN) or non-delivery report (to the AMHS). The specified actions aim at minimizing the assistance of the control position. However it may be a matter of policy local to the AMHS Management Domain operating an AFTN/AMHS Gateway to try to reduce the occurrence of message rejection with the assistance of the control position.

3.1.2.3.2.6.3 When the action chosen to handle an error situation includes the generation of an AMHS information object, the category of information object used for this purpose shall be an IPM conveying appropriate service information.

Note 1.— The service information to be conveyed may be derived, for example, from an AFTN service message.

Note 2.— The presentation of the service information is a matter of local policy.

3.1.2.3.3 General functions

3.1.2.3.3.1 Traffic logging

3.1.2.3.3.1.1 The Message Transfer and Control Unit shall perform long-term logging, as specified in 3.1.2.3.3.1.2 to 3.1.2.3.3.1.6, for a period of at least thirty days, of information related to the following exchanges of information objects with the ATN Component and with the AFTN Component:

- a) AMHS message transfer out (to the ATN Component);
- b) AMHS report transfer out (to the ATN Component);
- c) AMHS message transfer in (from the ATN Component);
- d) AMHS report transfer in (from the ATN Component);
- e) AFTN message conveyance out (to the AFTN Component);
- f) AFTN message conveyance in (from the AFTN Component);
- g) AFTN service message indicating an unknown addressee indicator conveyance in (from the AFTN Component); and
- h) AFTN service message indicating an unknown addressee indicator conveyance out (to the AFTN Component).

3.1.2.3.3.1.2 For the long-term logging of information related to an AMHS Message Transfer In and AFTN message conveyance out, the following parameters, relating to the messages, shall be logged by the Message Transfer and Control Unit:

- a) input *message-identifier*;
- b) *IPM-identifier*, if any;
- c) *common-fields* and either *receipt-fields* or *non-receipt-fields* of IPN (Inter-Personal Notification), if any;
- d) action taken thereon (reject with *non-delivery-reason-code* and *non-delivery-diagnostic-code*, convert as AFTN message, convert as AFTN acknowledgement message, splitting due to number of recipients or message length, delivery report generation);
- e) event date/time;
- f) Origin line of converted AFTN message or service message, if any; and
- g) transmission identification of AFTN message(s) or service message(s), if returned by the AFTN Component.

3.1.2.3.3.1.3 For the long-term logging of information related to AFTN message conveyance in and AMHS Message Transfer Out, the following parameters, relating to the messages, shall be logged by the Message Transfer and Control Unit:

- a) Origin line of AFTN message (or AFTN acknowledgement message);
- b) transmission identification of AFTN message or service message, if any;
- c) action taken thereon (reject with rejection cause, convert as IPM, convert as RN, AFTN service message indicating an unknown addressee indicator generation);
- d) event date/time;
- e) *MTS-identifier*, if any; and
- f) *IPM-identifier*, if any.

3.1.2.3.3.1.4 For the long-term logging of information related to an AMHS Message Report In and/or AFTN Service Message indicating an unknown addressee indicator conveyance out, the following parameters, relating to the report and/or service message, shall be logged by the Message Transfer and Control Unit:

- a) *report-identifier* (if report in);
- b) *subject-identifier* (if report in);

- c) action taken thereon if report in (discard, convert into AFTN service message);
- d) event date/time;
- e) Origin line of converted AFTN service message (if service message out);
- f) Origin line of subject AFTN message (if service message out and no report in); and
- g) transmission identification of AFTN message or service message, if any.

3.1.2.3.3.1.5 For the long-term logging of information related to an AFTN Service Message indicating an unknown addressee indicator conveyance in and/or to an AMHS Message Report Out, the following parameters, relating to the service message and/or report, shall be logged by the Message Transfer and Control Unit:

- a) Origin line of converted AFTN service message (if service message in);
- b) Origin line of subject AFTN message (if service message in);
- c) transmission identification of AFTN message or service message, if any;
- d) action taken thereon if AFTN service message in (discard, convert into AMHS report);
- e) *report-identifier* (if report out);
- f) *subject-identifier* (if report out); and
- g) event date/time

3.1.2.3.3.2 Address look-up tables

The Message Transfer and Control Unit shall include look-up tables used for address conversion, covering two aspects:

- a) a MD look-up table as specified in 3.1.2.3.3.2.1, for the algorithmic conversion of an AF-Address to an XF-Address; and
- b) a user address look-up table of individual users as specified in 3.1.2.3.3.2.2, for the conversion of an AF-Address to and from an MF-Address of any AMHS Addressing Scheme.

Note.— The way in which these tables are populated and maintained up-to-date is an organisational matter.

3.1.2.3.3.2.1 MD look-up Tables

3.1.2.3.3.2.1.1 The MD (Management Domain) look-up table maintained by in the Message Transfer and Control Unit shall include a list of entries identifying an organizational entity, which either is an AMHS Management Domain, or collectively uses the services of a given AMHS Management Domain, each entry comprising:

- a) a string of characters identifying one of the following:
 - 1) a country (two-letter designator as specified in ICAO Document 7910);
 - 2) a location (four-letter designator as specified in ICAO Document 7910);
 - 3) an organization within a country (combination of a two-letter designator as specified in ICAO Document 7910 with a three-letter designator as specified in ICAO Document 8585); or
 - 4) an organization at a location (combination of a four-letter designator as specified in ICAO Document 7910 with a three-letter designator as specified in ICAO Document 8585); and
- b) the set of attributes identifying either the AMHS Management Domain implemented by the organizational entity defined in a), if existing, or the AMHS Management Domain whose AFTN/AMHS Gateway may be used to communicate with indirect AMHS users within the aforementioned organisational entity, this set of attributes being composed of:
 - 1) country-name;
 - 2) ADMD-name; and
 - 3) PRMD-name (if any).

3.1.2.3.3.2.1.2 It shall be possible to derive unambiguously a single item b) from item a) by a search operation in the MD look-up table.

3.1.2.3.3.2.2 User address look-up Tables

3.1.2.3.3.2.2.1 The user address look-up table maintained by the Message Transfer and Control Unit shall include a list of entries, each of them comprising:

- a) the AF-Address of either an indirect AMHS user who also has a MF-Address, or of a direct AMHS user who has an AF-Address for communication with indirect AMHS users; and
- b) the MF-Address of that AMHS user, either direct or indirect, including all its address attributes.

3.1.2.3.3.2.2.2 It shall be possible to derive unambiguously item b) from item a), and vice-versa, by a searching operation in the user address look-up table.

3.1.2.3.3.2.2.3 In order not to restrict the potential form of an MF-Address, a user address look-up table shall support in the attributes included under item b) all the general attribute types authorized in ISO/IEC 10021-2, section 18.5, Table 10.

3.1.2.3.4 AFTN to AMHS Conversion

Note.— This clause specifies the actions to be performed by an AFTN/AMHS Gateway upon reception of messages from the AFTN for conveyance in the AMHS, after the accomplishment of the AFTN-related procedures by the AFTN Component as specified in 3.1.2.3.2.1.

3.1.2.3.4.1 Control function

3.1.2.3.4.1.1 Upon reception by the Message Transfer and Control Unit of a message passed from the AFTN Component, as the result of the provisions of 3.1.2.3.2.1.4 items a) and b), and of 3.1.2.3.2.1.6, the received message shall be processed in one of three mutually exclusive manners depending on the message category:

- a) processing as specified in 3.1.2.3.4.3, if the received message is an AFTN acknowledgement message as specified in Annex 10, Volume II, 4.4.15.6;
- b) processing as specified in 3.1.2.3.4.4, if the received message is an AFTN service message requesting correction by the originator of a message received with an unknown addressee indicator as specified in Annex 10, Volume II, 4.4.11.13.3; or
- c) processing as specified in 3.1.2.3.4.2, if the received message is other than those referred to in a) and b) above.

3.1.2.3.4.1.2 Upon completion of the processing specified in 3.1.2.3.4.1.1, the following transfers shall take place:

- a) transfer of the resulting AMHS information objects, if any, to the ATN Component for conveyance in the AMHS; and
- b) transfer of the resulting AFTN service messages, if any, to the AFTN Component for conveyance over the AFTN.

3.1.2.3.4.1.3 If, for any reason, the processing specified in clauses 3.1.2.3.4.1.1 and 3.1.2.3.4.1.2 cannot be properly achieved, the procedure shall unsuccessfully terminate, resulting in:

- a) logging of the error situation and reporting to a control position; and
- b) storage of the AFTN message for appropriate action at the control position.

3.1.2.3.4.2 Conversion of AFTN Messages

Upon reception by the Message Transfer and Control Unit of an AFTN message passed from the AFTN Component to be conveyed over the AMHS, this AFTN message shall be converted into an IPM conveyed with a Message Transfer Envelope to be transferred and delivered in the AMHS in compliance with the following:

- a) the specification of how the components of the AFTN Message are used for mapping onto the AMHS message parameters, as included in 3.1.2.3.4.2.1;
- b) the specification of how the IPM is generated, as included in 3.1.2.3.4.2.2; and
- c) the specification of how the Message Transfer Envelope elements are generated, as included in 3.1.2.3.4.2.3.

3.1.2.3.4.2.1 Use of AFTN Message components

3.1.2.3.4.2.1.1 Each component of an AFTN Message shall be processed as specified in the column “action” of Table 3.1.2-3.

3.1.2.3.4.2.1.2 These components which are classified as “T” or “T1” in the column “action” of Table 3.1.2-3 shall be translated into the AMHS parameter specified in the column “AMHS parameter” of Table 3.1.2-3 and according to the specification in the clause referred to in the column “mapping”.

Table 3.1.2-3. Use of AFTN Message Components

AFTN Message Part	Component	Action	AMHS parameter	Mapping
Heading	Start-of-Heading Character	-	-	-
	Transmission Identification	D	-	-
Address	Alignment Function	-	-	-
	Priority Indicator	T	ATS-Message-Priority (see Table 3.1.2-5/Part 5/1.2) priority (see Table 3.1.2-6/Part 1/1.1.6)	see 3.1.2.3.4.2.1.3
	Addressee Indicator(s)	T	primary-recipients (see Table 3.1.2-5/Part 2/4) recipient-name (see Table 3.1.2-6/Part 1/1.2.1)	see 3.1.2.3.4.2.1.4.2
	Alignment Function	-	-	-
Origin	Filing Time	T	ATS-Message-Filing-Time (see Table 3.1.2-5/ Part 5/1.3)	see 3.1.2.3.4.2.1.5
	Originator Indicator	T	originator (see Table 3.1.2-5/Part 2/2) this-IPM (see Table 3.1.2-5/Part 2/1) originator-name (see Table 3.1.2-6/Part 1/1.1.2)	see 3.1.2.3.4.2.1.4.1
	Priority Alarm	D	-	-
	Optional Heading Information	T1	ATS-Message-Optional-Heading-Info (see Table 3.1.2-5/Part 5/1.4)	see 3.1.2.3.4.2.1.6
	Alignment Function	-	-	-
	Start-of-Text Character	-	-	-

AFTN Message Part	Component	Action	AMHS parameter	Mapping
Text		T	ATS-Message-Text (see Table 3.1.2-5/Part 5/2)	see 3.1.2.3.4.2.1.7
Ending	Alignment Function	-	-	-
	Page-feed sequence	-	-	-
	End-of-Text Character	-	-	-

Legend (see 3.1.1):

T1 = conditionally translated

D = discarded

T = translated

- = not applicable

3.1.2.3.4.2.1.3 The value of the priority indicator of an AFTN message shall be:

- a) mapped into the abstract-value of the *priority* element of the message transfer envelope of the converted AMHS message as specified in the second column of Table 3.1.2-4; and
- b) conveyed as the value of the priority-indicator in the ATS-Message-Priority element of the IPM text of the converted AMHS message as specified in the third column of Table 3.1.2-4.

Note.— The transport priority used for the conveyance of AMHS messages is specified in 3.1.2.2.2.1.2.3.

Table 3.1.2-4. Mapping of AFTN Priority Indicator

AFTN Priority Indicator	AMHS Message Transfer Envelope priority	AMHS ATS-Message-Priority priority-indicator
SS	urgent	SS
DD	normal	DD
FF	normal	FF
GG	non-urgent	GG
KK	non-urgent	KK

3.1.2.3.4.2.1.4 The value of an AFTN address included in an AFTN message shall be converted into an MF-Address as respectively specified in 3.1.2.3.4.2.1.4.1 and 3.1.2.3.4.2.1.4.2 depending whether it is an originator indicator or an addressee indicator.

3.1.2.3.4.2.1.4.1 The following actions shall be performed in order to translate the originator indicator of an AFTN Message into the MF-Address included in the *originator-name* of the converted AMHS message:

- a) translation into the single MF-Address matching exactly the AF-Address of the originator, if such an MF-Address can be determined from the User address look-up table maintained in the Message Transfer and Control Unit; or
- b) if a) cannot be achieved, translation into the XF-address constructed using the single Management Domain identified by the set of *country-name*, *administration-domain-name* and (if any) *private-domain-name* attributes, determined among the entries in the MD look-up table, if any, matching exactly the following character substrings of the AFTN address and selected among these entries, if several are found, on the basis of a decreasing order of precedence from 1) to 4):
 - 1) characters 1 to 7,
 - 2) characters 1, 2, 5, 6 and 7,
 - 3) characters 1, 2, 3 and 4,
 - 4) characters 1 and 2; or
- c) if no adequate entry can be found in the MD look-up table, or if the procedure defined in b) does not result in a single resulting MD, unsuccessful termination of the procedure resulting in:
 - 1) logging of the error situation and reporting to a control position, and
 - 2) storage of the AFTN message for appropriate action at the control position.

Note.— The specification above does not constrain the search algorithm provided that the expected result is achieved.

3.1.2.3.4.2.1.4.2 Each addressee indicator of an AFTN Message shall be translated into the MF-Address included in a *recipient-name* of the converted AMHS message in the same way as an originator indicator, with the exception that the unsuccessful termination for one or several addressee indicators additionally results in the generation, in compliance with the provisions of Annex 10, Volume II, 4.4.11.13.3, of an AFTN service message requesting correction by the originator of a message received with an unknown addressee indicator, the unknown addressee indicator(s) included in item 8) of the text message taking the value of these addressee indicators for which the translation process failed.

Note.— A PDAI included in the addressee indicator(s) of an AFTN Message is translated into an MF-Address in the same way as any addressee indicator.

3.1.2.3.4.2.1.5 The value of the Filing Time of an AFTN message shall be conveyed as the value of the filing-time element in the ATS-Message-Filing-Time element of the IPM text of the converted AMHS message.

3.1.2.3.4.2.1.6 The ATS-Message-Optional-Heading-Info element of the IPM text in the converted AMHS message shall either:

- a) convey the value of the Optional Heading Information of the AFTN message as the value of its optional-heading-information element, if the Optional Heading Information element is present in the AFTN message; or
- b) be omitted in the converted AMHS message, if the Optional Heading Information element is not present in the AFTN message.

3.1.2.3.4.2.1.7 The content of the Text of an AFTN message, shall be conveyed in its entirety as the value of the ATS-Message-Text element in the IPM text of the converted AMHS message.

3.1.2.3.4.2.2 Generation of IPM

3.1.2.3.4.2.2.1 Each of the elements composing the IPM resulting from the conversion of an AFTN message in the Message Transfer and Control Unit shall be processed as specified in the column “action” of Table 3.1.2-5.

3.1.2.3.4.2.2.2 These elements which are classified as “G” or “T” in the column “action” of Table 3.1.2-5 shall be either generated or translated according to the specification in the clause referred to in the column “mapping” of Table 3.1.2-5.

Note.— Table 3.1.2-5 is structured as a PRL derived from the profile specification included in 2.2 and consequently from the ISPICS Proforma included in ISO/IEC ISP 12062-2 (AMH21) as well as from Table 3.1.2-2 in 3.1.2.2.3.2. The columns “Base” and “ISP” under “Origination” are extracted from ISO/IEC ISP 12062-2 and the column “Basic ATS Message Service” specifies the static capability of an IPM AU supporting the Basic ATS Message Service, i.e. the ability to generate the element as part of an IPM carrying an ATS Message. The references to the ISP Profile are indicated in the part titles as AMH21/ref where appropriate. The references in column Ref are those of the ISP.

Table 3.1.2-5. IPM Generation

Ref	Element	Origination			Action	Mapping / Notes
		Base	ISP	Basic ATS Mess. Service		
PART 1 : AMH21/A.1.1 SUPPORTED INFORMATION OBJECTS						
1	Interpersonal message (IPM)	M	M	M	T	see Part 1/1.1 and 1.2
1.1	heading	M	M	M	T	see Part 2
1.2	body	M	M	M	T	see Part 3

Ref	Element	Origination			Action	Mapping / Notes
		Base	ISP	Basic ATS Mess. Service		
2	Interpersonal Notification (IPN)	M	M	M	-	out of the scope of this clause
PART 2 : AMH21/A.1.2 IPM HEADING FIELDS						
1	this-IPM	M	M	M	T	see Part 4/3
2	originator	M	M	M	T	see 3.1.2.3.4.2.2.3 and Part 4/2
3	authorizing-users	O	O	O	X	-
4	primary-recipients	M	M	M	T	see 3.1.2.3.4.2.2.4 and Part 4/1
5	copy-recipients	M	M	M	X	-
6	blind-copy-recipients	O	O	O	X	-
7	replied-to-IPM	M	M	M	X	-
8	obsoleted-IPMs	O	O	O	X	-
9	related-IPMs	O	O	O	X	-
10	subject	M	M	M	X	-
11	expiry-time	O	O	O	X	-
12	reply-time	O	O	O	X	-
13	reply-recipients	O	O	O	X	-
14	importance	O	O	O	X	-
15	sensitivity	O	O	O	X	-
16	auto-forwarded	O	O	O	X	-
17	extensions	O	O	O	X	-
17.1	incomplete-copy	O	O	O	X	-
17.2	languages	O	O	O	X	-
17.3	auto-submitted	O	I	I	X	-

Ref	Element	Origination			Action	Mapping / Notes
		Base	ISP	Basic ATS Mess. Service		
PART 3 : AMH21/A.1.3 IPM BODY						
1	ia5-text	O	O	M	T	see Part 3/1.1 and 1.2
1.1	parameters	M	M	M	G	see Part 3/1.1.1
1.1.1	repertoire	O	O	O	G	see 3.1.2.3.4.2.2.5
1.2	data	M	M	M	T	see Part 5
2	voice	I	I	I	X	-
3	g3-facsimile	O	O	O	X	-
4	g4-class-1	O	O	O	X	-
5	teletex	O	O	O	X	-
6	videotex	O	O	O	X	-
7	encrypted	I	I	I	X	-
8	message	O	O	O	X	-
9	mixed-mode	O	O	O	X	-
10	bilaterally-defined	O	O	O	X	-
11	nationally-defined	O	O	O	X	-
12	externally-defined	O	M	M	X	-
PART 4 : AMH21/A.1.5 COMMON DATA TYPES						
1	RecipientSpecifier					
1.1	recipient	M	M	M	T	see 3.1.2.3.4.2.2.6 and Part 4/2
1.2	notification-requests	O	O	M	T	see Part 4/1.2.1-1.2.3
1.2.1	rn	O	O	M	T	see 3.1.2.3.4.2.2.7
1.2.2	nrn	O	O	M	T	see 3.1.2.3.4.2.2.7
1.2.3	ipm-return	O	O	O	X	-
1.3	reply-requested	O	O	O	X	-
1.4	recipient-extensions	O	I	I	X	-
2	ORDescriptor					
2.1	formal-name	M	M1	M	T	see 3.1.2.3.4.2.2.8
2.2	free-form-name	O	O	O	X	-

Ref	Element	Origination			Action	Mapping / Notes
		Base	ISP	Basic ATS Mess. Service		
2.3	telephone-number	O	O	O	X	-
3	IPMIdentifier					
3.1	user	M	M	M	T	see 3.1.2.3.4.2.2.9
3.2	user-relative-identifier	M	M	M	G	-
PART 5 : IPM SUPPORT OF THE BASIC ATS MESSAGE SERVICE						
1	ATS-Message-Header	-	-	M	T	see Part 5/1.1-1.6
1.1	start-of-heading	-	-	M	G	see 3.1.2.2.3.2
1.2	ATS-Message-Priority	-	-	M	T	see Part 5/1.2.1-1.2.3
1.2.1	priority-prompt	-	-	M	G	see 3.1.2.2.3.2
1.2.2	priority-indicator	-	-	M	T	see 3.1.2.3.4.2.1.3
1.2.3	priority-separator	-	-	M	G	see 3.1.2.2.3.2
1.3	ATS-Message-Filing-Time	-	-	M	T	see Part 5/1.3.1-1.3.3
1.3.1	filing-time-prompt	-	-	M	G	see 3.1.2.2.3.2
1.3.2	filing-time	-	-	M	T	see 3.1.2.3.4.2.1.5
1.3.3	filing-time-separator	-	-	M	G	see 3.1.2.2.3.2
1.4	ATS-Message-Optional-Heading-Info	-	-	O	T1	see Part 5/1.4.1-1.4.3
1.4.1	OHI-prompt	-	-	M	G	see 3.1.2.2.3.2
1.4.2	optional-heading-information	-	-	M	T	see 3.1.2.3.4.2.1.6
1.4.3	OHI-separator	-	-	M	G	see 3.1.2.2.3.2

Ref	Element	Origination			Action	Mapping / Notes
		Base	ISP	Basic ATS Mess. Service		
1.5	end-of-heading-blank-line	-	-	M	G	see 3.1.2.2.3.2
1.6	start-of-text	-	-	M	G	see 3.1.2.2.3.2
2	ATS-Message-Text	-	-	M	T	see 3.1.2.3.4.2.1.7

Legend (see 3.1.1) :

M = mandatory support

M1 = minimal O/R name mandatory support

O = optional support

I = out of scope

- = not applicable

G = generated

T = translated

T1 = conditionally translated

X = excluded (not used)

3.1.2.3.4.2.2.3 The *originator* heading field shall:

- a) identify the indirect AMHS user who originated the AFTN message; and
- b) be structured as specified in Table 3.1.2-5/ Part 4/2.

3.1.2.3.4.2.2.4 The *primary-recipients* heading field shall:

- a) include the identification of the recipient(s) of the AFTN message; and
- b) be structured as specified in Table 3.1.2-5/ Part 4/1.

3.1.2.3.4.2.2.5 The element *repertoire* shall take the abstract value “ia5”.

3.1.2.3.4.2.2.6 The element(s) *recipient* in the *primary-recipients* heading field shall:

- a) identify the recipient(s) of the AFTN message; and
- b) be structured as specified in Table 3.1.2-5/ Part 4/2.

3.1.2.3.4.2.2.7 The values “rn” and “nrn” shall be taken simultaneously by the element *notification-requests* if, and only if the element *priority-indicator* included in the message, as specified Table 3.1.2-5 / Part 5/1.2.2, has the value “SS”.

3.1.2.3.4.2.2.8 The element *formal-name* shall:

- a) take the form of an MF-Address; and

- b) be converted as specified in 3.1.2.3.4.2.1.4.

3.1.2.3.4.2.2.9 The element *user* in the *this-IPM* heading field shall:

- a) be the MF-Address of the indirect AMHS user who originated the AFTN message;
and
- b) be converted as specified in 3.1.2.3.4.2.1.4.1.

3.1.2.3.4.2.3 Generation of Message Transfer Envelope

3.1.2.3.4.2.3.1 Each of the elements composing the Message Transfer Envelope conveyed with an IPM resulting from the conversion of an AFTN message shall be processed as specified in the column “action” of Table 3.1.2-6.

3.1.2.3.4.2.3.2 These elements which are classified as “G”, “G1” and “T” in the column “action” of Table 3.1.2-6 shall be handled according to the specification in the clause referred to in the column “mapping” of Table 3.1.2-6.

Note 1.— Table 3.1.2-6 is structured as a PRL derived from the ISPICS Proforma included in ISO/IEC ISP 10611-3. The columns “Base” and “ISP” are extracted from ISO/IEC ISP 10611-3, and the column “Basic ATS Message Service” specifies the static capability of an AU, for the MT-Elements of Service, i.e. the ability to convey, handle and act in relation with the element. The references to the ISP Profile are indicated in the part titles as AMH11/ref where appropriate.

Table 3.1.2-6. MessageTransfer for conveyance of an IPM

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Mapping / Notes
PART 1 : AMH11/A.1.4.2 MESSAGETRANSFER						
1	MessageTransferEnvelope	M	M	M	T	see Part 1/1.1 and 1.2
1.1	(per message fields)					
1.1.1	message-identifier	M	M	M	G	see Part 2/1
1.1.2	originator-name	M	M	M	T	see 3.1.2.3.4.2.3.3
1.1.3	original-encoded-information-types	M	M-	M-	G	see 3.1.2.3.4.2.3.4 and Part 2/3
1.1.4	content-type	M	M-	M-	G	see 3.1.2.3.4.2.3.5 and Part 2/8

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Mapping / Notes
1.1.5	content-identifier	M	M	M	G1	see 3.1.2.3.4.2.3.6
1.1.6	priority	M	M	M	T	see 3.1.2.3.4.2.1.3
1.1.7	per-message-indicators	M	M	M	G	see Part 2/4
1.1.8	deferred-delivery-time	O	M-	M-	X	-
1.1.9	per-domain-bilateral-information	O	M-	M-	G1	see 3.1.2.3.4.2.3.7 and Part 2/5
1.1.10	trace-information	M	M	M	G	see Part 2/6
1.1.11	extensions	M	M	M	G/X	see 3.1.2.3.4.2.3.8 and Part 3/1
1.1.11.1	recipient-reassignment-prohibited	O	M	M	X	-
1.1.11.2	dl-expansion-prohibited	O	M	M	X	-
1.1.11.3	conversion-with-loss-prohibited	O	M	M	X	-
1.1.11.4	latest-delivery-time	O	M-	M-	X	-
1.1.11.5	originator-return-address	O	M-	M-	X	-
1.1.11.6	originator-certificate	O	M-	M-	X	-
1.1.11.7	content-confidentiality-algorithm-identifier	O	M-	M-	X	-
1.1.11.8	message-origin-authentication-check	O	M-	M-	X	-
1.1.11.9	message-security-label	O	M-	M-	X	-
1.1.11.10	content-correlator	M	M	M	G1	see 3.1.2.3.4.2.3.6
1.1.11.11	dl-expansion-history	M	M-	M-	X	see Note 2

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Mapping / Notes
1.1.11.12	internal-trace-information	M	M	M	G	see Part 3/5
1.2	per-recipient-fields	M	M	M	T	see Part 1/1.2.1-1.2.5
1.2.1	recipient-name	M	M	M	T	see 3.1.2.3.4.2.3.9
1.2.2	originally-specified-recipient-number	M	M	M	G	see 3.1.2.3.4.2.3.10
1.2.3	per-recipient-indicators	M	M	M	G	see 3.1.2.3.4.2.3.11
1.2.4	explicit-conversion	O	M-	M-	X	-
1.2.5	extensions	M	M	M	X	-
2	content	M	M	M	T	see 3.1.2.3.4.2.2
PART 2 : AMH11/A.1.5 COMMON DATA TYPES						
1	MTSIdentifier					
1.1	global-domain-identifier	M	M	M	G	see 3.1.2.3.4.2.3.12 and Part 2/2
1.2	local-identifier	M	M	M	G	see 3.1.2.3.4.2.3.13
2	GlobalDomainIdentifier					
2.1	country-name	M	M	M	G	see 3.1.2.3.4.2.3.14
2.2	administration-domain-name	M	M	M	G	see 3.1.2.3.4.2.3.15
2.3	private-domain-identifier	M	M	M	G	see 3.1.2.3.4.2.3.16
3	EncodedInformationTypes					
3.1	built-in-encoded-information-types	M	M	M	G	see 3.1.2.3.4.2.3.4

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Mapping / Notes
3.2	(non-basic parameters)	O	M-	M-	X	-
3.3	extended-encoded-information-types	M	M	M	X	-
4	PerMessageIndicators					
4.1	disclosure-of-other-recipients	M	M	M	G	see 3.1.2.3.4.2.3.17
4.2	implicit-conversion-prohibited	M	M	M	G	see 3.1.2.3.4.2.3.18
4.3	alternate-recipient-allowed	M	M	M	G	see 3.1.2.3.4.2.3.19
4.4	content-return-request	O	M-	M-	X	see 3.1.2.3.4.2.3.20
4.5	reserved	O	M-	M-	X	-
4.6	bit-5	O	M-	M-	X	-
4.7	bit-6	O	M-	M-	X	-
4.8	service-message	O	M-	M-	X	-
5	PerDomainBilateralInformation					
5.1	country-name	M	M-	M-	G1	see 3.1.2.3.4.2.3.21
5.2	administration-domain-name	M	M-	M-	G1	see 3.1.2.3.4.2.3.21
5.3	private-domain-identifier	O	M-	M-	G1	see 3.1.2.3.4.2.3.21
5.4	bilateral-information	M	M-	M-	G1	see 3.1.2.3.4.2.3.22
6	TraceInformation					

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Mapping / Notes
6.1	TraceInformationElement	M	M	M	G	see Part 2/6.1.1 and 6.1.2
6.1.1	global-domain-identifier	M	M	M	G	see 3.1.2.3.4.2.3.23 and Part 2/2
6.1.2	domain-supplied-information	M	M	M	G	see Part 2/6.1.2.1-6.1.2.4
6.1.2.1	arrival-time	M	M	M	G	see 3.1.2.3.4.2.3.24
6.1.2.2	routing-action	M	M	M	G	see Part 2/6.1.2.2.1 and 6.1.2.2.2
6.1.2.2.1	relayed	M	M	M	G	see 3.1.2.3.4.2.3.25
6.1.2.2.2	rerouted	O	C1	C1	X	see Note 3
6.1.2.3	attempted-domain	O	C1	C1	X	see Note 3
6.1.2.4	(additional actions)					
6.1.2.4.1	deferred-time	M	C2	C2	X	-
6.1.2.4.2	converted-encoded-information-types	O	M-	M-	X	-
6.1.2.4.3	other-actions	O	M-	M-	X	-
6.1.2.4.3.1	redirected	O	M-	M-	X	see Note 4
6.1.2.4.3.2	dl-operation	O	M-	M-	X	see Note 2
8	ContentType					
8.1	built-in	M	M-	M-	G	see 3.1.2.3.4.2.3.5
8.2	extended	O	M-	M-	X	-

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Mapping / Notes
PART 3 : AMH11/A.1.6 EXTENSION DATA TYPES						
1	ExtensionField					
1.1	type	M	M	M	G	see Part 3/1.1.1 and 1.1.2
1.1.1	standard-extension	M	M	M	G	see 3.1.2.3.4.2.3.8
1.1.2	private-extension	O	M-	M-	X	-
1.2	criticality	M	M	M	G	see 3.1.2.3.4.2.3.8
1.3	value	M	M	M	G	see 3.1.2.3.4.2.3.8
5	InternalTraceInformation					
5.1	global-domain-identifier	M	M	M	G	see 3.1.2.3.4.2.3.23
5.2	mta-name	M	M	M	G	see 3.1.2.3.4.2.3.26
5.3	mta-supplied-information	M	M	M	G	see Part 3/5.3.1-5.3.4
5.3.1	arrival-time	M	M	M	G	see 3.1.2.3.4.2.3.24
5.3.2	routing-action	M	M	M	G	see Part 3/5.3.2.1-5.3.2.2
5.3.2.1	relayed	M	M	M	G	see 3.1.2.3.4.2.3.25
5.3.2.2	rerouted	O	C1	C1	X	see Note 3
5.3.3	attempted	O	C1	C1	X	see Note 3
5.3.4	(additional actions)					
5.3.4.1	deferred-time	M	C2	C2	X	-
5.3.4.2	converted-encoded-information-types	O	M-	M-	X	-

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Mapping / Notes
5.3.4.3	other-actions	O	M-	M-	X	-
5.3.4.3.1	redirected	O	M-	M-	X	see Note 4
5.3.4.3.2	dl-operation	O	M-	M-	X	see Note 2

Legend (see 3.1.1):

- M = mandatory support
- M- = minimal mandatory support
- O = optional support
- I = out of scope
- = not applicable
- C1 = if rerouting is supported then M else M-
- C2 = if deferred delivery is supported then M else M-
- G = generated
- G1 = optionally generated
- T = translated
- X = excluded

Note 2.— The DL-expansion capability of an AFTN/AMHS Gateway is implemented in the ATN Component rather than in the Message Transfer and Control Unit.

Note 3.— The rerouting capability of an AFTN/AMHS Gateway, if any, is implemented in the ATN Component rather than in the Message Transfer and Control Unit.

Note 4.— The redirection capability of an AFTN/AMHS Gateway, if any, is implemented in the ATN Component rather than in the Message Transfer and Control Unit.

3.1.2.3.4.2.3.3 The value of the element *originator-name* shall:

- a) be the address of the indirect AMHS user who originated the AFTN message;
- b) take the form of an MF-Address; and
- c) be converted as specified in 3.1.2.3.4.2.1.4.1.

3.1.2.3.4.2.3.4 The element *original-encoded-information-types* shall:

- a) take the abstract-value “ia5-text”, which is a value of type BuiltInEncodedInformationTypes; and
- b) be formed as specified in Table 3.1.2-6/ Part 2/ 3.

3.1.2.3.4.2.3.5 The element *content-type* shall:

- a) take the abstract-value “interpersonal-messaging-1984”, which is a value of type BuiltInContentType; and
- b) be formed as specified in Table 3.1.2-6/ Part 2/ 8.

3.1.2.3.4.2.3.6 The generation of this element shall be optional, as a matter of policy local to the AMHS Management Domain operating the AFTN/AMHS Gateway.

3.1.2.3.4.2.3.7 The element *per-domain-bilateral-information* shall be:

- a) optionally generated, as a matter of policy local to the AMHS Management Domain operating the AFTN/AMHS Gateway; and
- b) if present, structured as specified in Table 3.1.2-6/ Part 2/ 5.

3.1.2.3.4.2.3.8 The only extensions used shall:

- a) belong to the type “standard-extension”;
- b) contain the following elements:
 - 1) *content-correlator*, if used; and
 - 2) *internal-trace-information*;
 - 3) *conversion-with-loss-prohibited* elements;
- e) take a criticality value as specified in ISO/IEC 10021-4, Figure 2; and
- f) take values as specified in 3.1.2.3.4.2.3.6 and Table 3.1.2-6/Part 3/5, respectively.

Note.— The non-use of the elements recipient-reassignment-prohibited, dl-expansion-prohibited and conversion-with-loss-prohibited implies, in compliance with ISO/IEC 10021-4, that they are assumed to take their default abstract-values, which are “recipient-reassignment allowed”, “DL-expansion-allowed” and “conversion-with-loss-allowed”, respectively.

3.1.2.3.4.2.3.9 The value of the element *recipient-name* in each of the *per-recipient-fields* elements shall:

- a) be the address of each addressee indicated in the AFTN message, respectively;
- b) take the form of a MF-Address; and
- c) be converted as specified in 3.1.2.3.4.2.1.4.2.

3.1.2.3.4.2.3.10 The value of the element *originally-specified-recipient-number* in each of the *per-recipient-fields* elements shall be generated by the Message Transfer and Control Unit as specified in ISO/IEC 10021-4, 12.2.1.1.1.5.

3.1.2.3.4.2.3.11 The components of the element *per-recipient-indicators* in each of the *per-recipient-fields* elements shall be generated taking the following abstract-values:

- a) “responsible” for the *responsibility* element;
- b) “non-delivery-report” for the *originating-MTA-report-request* element; and
- c) “non-delivery-report” for the *originator-report-request* element.

3.1.2.3.4.2.3.12 The element *global-domain-identifier* in the *MTS-identifier* shall:

- a) identify the AMHS Management Domain operating the AFTN/AMHS Gateway;
and
- b) be composed as specified in Table 3.1.2-6 / Part 2/2.

3.1.2.3.4.2.3.13 The element *local-identifier* in the *MTS-identifier* shall be generated locally so as to ensure that it distinguishes the message from all other messages, probes or reports generated in the AMHS Management Domain operating the AFTN/AMHS Gateway.

3.1.2.3.4.2.3.14 The element *country-name* in the *global-domain-identifier* element of the *MTS-identifier* and of the first *trace-information-element* shall:

- a) be part of the identification of the AMHS Management Domain operating the AFTN/AMHS Gateway by taking one of the following values:
 - 1) the two-character alphabetical country-indicator as specified in ISO 3166 for the country, or for one of the countries, where the AMHS Management Domain has been registered, if the AMHS Management Domain has been subject to national or multi-national registration; or
 - 2) a two-character alphabetical indicator dedicated to an international organization, if the AMHS Management Domain has been subject to international registration as specified in ITU-T Recommendation X.666;
and
- b) be encoded as a Printable String.

3.1.2.3.4.2.3.15 The element *administration-domain-name* in the *global-domain-identifier* element of the *MTS-identifier* and of the first *trace-information-element* shall:

- a) be part of the identification of the AMHS Management Domain operating the AFTN/AMHS Gateway by taking one of the following values, depending on its status:
 - 1) the name of the ADMD under which the AMHS Management Domain has been registered, either nationally or internationally, if the AMHS Management Domain operates as an ADMD;
 - 2) the name of the ADMD to which the AMHS Management Domain is connected, if the AMHS Management Domain operates as a PRMD; or

- 3) the value single-space if the AMHS Management Domain operates as a PRMD and is unique with regard to the country-name identifying the area where it is registered, either nationally or internationally; and
- b) be encoded as a Printable String.

3.1.2.3.4.2.3.16 The element *private-domain-identifier* in the *global-domain-identifier* element of the *MTS-identifier* and of the first *trace-information-element* shall be handled in one of the following manners, depending on the status under which the AMHS Management Domain operates:

- a) generation of the element, with the value of the name of the PRMD, encoded as a Printable String, if the AMHS Management Domain operates as an PRMD; or
- b) omission in the *global-domain-identifier* if the AMHS Management Domain operates as an ADMD.

3.1.2.3.4.2.3.17 The element *disclosure-of-other-recipients* shall take its default abstract-value, which is “disclosure-of-other-recipients-prohibited”.

3.1.2.3.4.2.3.18 The element *implicit-conversion-prohibited* shall take its default abstract-value, which is “implicit-conversion-allowed”.

3.1.2.3.4.2.3.19 The element *alternate-recipient-allowed* shall take the abstract-value “alternate-recipient-allowed”.

3.1.2.3.4.2.3.20 The element *content-return-request* shall take its default abstract-value, which is “content-return-not-requested”.

3.1.2.3.4.2.3.21 The elements *country-name*, *administration-domain-name* and *private-domain-identifier* shall together identify the AMHS Management Domain for which the bilateral-information is intended if, and only if, the element *bilateral-information* as specified in 3.1.2.3.4.2.3.22 is present.

3.1.2.3.4.2.3.22 The generation of this element shall be optional, as a matter of bilateral agreement between the AMHS Management Domain operating the AFTN/AMHS Gateway and an other AMHS Management Domain.

3.1.2.3.4.2.3.23 The element *global-domain-identifier* in the *trace-information* or in the *internal-trace-information* shall:

- a) identify the AMHS Management Domain operating the AFTN/AMHS Gateway; and
- b) be composed as specified in Table 3.1.2-6 / Part 2/2.

3.1.2.3.4.2.3.24 The element *arrival-time* in the first element of *trace-information* or of *internal-trace-information* shall take the semantic value of the time when the message was received by the Message Transfer and Control Unit for conveyance in the AMHS.

3.1.2.3.4.2.3.25 The element *routing-action* in the first element of *trace-information* or of *internal-trace-information* shall take the abstract-value “relayed”.

3.1.2.3.4.2.3.26 The element *mta-name* in the first element of *internal-trace-information* shall be the *mta-name* assigned to the Message Transfer and Control Unit included in the AFTN/AMHS Gateway.

Note.— The structure of the mta-name of the Message Transfer and Control Unit included in an AFTN/AMHS Gateway within an AMHS Management Domain is a matter of policy internal to the AMHS Management Domain.

3.1.2.3.4.3 Conversion of AFTN Acknowledgement Messages

3.1.2.3.4.3.1 Initial processing of AFTN Acknowledgement Message

3.1.2.3.4.3.1.1 Upon reception by the Message Transfer and Control Unit of an AFTN acknowledgement message, passed from the AFTN Component to be conveyed in the AMHS, the received message shall be processed in one of the following manners depending on whether or not the subject AFTN message previously passed through the Message Transfer and Control Unit:

- a) processing as specified in 3.1.2.3.4.3.1.2, if the subject AFTN message, as identified in the text of AFTN acknowledgement message, previously passed through the Message Transfer and Control Unit; or
- b) processing as follows, if the subject AFTN message did not previously pass through the Message Transfer and Control Unit:
 - 1) logging of the error situation and reporting to a control position; and
 - 2) conversion of the AFTN acknowledgement message into an IPM conveyed with a Message Transfer Envelope as specified in 3.1.2.3.4.3.1.5.

3.1.2.3.4.3.1.2 If the subject AFTN message previously passed through the Message Transfer and Control Unit, the AFTN acknowledgement message shall then be processed in one of the following manners depending on whether the subject IPM was received from the AMHS without or with *receipt-notification-request*:

- a) processing as follows, if the subject IPM was received from the AMHS without *receipt-notification-request*:
 - 1) conversion into an IPM conveyed with a Message Transfer Envelope as specified in 3.1.2.3.4.3.1.5; and
 - 2) logging of the error situation and reporting to a control position; or
- b) processing as specified in 3.1.2.3.4.3.1.3, if the subject IPM was received from the AMHS with *receipt-notification-request*.

3.1.2.3.4.3.1.3 If the subject IPM had been received from the AMHS with *receipt-notification-request*, the AFTN acknowledgement message shall be converted by the AFTN/AMHS Gateway into an Interpersonal Notification (IPN) taking the form of a Receipt Notification (RN), conveyed with a Message Transfer Envelope generated in compliance with the provisions of 3.1.2.3.4.3.1.4.

3.1.2.3.4.3.1.4 When the provisions of 3.1.2.3.4.3.1.3 apply, the generation of the RN and of the Message Transfer Envelope shall be performed in compliance with the following:

- a) the specification of how the components of the AFTN Service Message are used, as included in 3.1.2.3.4.3.2;
- b) the specification of how the RN is generated, as included in 3.1.2.3.4.3.3; and
- c) the provisions of 3.1.2.3.4.2.3 concerning the generation of the Message Transfer Envelope, with the exception of the differences specified in 3.1.2.3.4.3.4.

3.1.2.3.4.3.1.5 When an acknowledgement message is converted into an IPM as the result of 3.1.2.3.4.3.1.1 or 3.1.2.3.4.3.1.2, the specification of 3.1.2.3.4.2 shall apply with the exception of the *subject* element in the IPM heading fields, initially specified in Table 3.1.2-5/Part 2/10, which is then generated and takes the value “AFTN service information”.

3.1.2.3.4.3.2 Use of AFTN Service Message components

3.1.2.3.4.3.2.1 Each component of an AFTN acknowledgement message shall be processed for the generation of a RN as specified in the column “action” of Table 3.1.2-7.

3.1.2.3.4.3.2.2 These components which are classified as “T” or “T1” in the column “action” of Table 3.1.2-7 shall be translated into the AMHS parameter specified in the column “AMHS parameter” of Table 3.1.2-7 and according to the specification in the clause referred to in the column “mapping”.

Table 3.1.2-7. Use of AFTN Service Message Components

AFTN Message Part	Component	Action	AMHS parameter	Mapping
Heading	Start-of-Heading Character	-	-	-
	Transmission Identification	D	-	-
Address	Alignment Function	-	-	-
	Priority Indicator	T	priority (see Table 3.1.2-9/Part 1/1.1.6)	see 3.1.2.3.4.3.4.3
	Addressee Indicator	T	recipient-name (see Table 3.1.2-9/Part 1/1.2.1)	see 3.1.2.3.4.3.4.4
	Alignment Function	-	-	-
Origin	Filing Time	T	receipt-time (see Table 3.1.2-8/Part 2/7.1)	see 3.1.2.3.4.3.2.4
	Originator Indicator	T	ipn-originator (see Table 3.1.2-8/Part 2/2) originator-name (see Table 3.1.2-6/Part 1/1.1.2)	see 3.1.2.3.4.3.2.3 see 3.1.2.3.4.2.1.4.1
	Priority Alarm	D	-	-
	Optional Heading Information	D	-	-

AFTN Message Part	Component	Action	AMHS parameter	Mapping
	Alignment Function	-	-	-
	Start-of-Text Character	-	-	-
Text		D	-	-
Ending	Alignment Function	-	-	-
	Page-feed sequence	-	-	-
	End-of-Text Character	-	-	-

Legend: (see 3.1.1.)

D = discarded

T = translated

- = not applicable

3.1.2.3.4.3.2.3 Upon generation of a RN as the result of the receipt of an AFTN acknowledgement message by the Message Transfer and Control Unit, the originator indicator element of the AFTN acknowledgement message shall be translated into the *ipn-originator* element of the RN.

3.1.2.3.4.3.2.4 Upon generation of a RN as the result of the receipt of an AFTN acknowledgement message by the Message Transfer and Control Unit, the filing time of the AFTN acknowledgement message shall be converted into the *receipt-time* element, which is of ASN.1 (Abstract syntax notation one) type UTCTime, as the result of the following:

- a) generation by the Message Transfer and Control Unit of the YY figures identifying the current year (characters 1 and 2 of the string) in the *receipt-time* element;
- b) generation by the Message Transfer and Control Unit of the MM figures identifying the current month (characters 3 and 4 of the string) in the *receipt-time* element;
- c) mapping of the value of the first two figures of the date-time group into the value of the DD figures identifying the day (characters 5 and 6 of the string) in the *receipt-time* element;
- d) mapping of the value of the four last figures of the date-time group, which together represent the hours and minutes, into the value of the hhmm figures (characters 7 to 10 of the string) in the *receipt-time* element; and
- e) addition by the Message Transfer and Control Unit of an eleventh and last character in the string composing the *receipt-time* element taking the value "Z".

3.1.2.3.4.3.3 Generation of RN

3.1.2.3.4.3.3.1 Each of the elements composing the RN resulting from the receipt of an AFTN acknowledgement message in the Message Transfer and Control Unit shall be processed as specified in the column “action” of Table 3.1.2-8.

3.1.2.3.4.3.3.2 These elements are classified as “G” or “T” in the column “action” of Table 3.1.2-8 shall be either generated or translated according to the specification in the clause referred to in the column “mapping” of Table 3.1.2-8.

Note.— Table 3.1.2-8 is structured as a PRL derived from the profile specification included in 2.2 and consequently from the ISPICS Proforma included in ISO/IEC ISP 12062-2 (AMH21). The columns “Base” and “ISP” under “Origination” are extracted from ISO/IEC ISP 12062-2, and the column “Basic ATS Message Service” specifies the static capability of an IPM AU supporting the Basic ATS Message Service, i.e. the ability to generate the element as part of an IPN in the AMHS. The references to the ISP Profile are indicated in the part titles as AMH21/ref where appropriate. The references in column Ref are those of the ISP.

Table 3.1.2-8. RN Generation

Ref	Element	Origination			Action	Mapping / Notes
		Base	ISP	Basic ATS Mess. Service		
PART 1 : AMH21/A.1.1 SUPPORTED INFORMATION OBJECTS						
1	Interpersonal Message (IPM)	M	M	M	-	out of the scope of this clause
2	Interpersonal Notification (IPN)	M	M	M		see Part 2
PART 2 : AMH21/A.1.4 IPN FIELDS						
1	subject-ipm	M	M	M	G	see 3.1.2.3.4.3.3.3
2	ipn-originator	O	M	M	T	see 3.1.2.3.4.3.2.3 and Part 3/2
3	ipm-preferred-recipient	M	M	M	G2	see 3.1.2.3.4.3.3.4
4	conversion-eits	O	O	O	G2	see 3.1.2.3.4.3.3.5

Ref	Element	Origination			Action	Mapping / Notes
		Base	ISP	Basic ATS Mess. Service		
5	notification-extensions	O	I	I	X	-
6	non-receipt-fields	M	M	M	X	-
7	receipt-fields	O	O	O	T	see Part 2/7.1-7.4
7.1	receipt-time	M	M	M	T	see 3.1.2.3.4.3.2.4
7.2	acknowledgment-mode	O	O	O	G	see 3.1.2.3.4.3.3.6
7.3	suppl-receipt-info	O	O	O	X	-
7.4	rn-extensions	O	I	I	X	-
8	other-notification-type-fields	O	I	I	X	-
PART 3 : AMH21/A.1.5 COMMON DATA TYPES						
2	ORDescriptor					
2.1	formal-name	M	M1	M	T	see 3.1.2.3.4.3.3.7
2.2	free-form-name	O	O	O	X	
2.3	telephone-number	O	O	O	X	

Legend (see 3.1.1) :

- M = mandatory support
- M1 = minimal O/R name mandatory support
- O = optional support
- I = out of scope
- G = generated
- G2 = conditionally generated
- T = translated
- X = excluded (not used)

3.1.2.3.4.3.3.3 The element *subject-ipm* shall take the value of the *this-IPM* heading field of the subject IPM.

3.1.2.3.4.3.3.4 The element *ipm-preferred-recipient* shall:

- a) be present if, and only if:
 - 1) it would be different from the *ipn-originator* specified in 3.1.2.3.4.3.2.3; and
 - 2) it would not be the result of a DL-expansion;
- b) if present, identify the recipient of the subject IPM which caused the receipt of the AFTN acknowledgement message by the Message Transfer and Control Unit (as a result of the receipt by its addressee of the subject AFTN message); and
- c) if present, be the *O/R descriptor* of the recipient of the subject IPM.

3.1.2.3.4.3.3.5 The element *conversion-eits* shall:

- a) be present if, and only if, this encoded-information-types is different of the *originally-encoded-information-types* included in the subject IPM; and
- b) if present, take the value of the encoded-information-types of the subject IPM received by the Message Transfer and Control Unit.

3.1.2.3.4.3.3.6 The element *acknowledgement-mode* shall take the abstract-value “manual”, which is its default value.

3.1.2.3.4.3.3.7 The element *formal-name* in an *ORDescriptor* shall take the form of an O/R address and be converted from the originator indicator of the AFTN acknowledgement message as specified in 3.1.2.3.4.2.1.4.1.

3.1.2.3.4.3.4 Differences in the generation of Message Transfer Envelope

3.1.2.3.4.3.4.1 The elements composing the Message Transfer Envelope which is conveyed with a RN resulting from the receipt of an AFTN acknowledgement message by the Message Transfer and Control Unit, which are different from the specification of 3.1.2.3.4.2.3 shall be processed according to the specification in the clause referred to in the column “mapping” of Table 3.1.2-9.

3.1.2.3.4.3.4.2 An element subject to the provisions of 3.1.2.3.4.3.4.1 shall be processed as specified in the column “action” of Table 3.1.2-9, and in accordance with the specification referred to in the column “mapping” of Table 3.1.2-9.

Note.— Table 3.1.2-9 is structured as an extract of Table 3.1.2-6. The references used in the part titles and in the column “Ref” are those of Table 3.1.2-6.

**Table 3.1.2-9. MessageTransfer Envelope generation for conveyance with a RN
(Differences with Table 3.1.2-6)**

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Mapping / Notes
PART 1 : AMH11/A.1.4.2 MESSAGETRANSFER						
1	MessageTransferEnvelope	M	M	M	T	see Part 1/1.1 and 1.2
1.1	(per message fields)					
1.1.3	original-encoded-information-types	M	M-	M-	X	-
1.1.6	priority	M	M	M	G	see 3.1.2.3.4.3.4.3
1.1.7	per-message-indicators	M	M	M	G	see Part 2/4
1.2	per-recipient-fields	M	M	M	T	see Part 1/1.2.1 and 1.2.3
1.2.1	recipient-name	M	M	M	T	see 3.1.2.3.4.3.4.4
1.2.3	per-recipient-indicators	M	M	M	G	see 3.1.2.3.4.3.4.5
2	content	M	M	M	T	see 3.1.2.3.4.3.3
PART 2 : AMH11/A.1.5 COMMON DATA TYPES						
4	PerMessageIndicators					
4.2	implicit-conversion-prohibited	M	M	M	G	see 3.1.2.3.4.3.4.6

Legend (see 3.1.1) :

- M = mandatory support
- M- = minimal mandatory support
- G = generated
- T = translated
- X = excluded (not used)

3.1.2.3.4.3.4.3 The element *priority* shall take the abstract-value “urgent”.

3.1.2.3.4.3.4.4 The element *recipient-name* shall:

- a) identify the originator of the subject IPM; and
- b) take the form of an MF-Address.

3.1.2.3.4.3.4.5 The components of the element *per-recipient-indicators* shall be generated taking the following abstract-values:

- a) “responsible” for the *responsibility* element;
- b) “non-delivery-report” for the *originating-MTA-report-request* element; and
- c) “no-report” for the *originator-report-request* element.

3.1.2.3.4.3.4.6 The element *implicit-conversion-prohibited* shall take the abstract-value “implicit-conversion-prohibited”.

3.1.2.3.4.4 Conversion of AFTN Service Messages related to unknown addressee indicators

3.1.2.3.4.4.1 Initial Processing of the AFTN Service Message

3.1.2.3.4.4.1.1 Upon reception by the Message Transfer and Control Unit of an unknown address AFTN service message, passed from the AFTN Component to be conveyed in the AMHS, the received message shall be processed in one of the following manners:

- a) processing as specified in 3.1.2.3.4.4.1.2, if the subject AFTN message, as identified in the unknown address AFTN service message text, previously passed through the Message Transfer and Control Unit; or
- b) if the subject AFTN message did not previously pass through the Message Transfer and Control Unit, conversion of the unknown address AFTN service message into an IPM conveyed with a Message Transfer Envelope as specified in 3.1.2.3.4.4.1.7.

3.1.2.3.4.4.1.2 If the subject AMHS message previously passed through the Message Transfer and Control Unit, the received message shall be processed in either of the following manners depending on whether or not the unknown addressee indicator(s) which caused the generation of the unknown address AFTN service message can be determined:

- a) processing as specified in 3.1.2.3.4.4.1.3, if at least one valid addressee indicator which caused the generation of the unknown address AFTN service message can be found; or
- b) if no such valid addressee indicator can be found, conversion of the unknown address AFTN service message into an IPM conveyed with a Message Transfer Envelope as specified in 3.1.2.3.4.4.1.7.

3.1.2.3.4.4.1.3 For the addressee indicators determined as causing the generation of the unknown address AFTN service message, as the result of 3.1.2.3.4.4.1.2, the received message shall be processed as follows, depending on whether or not the conversion of each unknown addressee indicator into a recipient MF-Address in the same way as specified for an originator indicator in 3.1.2.3.4.2.1.4.1 can be successfully performed by the Message Transfer and Control Unit:

- a) processing as specified in 3.1.2.3.4.4.1.4, for the set of unknown addressee indicators which can be successfully translated into an MF-Address, if any; and

- b) for the set of unknown addressee indicators which cannot be successfully translated, if any, processing as follows:
 - 1) deletion in the text of the unknown address AFTN service message of all unknown addressee indicators processed as specified in a) above; and
 - 2) conversion of the resulting unknown address AFTN service message into an IPM conveyed with a Message Transfer Envelope as specified in 3.1.2.3.4.4.1.7.

3.1.2.3.4.4.1.4 For the unknown recipient MF-Addresses determined as the result of 3.1.2.3.4.4.1.3 a), the received message shall be processed as follows, depending on the abstract-values of the *originator-report-request* and of the *originating-MTA-report-request* elements in the *per-recipient-indicators* in the corresponding *per-recipient-fields* of the subject AMHS message:

- a) processing as specified in 3.1.2.3.4.4.1.5, for the set of recipients which meet the following condition, if any:
 - 1) the abstract-value of the *originator-report-request* differs from “report”; and
 - 2) the abstract-value of the *originating-MTA-report-request* differs from “report” and from “audited-report”; or
- b) processing as follows, for all other recipients, if any:
 - 1) replacement, in the text of the unknown address AFTN service message, of the entire list of unknown addressee indicators with a list restricted to the addressee indicators of these recipients; and
 - 2) conversion of the resulting unknown address AFTN service message into an IPM conveyed with a Message Transfer Envelope as specified in 3.1.2.3.4.4.1.7.

Note.— This clause aims at avoiding the generation of a non-delivery-report after the generation of a delivery-report by the MTCU for the same subject AMHS message.

3.1.2.3.4.4.1.5 For each unknown recipient MF-Address which has not been subject to the generation of a delivery-report, the received message shall be processed in one of the following manners:

- a) processing as specified in 3.1.2.3.4.4.1.6, if, for a given recipient, no non-delivery report has been generated yet in relation with the same subject AMHS message and with the same message recipient; or

- b) discarding of the unknown address AFTN service message for the considered unknown recipient MF-Address and termination of the procedure for the given recipient if a non-delivery report has already been generated in relation with the same subject AMHS message and with the same message recipient.

Note.— This clause aims at avoiding the generation of a multiple non-delivery-reports in relation with a single subject AMHS message which would have been split in several AFTN messages when converted from the AMHS to the AFTN, as the result of 3.1.2.3.5.2.1.7.

3.1.2.3.4.4.1.6 A non-delivery report related to the unknown recipient MF-Addresses which have not caused the conversion of the unknown address AFTN service message into an IPM as the result of 3.1.2.3.4.4.1.4 and 3.1.2.3.4.4.1.5, shall be generated in compliance with:

- a) the specification of 3.1.2.3.5.6 using the elements of the subject AMHS message; and
- b) the following specification of abstract-values:
 - 1) “unable-to-transfer” for the *non-delivery-reason-code*; and
 - 2) “unrecognised-OR-name” for the *non-delivery-diagnostic-code*; and
- c) the exception with respect to 3.1.2.3.5.6, that the *actual-recipient-name* element(s) in each *per-recipient-fields* element of the report take the value of the unknown recipient MF-Address(es) as determined in 3.1.2.3.4.4.1.5.

Note.— The potential future reception of an unknown address AFTN service message to be converted into a non-delivery-report requires the retention by the AFTN/AMHS Gateway of certain elements of the subject AMHS message for later report generation, if required.

3.1.2.3.4.4.1.7 When an unknown address AFTN service message is converted into an IPM as the result of 3.1.2.3.4.4.1.1 to 3.1.2.3.4.4.1.4, the specification of 3.1.2.3.4.2 shall apply, with the exception of the *subject* element in the IPM heading fields, initially specified in Table 3.1.2-5/Part2/10, which is then generated and takes the value “AFTN service information”.

3.1.2.3.5 AMHS to AFTN Conversion

Note.— This clause specifies the actions to be performed by an AFTN/AMHS Gateway upon reception of information objects from the AMHS for conveyance over the AFTN, after the accomplishment of the AMHS-related procedures by the ATN Component as specified in 3.1.2.3.2.2.

3.1.2.3.5.1 Control Function

3.1.2.3.5.1.1 Upon reception by the Message Transfer and Control Unit of an AMHS message passed by the ATN Component, the received message shall be processed in one of the following manners, depending on the abstract-value of the *content-type* element in the Message Transfer Envelope:

- a) processing as specified in 3.1.2.3.5.1.2 if the abstract-value of the element is either “interpersonal-messaging-1984”, or “interpersonal-messaging-1988”; or

- b) if the abstract-value of the element is neither “interpersonal-messaging-1984”, nor “interpersonal-messaging-1988”:
 - 1) rejection of the message for all the message recipients for which the *responsibility* element of the *per-recipient-indicators* had the abstract-value “responsible”; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values:
 - i) “unable-to-transfer” for the non-delivery-reason-code; and
 - ii) “content-type-not-supported” for the non-delivery-diagnostic-code.

Note 1.— The message recipients towards which the Message Transfer and Control Unit conveys the message are those identified by a recipient-name element in the per-recipient-fields element of the Message Transfer Envelope, and for which the responsibility element in the per-recipient-indicators element has the abstract-value “responsible”. In 3.1.2.3.5 the term “message recipient” refers to such a recipient.

Note 2.— Support of other content-types, e.g. edi-messaging, may be added in future packages.

3.1.2.3.5.1.2 Upon reception by the Message Transfer and Control Unit of an AMHS message whose *content-type* is either “interpersonal-messaging-1984” or “interpersonal-messaging-1988” passed from the ATN Component, the message shall be processed for conversion into an AFTN message in one of three mutually exclusive manners, depending on the nature of the content:

- a) processing for conversion into an AFTN message as specified in 3.1.2.3.5.2, if the content is an IPM;
- b) processing for conversion into an AFTN service message as specified in 3.1.2.3.5.3, if the content is an IPN which is a Receipt Notification (RN); or
- c) unsuccessful termination of the procedure, if the content is an IPN but not a RN, resulting in:
 - 1) logging of the error situation and reporting to a control position; and
 - 2) storage of the message for appropriate processing at the control position.

3.1.2.3.5.1.3 Upon reception by the Message Transfer and Control Unit of an AMHS non-delivery report passed from the ATN Component, the report shall be processed as specified in 3.1.2.3.5.4.

3.1.2.3.5.1.4 Upon reception by the Message Transfer and Control Unit of an AMHS probe passed by the ATN Component, the received probe shall be processed in one of the following manners, depending on the abstract-value of the *content-type* element in the Probe Transfer Envelope:

- a) processing for conveyance test as specified in 3.1.2.3.5.5 if the abstract-value of the element is either “interpersonal-messaging-1984”, or “interpersonal-messaging-1988”; or

- b) if the abstract-value of the element is neither “interpersonal-messaging-1984”, nor “interpersonal-messaging-1988”:
 - 1) rejection of the probe for all the probe recipients for which the *responsibility* element of the *per-recipient-indicators* had the abstract-value “responsible”; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values:
 - i) “unable-to-transfer” for the *non-delivery-reason-code*; and
 - ii) “content-type-not-supported” for the *non-delivery-diagnostic-code*.

3.1.2.3.5.1.5 Upon reception by the Message Transfer and Control Unit of an ISO/IEC 10021 information object other than those referred to in clauses 3.1.2.3.5.1.1 to 3.1.2.3.5.1.4 above, the processing by the Message Transfer and Control Unit shall unsuccessfully terminate, resulting in:

- a) logging of the error situation and reporting to a control position; and
- b) storage of the information object for appropriate processing at the control position.

Note.— The Message Transfer and Control Unit requests non-delivery-reports, but never delivery-reports when generating AMHS messages.

3.1.2.3.5.1.6 Upon completion by the Message Transfer and Control Unit of the processing specified in clauses 3.1.2.3.5.1.1 to 3.1.2.3.5.1.4 above, the resulting AFTN message(s) or AFTN service message(s), if any, shall be passed to the AFTN component, for conveyance over the AFTN.

3.1.2.3.5.1.7 If the generation of a report is required in relation with the result of the processing specified in clauses 3.1.2.3.5.1.1 or 3.1.2.3.5.1.4 above, either due to message rejection or probe test failure by the Message Transfer and Control Unit, or due to a delivery-report request in the subject AMHS message or probe, an appropriate AMHS report shall be generated as specified in 3.1.2.3.5.6.

3.1.2.3.5.2 AMHS IPM Conversion

Upon reception by the Message Transfer and Control Unit of an IPM conveyed with a Message Transfer Envelope passed from the ATN Component to be conveyed over the AFTN, this message shall be converted into an AFTN message in compliance with the following:

- a) the specification of the initial processing to be performed by the Message Transfer and Control Unit to determine the ability to convert the message and to split it into individually convertible messages, as included in 3.1.2.3.5.2.1;
- b) the specification of how the AFTN message is generated and how the AFTN message components are mapped from AMHS parameters, as included in 3.1.2.3.5.2.2;

- c) the specification of how the elements of the received IPM are handled, as included in 3.1.2.3.5.2.3; and
- d) the specification of how the Message Transfer Envelope elements are handled, as included in 3.1.2.3.5.2.4.

3.1.2.3.5.2.1 Initial processing of AMHS Messages

3.1.2.3.5.2.1.1 Upon reception by the Message Transfer and Control Unit of an IPM, the received message shall be processed in one of the following manners, depending on the abstract-value of the current encoded-information-types, determined as either the abstract-value of the latest *converted-encoded-information-types*, if existing, in the *trace-information* element, or as the abstract-value of the *original-encoded-information-types* element if the previous does not exist:

- a) processing as specified in 3.1.2.3.5.2.1.2 if the abstract-value of the current encoded-information-types is any of the following:
 - 1) basic “ia5-text”;
 - 2) externally-defined “ia5-text”;
 - 3) OID {id-cs-eit-authority 1};
 - 4) OID {id-cs-eit-authority 2};
 - 5) OID {id-cs-eit-authority 6}; or
 - 6) OID {id-cs-eit-authority 100}; or
- b) if the abstract-value differs from all values indicated in item a) above:
 - 1) rejection of the message for all the message recipients; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in all the *per-recipient-fields* of the report:
 - i) “unable-to-transfer” for the *non-delivery-reason-code*; and
 - ii) “encoded-information-types-unsupported” for the *non-delivery-diagnostic-code*.

3.1.2.3.5.2.1.2 A message which was not rejected as the result of 3.1.2.3.5.2.1.1 shall be processed in one of the following manners:

- a) processing as specified in 3.1.2.3.5.2.1.3 if the abstract-value of the *implicit-conversion-prohibited* in the *per-message-indicators* element in the Message Transfer Envelope differs from “prohibited”; or

- b) if the abstract-value of the element is “prohibited” and if the abstract-value of the encoded-information-types includes OID {id-cs-eit-authority 100}:
 - 1) rejection of the message for all the message recipients; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in all the *per-recipient-fields* of the report:
 - i) “conversion-not-performed” for the *non-delivery-reason-code*;
 - ii) “implicit-conversion-prohibited” for the *non-delivery-diagnostic-code*; and
 - iii) “unable to convert to AFTN” for the *supplementary-information*.

3.1.2.3.5.2.1.3 A message which was not rejected as the result of 3.1.2.3.5.2.1.2 shall be processed in one of the following manners:

- a) processing as specified in 3.1.2.3.5.2.1.4 if there is one single body part in the IPM body; or
- b) if there are multiple body parts in the IPM body:
 - 1) rejection of the message for all the message recipients; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in all the *per-recipient-fields* of the report:
 - i) “unable-to-transfer” for the *non-delivery-reason-code*;
 - ii) “content-syntax-error” for the *non-delivery-diagnostic-code*; and
 - iii) “unable to convert to AFTN due to multiple body parts” for the *supplementary-information*.

3.1.2.3.5.2.1.4 A message which was not rejected as the result of 3.1.2.3.5.2.1.3 shall be processed in one of the following manners:

- a) processing as specified in 3.1.2.3.5.2.1.5 if the body part type is one of the following:
 - 1) a basic body part type “ia5-text”;
 - 2) a standard extended body part type “ia5-text-body-part”;
 - 3) a standard extended body part type “general-text-body-part” of which the repertoire set description is Basic (ISO 646); or

- 4) a standard extended body part type “general-text-body-part” of which the repertoire set description is Basic-1 (ISO 8859-1), if and only if the local policy of the AMHS Management Domain is to support the conversion of this repertoire set into IA5IRV characters according to locally defined conversion rules; or
- b) if the body part type is different from the body part types 1) to 4) under a) above, or if the local policy of the AMHS Management Domain is not to support the conversion of the ISO 8859-1 repertoire set:
 - 1) rejection of the message for all the message recipients; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in all the *per-recipient-fields* of the report:
 - i) “unable-to-transfer” for the *non-delivery-reason-code*;
 - ii) “content-syntax-error” for the *non-delivery-diagnostic-code*; and
 - iii) “unable to convert to AFTN due to unsupported body part type” for the *supplementary-information*.

Note.— The locally defined conversion rules mentioned in bullet 4), item a) may be for example CCITT Recommendation X.408.

3.1.2.3.5.2.1.5 A message not rejected as the result of 3.1.2.3.5.2.1.4 shall then be processed in one of the following manners:

- a) processing as specified in 3.1.2.3.5.2.1.6 if the text structure in the body part in the body part complies with the requirements of 3.1.2.2.3.2; or
- b) if the text structure does not comply with the requirements of 3.1.2.2.3.2:
 - 1) rejection of the message for all the message recipients; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in all the *per-recipient-fields* of the report:
 - i) “unable-to-transfer” for the *non-delivery-reason-code*;
 - ii) “content-syntax-error” for the *non-delivery-diagnostic-code*; and
 - iii) “unable to convert to AFTN due to ATS-Message-Header syntax error” for the *supplementary-information*.

Note.— The compliance requested to meet the condition of item b) includes the requirement that the element is present and has a value which is syntactically valid for the priority indicator, i.e. a value among

SS, DD, FF, GG and KK, and for the filing time, i.e. a value in which the first six figures in the sequence build a valid date-time group.

3.1.2.3.5.2.1.6 A message which was not rejected as the result of 3.1.2.3.5.2.1.5 shall be processed in one of five mutually exclusive manners:

- a) processing as specified in 3.1.2.3.5.2.1.7 if the abstract-value of the *conversion-with-loss-prohibited* element in the *extensions* of the per message fields is “allowed”;
- b) if the abstract-value of the element *conversion-with-loss-prohibited* is “prohibited” and at least one line in the message exceeds 69 characters:
 - 1) rejection of the message for all the message recipients; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in all the *per-recipient-fields* of the report:
 - i) “conversion-not-performed” for the *non-delivery-reason-code*; and
 - ii) “line-too-long” for the *non-delivery-diagnostic-code*;
- c) if the abstract-value of the element *conversion-with-loss-prohibited* is “prohibited” and at least one punctuation symbol in the text is not authorized in Annex 10, Volume II, 4.1.2:
 - 1) rejection of the message for all the message recipients; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in all the *per-recipient-fields* of the report:
 - i) “conversion-not-performed” for the *non-delivery-reason-code*; and
 - ii) “punctuation-symbol-loss” for the *non-delivery-diagnostic-code*;
- d) if the abstract-value of the element *conversion-with-loss-prohibited* is “prohibited” and at least one alphabetical character in the text is not authorized in Annex 10, Volume II, 4.1.2:
 - 1) rejection of the message for all the message recipients; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in all the *per-recipient-fields* of the report:
 - i) “conversion-not-performed” for the *non-delivery-reason-code*; and

- ii) “alphabetical-character-loss” for the *non-delivery-diagnostic-code*;
or
- e) if several of the conditions under b) to d) above are simultaneously met:
 - 1) rejection of the message for all the message recipients; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in all the *per-recipient-fields* of the report:
 - i) “conversion-not-performed” for the *non-delivery-reason-code*; and
 - ii) “multiple-information-loss” for the *non-delivery-diagnostic-code*.

3.1.2.3.5.2.1.7 A message which was not rejected as the result of 3.1.2.3.5.2.1.6 shall be processed in one of three mutually exclusive manners:

- a) if the length of the ATS-Message-Text element exceeds 1800 characters, and if, due to system resource limitation, the procedure proposed in Annex 10, Volume II, Attachment D cannot be properly achieved by the AFTN/AMHS Gateway:
 - 1) rejection of the message for all the message recipients; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in all the *per-recipient-fields* of the report:
 - i) “unable-to-transfer” for the *non-delivery-reason-code*;
 - ii) “content-too-long” for the *non-delivery-diagnostic-code*; and
 - iii) “unable to convert to AFTN due to message text length” for the *supplementary-information*.
- b) if the length of the ATS-Message-Text element exceeds 1800 characters, and if the procedure proposed in Annex 10, Volume II, Attachment D is applied in the AFTN/AMHS Gateway:
 - 1) splitting of the message, internally to the Message Transfer and Control Unit, into several messages in accordance with the aforementioned Annex 10 procedure:
 - i) each of the resulting messages having for conversion purposes the same Message Transfer Envelope, the same IPM Heading and the ATS-Message-Header as the message subject to the splitting; and
 - ii) only the ATS-Message-Text element varying between the different resulting messages; and

- 2) processing of each of these messages as specified in 3.1.2.3.5.2.1.8; or
- c) processing as specified in 3.1.2.3.5.2.1.8 if the length of the ATS-Message-Text element does not exceed 1800 characters.

3.1.2.3.5.2.1.8 A message resulting from the situations in items b) and c) of 3.1.2.3.5.2.1.7 above shall be processed in one of three manners, depending on the number of message recipients towards which the Message Transfer and Control Unit is responsible for conveyance of the message, and on the AFTN/AMHS Gateway resources:

- a) if this number exceeds 21 message recipients:
 - 1) attempt to split the message, internally to the Message Transfer and Control Unit, into several messages, each of them with no more than 21 message recipients:
 - i) each of the resulting messages having for conversion purposes the same *per-message-fields* in the Message Transfer Envelope, and the same content as the message subject to the splitting; and
 - ii) only the *per-recipient-fields* elements in the Message Transfer Envelope varying between the different resulting messages; and
 - 2) processing of each of these messages as specified in 3.1.2.3.5.2.2 to 3.1.2.3.5.2.4;
- b) if this number exceeds 21 message recipients, and if, due to system resource limitation, the splitting attempt made by the gateway as specified in item a) above cannot be properly achieved:
 - 1) rejection of the message for all the message recipients; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in all the *per-recipient-fields* of the report:
 - i) “unable-to-transfer” for the *non-delivery-reason-code*;
 - ii) “too-many-recipients” for the *non-delivery-diagnostic-code*; and
 - iii) “unable to convert to AFTN due to number of recipients” for the *supplementary-information*; or
- c) processing as specified in 3.1.2.3.5.2.2 to 3.1.2.3.5.2.4, if this number does not exceed 21 message recipients.

Note 1.— In the processing defined in item a), the per-recipient-fields related to a particular recipient remain unchanged by the splitting. This applies in particular to the originally-specified-recipient-number, which is not altered by the processing specified in this clause.

Note 2.— The combination of 3.1.2.3.5.2.1.7 and 3.1.2.3.5.2.1.8 above may result in a very high number of AFTN messages being generated from one single AMHS message. Items 3.1.2.3.5.2.1.7 a) and 3.1.2.3.5.2.1.8 b) may, as a local matter, be used under such circumstances.

3.1.2.3.5.2.2 Generation of AFTN Message

3.1.2.3.5.2.2.1 Each message resulting from the processing specified in 3.1.2.3.5.2.1 above shall be converted by the Message Transfer and Control Unit into an AFTN Message composed of elements as specified in Table 3.1.2-10.

3.1.2.3.5.2.2.2 Those components which are classified as “G” in the column “action” of Table 3.1.2-10 shall be generated in compliance with the provisions of Annex 10, Volume II referred to in the column “mapping”.

3.1.2.3.5.2.2.3 Those components which are classified as “T” or “T1” in the column “action” of Table 3.1.2-10 shall be converted from the AMHS parameter specified in the column “converted from AMHS parameter” of Table 3.1.2-10 and according to the specification in the clause referred to in the column “mapping”.

Table 3.1.2-10. AFTN Message Generation

AFTN Message Part	Component	Action	Converted from AMHS parameter	Mapping
Heading	Start-of-Heading Character	X	-	-
	Transmission Identification	X	-	see 3.1.2.3.5.2.2.4
Address	Alignment Function	G	-	see Annex 10, Vol. II, 4.4.15.2.1
	Priority Indicator	T	ATS-Message-Priority (see Table 3.1.2-11/Part 6/1.2)	see 3.1.2.3.5.2.2.5
	Addressee Indicator(s)	T	recipient-name (see Table 3.1.2-12/Part 1/1.2.1)	see 3.1.2.3.5.2.2.6.2
	Alignment Function	G	-	see Annex 10, Vol. II, 4.4.15.2.1
Origin	Filing Time	T	ATS-Message-Filing-Time (see Table 3.1.2-11/Part 6/1.3)	see 3.1.2.3.5.2.2.7
	Originator Indicator	T	originator-name (see Table 3.1.2-12/Part 1/1.1.2)	see 3.1.2.3.5.2.2.6.1
	Priority Alarm	G	-	see Annex 10, Vol. II, 4.4.15.2.2
	Optional Heading Information	T1	ATS-Message-Optional-Heading-Info (see Table 3.1.2-11/Part 6/1.4)	see 3.1.2.3.5.2.2.8
	Alignment Function	G	-	see Annex 10, Vol. II, 4.4.15.2.2
	Start-of-Text Character	G	-	see Annex 10, Vol. II, 4.4.15.2.2
Text		T	ATS-Message-Text (see Table 3.1.2-11/Part 6/2)	see 3.1.2.3.5.2.2.9

AFTN Message Part	Component	Action	Converted from AMHS parameter	Mapping
Ending	Alignment Function	G	-	see Annex 10, Vol. II, 4.4.15.3.12
	Page-feed sequence	G	-	see Annex 10, Vol. II, 4.4.15.3.12
	End-of-Text Character	G	-	see Annex 10, Vol. II, 4.4.15.3.12

Legend: (see 3.1.1)

X = excluded (not used)

T1 = conditionally translated

G = generated

T = translated

3.1.2.3.5.2.2.4 As specified in 3.1.2.3.2.5.3, the element transmission identification shall be:

- a) generated by the AFTN Component rather than by the Message Transfer and Control Unit; and
- b) returned to the Message Transfer and Control Unit as the result of the operation transferring the generated AFTN Message from the Message Transfer and Control Unit to the AFTN Component.

3.1.2.3.5.2.2.5 The value of the priority indicator of the converted AFTN message shall be the value of the priority-indicator in the ATS-message-priority element of the AMHS message.

3.1.2.3.5.2.2.6 The value of an AF-Address included in the converted AFTN message shall be converted from an MF-Address as respectively specified in 3.1.2.3.5.2.2.6.1 and 3.1.2.3.5.2.2.6.2 depending whether it is an originator MF-Address or a recipient MF-Address.

3.1.2.3.5.2.2.6.1 The originator MF-Address included in an AMHS message shall be processed for translation into the originator indicator of the converted AFTN Message in one of three mutually exclusive manners, depending on the value of the *organization-name* attribute and on the contents of the User address look-up table, after preliminary conversion of the value of all AMHS address attributes from lower case IA5IRV characters, if any, to upper case IA5IRV characters:

- a) allocation of the value of the first element of the *organizational-unit-names* attribute to the originator indicator of the converted AFTN Message, if this value is a syntactically valid AF-Address and if the *organization-name* attribute has the value "AFTN";
- b) determination of an AF-Address matching exactly the MF-Address of the originator in the User address look-up table maintained in the Message Transfer and Control Unit, if the value of the *organization-name* attribute differs from "AFTN" and if such an exact match can be found; or
- c) if none of the conditions in a) and b) can be met, then:
 - 1) rejection of the message for all the message recipients; and

- 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in all the *per-recipient-fields* of the report:
 - i) “unable-to-transfer” for the *non-delivery-reason-code*;
 - ii) “invalid-arguments” for the *non-delivery-diagnostic-code*; and
 - iii) “unable to convert to AFTN due to unrecognized originator O/R address” for the *supplementary-information*.

3.1.2.3.5.2.2.6.2 To build the address part of the converted AFTN Message as specified in Annex 10, Volume II, 4.4.15.2.1, each of the recipient MF-Addresses included in an AMHS message, whose *responsibility* element in the *per-recipient-indicators* has the abstract-value “responsible”, shall be processed for translation into an addressee indicator in one of three mutually exclusive manners:

- a) allocation of the value of the first element of the *organizational-unit-names* attribute, converted from lower case IA5IRV characters, if any, to upper case IA5IRV characters, to an addressee indicator in the converted AFTN Message, if this value is a syntactically valid AF-Address and if the *organization-name* attribute has the value “AFTN”;
- b) determination of an AF-Address matching exactly the MF-Address of the recipient in the User address look-up table maintained in the Message Transfer and Control Unit, if the value of the *organization-name* attribute differs from “AFTN” and if such an exact match can be found; or
- c) if none of the conditions in a) and b) can be met, then:
 - 1) rejection of the message for the considered message recipient; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in all the *per-recipient-fields* of the report:
 - i) “unable-to-transfer” for the *non-delivery-reason-code*; and
 - ii) “unrecognised-OR-name” for the *non-delivery-diagnostic-code*.

Note.— Although the potential generation of a non-delivery report is mentioned for each recipient-name which cannot be properly translated into an AF-Address, a single report with different per-recipient-fields may be generated for all recipient-names which cannot be translated.

3.1.2.3.5.2.2.7 The value of the filing time of a converted AFTN message shall be the value of the filing-time component in the ATS-Message-Filing-Time element of the AMHS message.

3.1.2.3.5.2.2.8 The Optional Heading Information of a converted AFTN message shall either:

- a) take the value of the optional-heading-information in the ATS-Message-Optional-Heading-Info element, if this element is present; or
- b) be omitted in the converted AFTN message, if the ATS-Message-Optional-Heading-Info element is absent from the AMHS message.

3.1.2.3.5.2.2.9 The content of the Text part of a converted AFTN message shall be derived from the value of the ATS-Message-Text element of the IPM text of the AMHS message, in compliance with the following procedure:

- a) conversion of each character which is not in the IA5IRV character repertoire, into an IA5IRV character according to the locally defined conversion rules;
- b) conversion of each IA5IRV character, if it is in lower case, into the equivalent upper case character;
- c) replacement by question-marks (“?”) of all characters or character sequences in the text, if any, of which the use is not authorized in Annex 10, Volume II, 4.1.2;
- d) folding of any line longer than 69 characters; and
- e) allocation of the result of items a) to d) above to the Text part of the converted AFTN message.

Note 1.— The locally defined conversion rules mentioned in item a) may be for example CCITT Recommendation X.408, if support of the ISO 8859-1 character set is a local policy of the AMHS Management Domain.

Note 2.— A lower case IA5IRV character is one whose position is between 6/1 and 6/15 or 7/0 and 7/10. The corresponding upper case IA5IRV characters have positions extending from 4/1 to 4/15 and 5/0 to 5/10.

3.1.2.3.5.2.3 Use of IPM elements

3.1.2.3.5.2.3.1 Each of the elements composing the IPM in an AMHS message to be converted into an AFTN message in the Message Transfer and Control Unit shall be processed as specified in the column “action” of Table 3.1.2-11.

3.1.2.3.5.2.3.2 The elements composing the IPM shall be used according to the specification in the clause referred to in the column “mapping” of Table 3.1.2-11.

Note 1.— Table 3.1.2-11 is structured as a PRL derived from the profile specification included in 2.2 and consequently from the SPICS Proforma included in ISO/IEC ISP 12062-2 as well as from Table 3.1.2-2 in 3.1.2.2.3.2. The columns “Base” and “ISP” under “Reception” are extracted from ISO/IEC ISP 12062-2 and the column “Basic ATS Message Service” specifies the static capability of an IPM AU supporting the Basic ATS Message Service, i.e. the ability to handle in reception the element as part of an

IPM carrying an ATS Message. The references to the ISP Profile are indicated in the part titles as AMH21/ref where appropriate. The references in column Ref are those of the ISP.

Table 3.1.2-11. Use of IPM Elements

Ref	Element	Reception			Action	Mapping / Notes
		Base	ISP	Basic ATS Mess. Service		
PART 1 : AMH21/A.1.1 SUPPORTED INFORMATION OBJECTS						
1	Interpersonal Message (IPM)	M	M	M	T	see Part 1/1.1 and 1.2
1.1	heading	M	M	M	T	see Part 2
1.2	body	M	M	M	T	see Part 3
2	Interpersonal Notification (IPN)	O	M	M	-	out of the scope of this clause
PART 2 : AMH21/A.1.2 IPM HEADING FIELDS						
1	this-IPM	M	M	M	D	-
2	originator	M	M	M	D	-
3	authorizing-users	M	M	M	D	-
4	primary-recipients	M	M	M	D	see 3.1.2.3.5.2.3.3 and Part 5/1
5	copy-recipients	M	M	M	D	see 3.1.2.3.5.2.3.3 and Part 5/1
6	blind-copy-recipients	M	M	M	D	see 3.1.2.3.5.2.3.3 and Part 5/1
7	replied-to-IPM	M	M	M	D	-
8	obsoleted-IPMs	M	M	M	D	-

Ref	Element	Reception			Action	Mapping / Notes
		Base	ISP	Basic ATS Mess. Service		
9	related-IPMs	M	M	M	D	-
10	subject	M	M	M	D	-
11	expiry-time	M	M	M	D	-
12	reply-time	M	M	M	D	-
13	reply-recipients	M	M	M	D	-
14	importance	M	M	M	D	-
15	sensitivity	M	M	M	D	-
16	auto-forwarded	M	M	M	D	-
17	extensions	M	M	M	D	-
17.1	incomplete-copy	O	M	M	D	-
17.2	languages	M	M	M	D	-
17.3	auto-submitted	O	I	I	D	-
PART 3 : AMH21/A.1.3 IPM BODY						
1	ia5-text	O	M	M	T	see Part 3/1.1 and 1.2
1.1	parameters	M	M	M	D	-
1.1.1	repertoire	M	M	M	D	-
1.2	data	M	M	M	T	see Part 6
2	voice	I	I	I	X	see Note 2

Ref	Element	Reception			Action	Mapping / Notes
		Base	ISP	Basic ATS Mess. Service		
3	g3-facsimile	O	O	O	X	see Note 2
4	g4-class-1	O	O	O	X	see Note 2
5	teletex	O	O	O	X	see Note 2
6	videotex	O	O	O	X	see Note 2
7	encrypted	I	I	I	X	see Note 2
8	message	O	M	M	X	see Note 2
9	mixed-mode	O	O	O	X	see Note 2
10	bilaterally-defined	O	O	O	X	see Note 2
11	nationally-defined	O	O	O	X	see Note 2
12	externally-defined	O	M	M	X/T	see Note 3 and Part 4
PART 4 : AMH21/A.1.3.1 EXTENDED BODY PART SUPPORT						
1	ia5-text-body-part	O	M	M	T	see Part 3/1
2	g3-facsimile-body-part	O	O	O	X	see Note 2
3	g4-class1-body-part	O	O	O	X	see Note 2
4	teletex-body-part	O	O	O	X	see Note 2
5	videotex-body-part	O	O	O	X	see Note 2
6	encrypted-body-part	I	I	I	X	see Note 2
7	message-body-part	O	M	M	X	see Note 2

Ref	Element	Reception			Action	Mapping / Notes
		Base	ISP	Basic ATS Mess. Service		
8	mixed-mode-body-part	O	O	O	X	see Note 2
9	bilaterally-defined-body-part	O	O	O	X	see Note 2
10	nationally-defined-body-part	O	O	O	X	see Note 2
11	general-text-body-part	O	M	M	T/X	see 3.1.2.3.5.2.1.4, 3.1.2.3.5.2.3.4 and Part 6
12	file-transfer-body-part	O	I	I	X	see Note 2
13	voice-body-part	O	I	I	X	see Note 2
14	oda-body-part	O	O	O	X	see Note 2
PART 5 : AMH21/A.1.5 COMMON DATA TYPES						
1	RecipientSpecifier					
1.1	recipient	M	M	M	D	-
1.2	notification-requests	M	M	M	D	see Part 5/1.2.1-1.2.3
1.2.1	rn	O	O	O	D	see 3.1.2.3.5.2.3.3
1.2.2	nrn	M	M	M	D	-
1.2.3	ipm-return	O	O	O	D	-
1.3	reply-requested	M	M	M	D	-
1.4	recipient-extensions	O	I	I	D	-

Ref	Element	Reception			Action	Mapping / Notes
		Base	ISP	Basic ATS Mess. Service		
PART 6 : IPM SUPPORT OF THE BASIC ATS MESSAGE SERVICE						
1	ATS-Message-Header	-	-	M	T	see Part 6/1.1-1.6
1.1	start-of-heading	-	-	M	-	-
1.2	ATS-Message-Priority	-	-	M	T	see Part 6/1.2.1-1.2.3
1.2.1	priority-prompt	-	-	M	-	-
1.2.2	priority-indicator	-	-	M	T	see 3.1.2.3.5.2.2.5 and 3.1.2.3.5.2.3.3
1.2.3	priority-separator	-	-	M	-	-
1.3	ATS-Message-Filing-Time	-	-	M	T	see Part 6/1.3.1-1.3.3
1.3.1	filing-time-prompt	-	-	M	-	-
1.3.2	filing-time	-	-	M	T	see 3.1.2.3.5.2.2.7
1.3.3	filing-time-separator	-	-	M	-	-
1.4	ATS-Message-Optional-Heading-Info	-	-	M	T1	see Part 6/1.4.1-1.4.3
1.4.1	OHI-prompt	-	-	M	-	-
1.4.2	optional-heading-information	-	-	M	T	see 3.1.2.3.5.2.2.8
1.4.3	OHI-separator	-	-	M	-	-
1.5	end-of-heading-blank-line	-	-	M	-	-
1.6	start-of-text	-	-	M	-	-

Ref	Element	Reception			Action	Mapping / Notes
		Base	ISP	Basic ATS Mess. Service		
2	ATS-Message-Text	-	-	M	T	see 3.1.2.3.5.2.2.9

Legend (see 3.1.1) :

- M = mandatory support
- O = optional support
- I = out of scope
- = not applicable
- T1 = conditionally translated
- D = discarded
- T = translated
- X = excluded

Note 2.— This body part type is excluded as the result of 3.1.2.3.5.2.1.4.

Note 3.— This body part type may be either excluded or translated, depending on whether or not it is a standard extended body part type, and if yes, depending on the type of extended body part type, as specified in Part 4 and as the result of 3.1.2.3.5.2.1.4.

3.1.2.3.5.2.3.3 If the priority-indicator of a received AMHS message has the value “SS” and if the *responsibility* element of the corresponding *per-recipient-fields* of the Message Transfer Envelope has the value “responsible”, then an error situation shall be logged and reported to a control position for appropriate action if any of the following situations, or both, occurs:

- a) if the *notification-requests* element of either a *primary-recipient*, or a *copy-recipient*, or a *blind-copy-recipient* element has an abstract-value different from “rn”; or
- b) if the *priority* element of the Message Transfer Envelope has an abstract-value different from “urgent”.

Note 1.— The Message Transfer and Control Unit generates RNs only for SS priority messages, since they are the only messages for which an end-to-end acknowledgement is possible in the AFTN. A receipt-notification-request included in a message with another priority is ignored, considering that the Message Transfer and Control Unit cannot ensure the actual reception of the message by the end-user.

Note 2.— The above specified error situation, if any, does not cause message rejection.

3.1.2.3.5.2.3.4 The components of a general-text body part shall be used as follows for the conversion of the IPM body into the text of the AFTN Message:

- a) the parameters component identify the character set used for the message, as specified in ISO/IEC 10021-7, B.2; and

- b) the data component of a general-text body part are used for the generation of the converted AFTN message as specified in Part 6 of Table 3.1.2-11.

3.1.2.3.5.2.4 Use of Message Transfer Envelope parameters

3.1.2.3.5.2.4.1 Each of the elements composing the Message Transfer Envelope of an AMHS message to be converted into an AFTN message in a Message Transfer and Control Unit shall be processed as specified in the column “action” of Table 3.1.2-12.

3.1.2.3.5.2.4.2 The elements composing the Message Transfer Envelope shall be handled according to the specification in the clause referred to in the column “mapping” of Table 3.1.2-12.

Note 1.— Table 3.1.2-12 is structured as a PRL derived from the ISPICS Proforma included in ISO/IEC ISP 10611-3. The columns “Base” and “ISP” are extracted from ISO/IEC ISP 10611-3 and the column “Basic ATS Message Service” specifies the static capability of an AU in relation with the MT-EoS (Message Transfer Elements of Service), i.e. the ability to convey, handle and act in relation with the element. The references to the ISP Profile are indicated in the part titles as AMH11/ref where appropriate.

Note 2.— Although not used for mapping, some elements may generate specific actions for the gateway in the handling of the considered message.

Note 3.— Some elements may have two classifications, e.g. D/X where certain values of the element may cause message rejection, while other values are simply discarded when the AMHS message is converted into an AFTN message.

Table 3.1.2-12. Use of the Message Transfer Envelope

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Mapping / Notes
PART 1 : AMH11/A.1.4.2 MESSAGE TRANSFER						
1	MessageTransferEnvelope	M	M	M	T	see Part 1/1.1 and 1.2
1.1	(per message fields)					
1.1.1	message-identifier	M	M	M	D	-
1.1.2	originator-name	M	M	M	T	see 3.1.2.3.5.2.2.6.1
1.1.3	original-encoded-information-types	M	M-	M-	D/X	see 3.1.2.3.5.2.1.2
1.1.4	content-type	M	M-	M-	D/X	see 3.1.2.3.5.1.1
1.1.5	content-identifier	M	M	M	D	-

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Mapping / Notes
1.1.6	priority	M	M	M	D	-
1.1.7	per-message-indicators	M	M	M	D	see Part 2/4
1.1.8	deferred-delivery-time	O	M-	M-	D	see 3.1.2.3.5.2.4.4
1.1.9	per-domain-bilateral-information	O	M-	M-	D	see 3.1.2.3.5.2.4.5 and Part 2/5
1.1.10	trace-information	M	M	M	D	see Part 2/6
1.1.11	extensions	M	M	M	D/X	see 3.1.2.3.5.2.4.6 and Part 3/1
1.1.11.1	recipient-reassignment-prohibited	O	M	M	D	see 3.1.2.3.5.2.4.3
1.1.11.2	dl-expansion-prohibited	O	M	M	D	see 3.1.2.3.5.2.4.7
1.1.11.3	conversion-with-loss-prohibited	O	M	M	D/X	see 3.1.2.3.5.2.1.6
1.1.11.4	latest-delivery-time	O	M-	M-	D/X	see 3.1.2.3.5.2.4.8
1.1.11.5	originator-return-address	O	M-	M-	D	-
1.1.11.6	originator-certificate	O	M-	M-	X	see 3.1.2.3.5.2.4.9
1.1.11.7	content-confidentiality-algorithm-identifier	O	M-	M-	X	see 3.1.2.3.5.2.4.9
1.1.11.8	message-origin-authentication-check	O	M-	M-	X	see 3.1.2.3.5.2.4.9
1.1.11.9	message-security-label	O	M-	M-	X	see 3.1.2.3.5.2.4.9
1.1.11.10	content-correlator	M	M	M	D	-
1.1.11.11	dl-expansion-history	M	M-	M-	D	-
1.1.11.12	internal-trace-information	M	M	M	D	-
1.2	per-recipient-fields	M	M	M	T	see Part 1/1.2.1-1.2.5
1.2.1	recipient-name	M	M	M	T	see 3.1.2.3.5.2.2.6.2

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Mapping / Notes
1.2.2	originally-specified-recipient-number	M	M	M	D	-
1.2.3	per-recipient-indicators	M	M	M	D	-
1.2.4	explicit-conversion	O	M-	M-	D	-
1.2.5	extensions	M	M	M	D/X	see 3.1.2.3.5.2.4.6 and Part 3/1
1.2.5.1	originator-requested-alternate-recipient	O	M-	M-	D	see 3.1.2.3.5.2.4.3
1.2.5.2	requested-delivery-method	O	M-	M-	D	see 3.1.2.3.5.2.4.10
1.2.5.3	physical-forwarding-prohibited	O	M-	M-	X	see 3.1.2.3.5.2.4.11
1.2.5.4	physical-forwarding-address-request	O	M-	M-	X	see 3.1.2.3.5.2.4.11
1.2.5.5	physical-delivery-modes	O	M-	M-	X	see 3.1.2.3.5.2.4.11
1.2.5.6	registered-mail-type	O	M-	M-	X	see 3.1.2.3.5.2.4.11
1.2.5.7	recipient-number-for-advice	O	M-	M-	X	see 3.1.2.3.5.2.4.11
1.2.5.8	physical-rendition-attributes	O	M-	M-	X	see 3.1.2.3.5.2.4.11
1.2.5.9	physical-delivery-report-request	O	M-	M-	X	see 3.1.2.3.5.2.4.11
1.2.5.10	message-token	O	M-	M-	X	see 3.1.2.3.5.2.4.9
1.2.5.11	content-integrity-check	O	M-	M-	X	see 3.1.2.3.5.2.4.9
1.2.5.12	proof-of-delivery-request	O	M-	M-	X	see 3.1.2.3.5.2.4.9
1.2.5.13	redirection-history	M	M-	M-	D	-
2	content	M	M	M	T	see 3.1.2.3.5.2.3

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Mapping / Notes
PART 2 : AMH11/A.1.5 COMMON DATA TYPES						
4	PerMessageIndicators					
4.1	disclosure-of-other-recipients	M	M	M	D	-
4.2	implicit-conversion-prohibited	M	M	M	D/X	see 3.1.2.3.5.2.1.1
4.3	alternate-recipient-allowed	M	M	M	D	see 3.1.2.3.5.2.4.3
4.4	content-return-request	O	M-	M-	D	-
4.5	reserved	O	M-	M-	D	-
4.6	bit-5	O	M-	M-	D	-
4.7	bit-6	O	M-	M-	D	-
4.8	service-message	O	M-	M-	D	-
5	PerDomainBilateralInformation					
5.1	country-name	M	M-	M-	D	see 3.1.2.3.4.2.4.5
5.2	administration-domain-name	M	M-	M-	D	see 3.1.2.3.4.2.4.5
5.3	private-domain-identifier	O	M-	M-	D	see 3.1.2.3.4.2.4.5
5.4	bilateral-information	M	M-	M-	D	see 3.1.2.3.4.2.4.5
6	TraceInformation					
6.1	TraceInformationElement	M	M	M	D	-

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Mapping / Notes
6.1.1	global-domain-identifier	M	M	M	D	-
6.1.2	domain-supplied-information	M	M	M	D	-
6.1.2.1	arrival-time	M	M	M	D	-
6.1.2.2	routing-action	M	M	M	D	-
6.1.2.2.1	relayed	M	M	M	D	-
6.1.2.2.2	rerouted	O	C1	C1	D	-
6.1.2.3	attempted-domain	O	C1	C1	D	-
6.1.2.4	(additional actions)				D	-
6.1.2.4.1	deferred-time	M	C2	C2	D	-
6.1.2.4.2	converted-encoded-information-types	O	M-	M-	D	see 3.1.2.3.5.2.1.2
6.1.2.4.3	other-actions	O	M-	M-	D	-
6.1.2.4.3.1	redirected	O	M-	M-	D	-
6.1.2.4.3.2	dl-operation	O	M-	M-	D	-
PART 3 : AMH11/A.1.6 EXTENSION DATA TYPES						
1	ExtensionField					
1.1	type	M	M	M	D/X	see Part 3/1.1.1 and 1.1.2
1.1.1	standard-extension	M	M	M	D/X	see 3.1.2.3.5.2.4.6
1.1.2	private-extension	O	M-	M-	D/X	see 3.1.2.3.5.2.4.6

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Mapping / Notes
1.2	criticality	M	M	M	D/X	see 3.1.2.3.5.2.4.6
1.3	value	M	M	M	D	-

Legend (see 3.1.1) :

- M = mandatory support
- M- = minimal mandatory support
- O = optional support
- C1 = if rerouting is supported then M else M-
- C2 = if deferred delivery is supported then M else M-
- D = discarded
- T = translated
- X = excluded

3.1.2.3.5.2.4.3 The elements *alternate-recipient-allowed* and *originator-requested-alternate-recipient* shall be discarded by the Message Transfer and Control Unit, since the optional Redirection Functional Group, if implemented in an AFTN/AMHS Gateway, is supported by the ATN Component and not by the Message Transfer and Control Unit.

3.1.2.3.5.2.4.4 The element *deferred-delivery-time* shall be discarded by the Message Transfer and Control Unit, since this functionality, if implemented in an AFTN/AMHS Gateway, is supported by the ATN Component and not by the Message Transfer and Control Unit.

3.1.2.3.5.2.4.5 For mapping purposes the whole *per-domain-bilateral-information* element shall be discarded.

Note.— If the elements country-name, administration-domain-name and private-domain-identifier in an element of the per-domain-bilateral-information together identify the AMHS Management Domain operating the AFTN/AMHS Gateway, the use made of the bilateral-information element is a local matter.

3.1.2.3.5.2.4.6 If any extension-field is present in the *extensions* of the Message Transfer Envelope and not semantically understood by the Message Transfer and Control Unit, then the element shall either:

- a) cause the following actions to be performed if its criticality is set to “CRITICAL FOR TRANSFER” or to “CRITICAL FOR DELIVERY”:
 - 1) message rejection of the message for either:
 - i) all the message recipients if the extension is part of the *per-message-fields*; or
 - ii) the considered message recipient if the extension is part of the *per-recipient-fields*; and

- 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in the appropriate *per-recipient-fields* of the report:
 - i) “unable-to-transfer” for the *non-delivery-reason-code*; and
 - ii) “unsupported-critical-function” for the *non-delivery-diagnostic-code*; or
- b) be simply discarded if there is no criticality given.

3.1.2.3.5.2.4.7 The element *dl-expansion-prohibited* shall be discarded by the Message Transfer and Control Unit, since the DL-expansion capability of an AFTN/AMHS Gateway is supported by the ATN Component and not by the Message Transfer and Control Unit.

3.1.2.3.5.2.4.8 If the *latest-delivery-time* element is present, and if, when the AMHS message is handled by the Message Transfer and Control Unit, the current time exceeds the value of the *latest-delivery-time*, then the following actions shall be performed:

- a) message rejection for all the message recipients; and
- b) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in the appropriate *per-recipient-fields* of the report:
 - 1) “transfer-failure” for the *non-delivery-reason-code*; and
 - 2) “maximum-time-expired” for the *non-delivery-diagnostic-code*.

3.1.2.3.5.2.4.9 The Message Transfer and Control Unit does not implement Security Elements of Service. Thus, if any security-related extension-field set to “CRITICAL FOR DELIVERY” is present in the *extensions* of the Message Transfer Envelope, the following actions shall be performed:

- a) message rejection of the message for either:
 - 1) all the message recipients if the extension is part of the *per-message-fields*; or
 - 2) the considered message recipient if the extension is part of the *per-recipient-fields*; and
- b) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in the appropriate *per-recipient-fields* of the report:
 - 1) “unable-to-transfer” for the *non-delivery-reason-code*; and
 - 2) “unsupported-critical-function” for the *non-delivery-diagnostic-code*.

3.1.2.3.5.2.4.10 The element *requested-delivery-method* shall be discarded by the Message Transfer and Control Unit.

Note.— *The Message Transfer and Control Unit handles the message irrespective of the value of this attribute, since it indicates only a preferred delivery method (see Technical Corrigendum 5 to ISO/IEC 10021-4).*

3.1.2.3.5.2.4.11 The Message Transfer and Control Unit does not implement Physical Delivery Elements of Service. Thus, if any physical delivery-related extension-field set to “CRITICAL FOR DELIVERY” is present in the *extensions* of the Message Transfer Envelope, the following actions shall be performed:

- a) message rejection of the message for either:
 - 1) all the message recipients if the extension is part of the *per-message-fields*;
or
 - 2) the considered message recipient if the extension is part of the *per-recipient-fields*; and
- b) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in the appropriate *per-recipient-fields* of the report:
 - 1) “physical-rendition-not-performed” for the *non-delivery-reason-code*; and
 - 2) “unsupported-critical-function” for the *non-delivery-diagnostic-code*.

3.1.2.3.5.3 AMHS RN Conversion

Upon reception by the Message Transfer and Control Unit of a RN conveyed with a Message Transfer Envelope passed from the ATN Component, for the acknowledgement of a SS message, this message shall be converted into an AFTN acknowledgement message in compliance with the following:

- a) the specification of the initial processing performed to determine the Message Transfer and Control Unit ability to convert the RN, as included in 3.1.2.3.5.3.1;
- b) the specification of how the AFTN service message is generated and how the AFTN service message components are mapped from AMHS parameters, as included in 3.1.2.3.5.3.2;
- c) the specification of how the elements of the received RN are handled, as included in 3.1.2.3.5.3.3; and
- d) the specification of how the Message Transfer Envelope elements are handled, as included in 3.1.2.3.5.3.4.

3.1.2.3.5.3.1 Initial processing of AMHS Receipt Notifications

3.1.2.3.5.3.1.1 Upon reception by the Message Transfer and Control Unit of a RN, passed from the ATN Component to be potentially converted into an AFTN acknowledgement message, the received RN shall be processed in one of the following manners:

- a) processing as specified in 3.1.2.3.5.3.1.2, if the subject IPM has been previously generated by the Message Transfer and Control Unit; or
- b) unsuccessful termination of the procedure, if the subject IPM has not been previously generated by the Message Transfer and Control Unit, resulting in:
 - 1) logging of the error situation and reporting to a control position;
 - 2) storage of the RN for appropriate action at the control position; and
 - 3) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values:
 - i) “unable-to-transfer” for the *non-delivery-reason-code*;
 - ii) “invalid-arguments” for the *non-delivery-diagnostic-code*; and
 - iii) “unable to convert RN to AFTN Ack service message due to misrouted RN” for the *supplementary-information*.

3.1.2.3.5.3.1.2 For an AMHS RN passed from the ATN Component to the Message Transfer and Control Unit and not rejected as the result of 3.1.2.3.5.3.1.1, the received RN shall be processed in one of the following manners:

- a) processing as specified in 3.1.2.3.5.3.1.3, if the value of the priority indicator of the subject AFTN message was “SS”; or
- b) unsuccessful termination of the procedure, if the value of the priority indicator was different from “SS”, resulting in:
 - 1) logging of the error situation and reporting to a control position; and
 - 2) storage of the RN for appropriate action at the control position.

3.1.2.3.5.3.1.3 An AMHS RN passed from the ATN Component to the Message Transfer and Control Unit and not rejected as the result of 3.1.2.3.5.3.1.2 shall be processed as specified in 3.1.2.3.5.3.2.

3.1.2.3.5.3.2 Generation of the AFTN acknowledgement message

3.1.2.3.5.3.2.1 An AMHS RN received by the Message Transfer and Control Unit and not rejected as the result of 3.1.2.3.5.3.1 shall be converted into an AFTN acknowledgement message in compliance with:

- a) the specification of 3.1.2.3.5.2.2 with the exception of the components listed in Table 3.1.2-13; and
- b) the classification of the components included in Table 3.1.2-13, as specified in the column “action” of Table 3.1.2-13.

3.1.2.3.5.3.2.2 These components which are classified as “G” shall be generated in compliance with the clause referred to in the column “mapping” of Table 3.1.2-13.

3.1.2.3.5.3.2.3 These components which are classified as “T” shall be converted from the AMHS parameter specified in the column “converted from AMHS parameter” of Table 3.1.2-13 and according to the specification in the clause referred to in the column “mapping”.

Table 3.1.2-13. Generation of AFTN acknowledgement message

AFTN Message Part	Component	Action	converted from AMHS parameter	Mapping
Address	Priority Indicator	G	-	see 3.1.2.3.5.3.2.4
Origin	Filing Time	T	receipt-time (see Table 3.1.2-14/Part 1/7.1)	see 3.1.2.3.5.3.2.5
	Optional Heading Information	X	-	-
Text		G	-	see 3.1.2.3.5.3.2.6

Legend: (see 3.1.1)

G = generated

T = translated

X = excluded (not used)

3.1.2.3.5.3.2.4 In an AFTN acknowledgement message, generated as the result of the conversion of an AMHS RN message, the priority indicator component shall take the value SS.

3.1.2.3.5.3.2.5 In an AFTN acknowledgement message, generated as the result of the conversion of an AMHS RN message, the filing time component shall:

- a) be a date-time group as specified in Annex 10, Volume II, 4.4.15.2.2.1; and
- b) take the value of the six characters between the fifth and tenth position from the *receipt-time* element of the RN.

3.1.2.3.5.3.2.6 In an AFTN acknowledgement message, generated as the result of the conversion of an AMHS RN message, the value of the Text component shall be generated as specified in Annex 10, Volume II, 4.4.15.6 using the origin of the subject AFTN message.

3.1.2.3.5.3.3 Use of RN fields

3.1.2.3.5.3.3.1 Each of the elements composing the RN to be converted into an AFTN acknowledgement message in an AFTN/AMHS Gateway shall be processed as specified in the column “action” of Table 3.1.2-14.

3.1.2.3.5.3.3.2 The elements composing the RN shall be handled according to the specification in the clause referred to in the column “mapping” of Table 3.1.2-14.

Note.— Table 3.1.2-14 is structured as a PRL derived from the profile specification included in 2.2 and consequently from the SPICS Proforma included in ISO/IEC ISP 12062-2 (AMH21). The columns “Base” and “ISP” under “Reception” are extracted from ISO/IEC ISP 12062-2, and the column “Basic ATS Message Service” specifies the static capability of an IPM AU supporting the Basic ATS Message Service, i.e. the ability to handle in reception the element as part of a RN. The references to the ISP Profile are indicated in the part titles as AMH21/ref where appropriate. The references in column Ref are those of the ISP.

Table 3.1.2-14. Use of RN fields

Ref	Element	Reception			Action	Mapping / Notes
		Base	ISP	Basic ATS Mess. Service		
PART 1: AMH21/A.1.4 IPN FIELDS						
1	subject-ipm	M	M	M	D	-
2	ipn-originator	M	M	M	D	-
3	ipm-preferred-recipient	M	M	M	D	-
4	conversion-eits	M	M	M	D	-
5	notification-extensions	O	I	I	-	-
6	non-receipt-fields	O	M	M	-	out of the scope of this clause
7	receipt-fields	O	M	M	T	see Part 1/7.1-7.4
7.1	receipt-time	M	M	M	T	see 3.1.2.3.5.3.2.5

Ref	Element	Reception			Action	Mapping / Notes
		Base	ISP	Basic ATS Mess. Service		
7.2	acknowledgment-mode	M	M	M	D	-
7.3	suppl-receipt-info	O	O	O	D	-
7.4	rn-extensions	O	I	I	-	-
8	other-notification-type-fields	O	I	I	-	-

Legend (see 3.1.1) :

- M = mandatory support
- O = optional support
- I = out of scope
- = not applicable
- D = discarded
- T = translated
- = out of scope

3.1.2.3.5.3.4 Use of Message Transfer Envelope parameters conveyed with a RN

3.1.2.3.5.3.4.1 The elements composing the Message Transfer Envelope conveyed with a RN to be converted into an AFTN acknowledgement message shall be used in compliance with:

- a) the specification of 3.1.2.3.5.2.4 with the exception of those elements included in Table 3.1.2-15; and
- b) the specification included in the clause referred to in the column “Mapping” of Table 3.1.2-15.

Note.— Table 3.1.2-15 is structured as an extraction of Table 3.1.2-12.

**Table 3.1.2-15. Use of the MessageTransfer Envelope conveyed with a RN
(differences from Table 3.1.2-12)**

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Mapping / Notes
PART 1 : AMH11/A.1.4.2 MESSAGE TRANSFER						
1	MessageTransferEnvelope	M	M	M	T	see Part 1/1.1 and 1.2

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Mapping / Notes
1.1	(per message fields)					
1.1.3	original-encoded-information-types	M	M-	M-	D	see 3.1.2.3.5.3.4.2
1.1.7	per-message-indicators	M	M	M	D	see Part 2/4
1.1.10	trace-information	M	M	M	D	see Part 2/6
1.2	per-recipient-fields	M	M	M	D	see Part 1/1.2.1
1.2.1	recipient-name	M	M	M	D	see 3.1.2.3.5.3.4.3
2	content	M	M	M	T	see 3.1.2.3.5.3.3
PART 2 : AMH11/A.1.5 COMMON DATA TYPES						
4	PerMessageIndicators					
4.2	implicit-conversion-prohibited	M	M	M	D	see 3.1.2.3.5.3.4.2
6	TraceInformation					
6.1	TraceInformationElement	M	M	M	D	-
6.1.2	domain-supplied-information	M	M	M	D	-
6.1.2.4	(additional actions)				D	-
6.1.2.4.2	converted-encoded-information-types	O	M-	M-	D	see 3.1.2.3.5.3.4.2

Legend (see 3.1.1) :

M = mandatory support
 M- = minimal mandatory support
 O = optional support
 D = discarded
 T = translated

3.1.2.3.5.3.4.2 The elements related to the encoded-information-types in the Message Transfer Envelope conveyed with a RN shall be discarded when converting the RN into an AFTN acknowledgement message.

3.1.2.3.5.3.4.3 The *recipient-name* element in the Message Transfer Envelope conveyed with a RN shall be discarded when converting the RN into an AFTN acknowledgement message.

Note.— The Message Transfer and Control Unit uses the information contained in the subject AFTN message to construct an AFTN acknowledgement message.

3.1.2.3.5.4 AMHS Non-delivery Report Conversion

Upon reception by the Message Transfer and Control Unit of an AMHS Non-Delivery Report passed from the ATN Component, this report shall be processed in compliance with the following:

- a) the specification of the initial processing performed to determine the Message Transfer and Control Unit ability to convert the report, as included in 3.1.2.3.5.4.1;
- b) the specification of how the AFTN service message is generated, if any, and how the AFTN service message components are mapped from AMHS parameters, as included in 3.1.2.3.5.4.2; and
- c) the specification of how the Report Transfer Envelope elements are handled, as included in 3.1.2.3.5.4.3.

3.1.2.3.5.4.1 Initial processing of AMHS Non-Delivery Reports

3.1.2.3.5.4.1.1 Upon reception by the Message Transfer and Control Unit of a non-delivery report, passed from the ATN Component to be potentially converted into an AFTN service message, the received non-delivery report shall be processed in one of the following manners:

- a) processing as specified in 3.1.2.3.5.4.1.2, if the subject AMHS message has been previously generated by the Message Transfer and Control Unit; or
- b) unsuccessful termination of the procedure, if the subject AMHS message has not been previously generated by the Message Transfer and Control Unit, resulting in:
 - 1) logging of the error situation and reporting to a control position; and
 - 2) storage of the non-delivery report for appropriate action at the control position.

3.1.2.3.5.4.1.2 A non-delivery report received by the Message Transfer and Control Unit, and regarding a subject message which had been generated by the Message Transfer and Control Unit, shall be processed by the Message Transfer and Control Unit in one of three mutually exclusive manners:

- a) processing as specified in 3.1.2.3.5.4.1.3 if there is no *originally-intended-recipient-name* element with a value different of the *actual-recipient-name* in any of the *per-recipient-fields* elements of the report;

- b) processing as follows, if at least one *originally-intended-recipient-name* element in one of the *per-recipient-fields* elements has a value different from the value of the *actual-recipient-name*, and if at least one *per-recipient-fields* element in the report does not meet the same condition:
 - 1) logging of the error situation and reporting to a control position;
 - 2) storage of the non-delivery report and of the corresponding *per-recipient-fields* elements for appropriate action at the control position;
 - 3) processing of the report as specified in 2.3.5.4.1.3 for the *per-recipient-fields* where there is no *originally-intended-recipient-name* element with a value different of the *actual-recipient-name*; or
- c) processing as follows, if all *per-recipient-fields* elements of the report include an *originally-intended-recipient-name* element which has a value different from the value of the *actual-recipient-name*:
 - 1) logging of the error situation and reporting to a control position;
 - 2) storage of the non-delivery report and of the corresponding *per-recipient-fields* elements for appropriate action at the control position.

3.1.2.3.5.4.1.3 If the non-delivery report did not cause any error situation to be reported, or for the *per-recipient-fields* of the report which did not cause any error to be reported, the report shall be processed by the Message Transfer and Control Unit in one of the following manners:

- a) conversion of the report into an unknown address AFTN service message as specified in 3.1.2.3.5.4.2, if the *non-delivery-diagnostic-code* has the abstract-value “unrecognised-OR-name”; or
- b) processing as follows, if the *non-delivery-diagnostic-code* has any abstract-value other than “unrecognised-OR-name”
 - 1) logging of the non-delivery situation and reporting to a control position;
 - 2) storage of the non-delivery report for appropriate action at the control position.

3.1.2.3.5.4.2 Generation of unknown address AFTN service message

3.1.2.3.5.4.2.1 An AMHS Non-Delivery Report received by the Message Transfer and Control Unit, which *non-delivery-diagnostic-code* has the abstract-value “unrecognised-OR-name”, and not stored for action at the control position as the result of 3.1.2.3.5.4.1, shall be converted into an AFTN service message to the originator of the subject AFTN message, indicating that an unknown addressee indicator was specified in the subject AFTN message (unknown address AFTN service message) in compliance with:

- a) the specification of Annex 10, Volume II, 4.4.11.13.3; and

- b) the classification of the components included in Table 3.1.2-16, as specified in the column “action” of Table 3.1.2-16 in accordance with the definition in 3.1.1.

3.1.2.3.5.4.2.2 These components which are classified as “G” shall be generated in compliance with the provisions of Annex 10, Volume II or with the clause referred to in the column “mapping” of Table 3.1.2-16.

3.1.2.3.5.4.2.3 These components which are classified as “T” shall be converted from the AMHS parameter specified in the column “converted from AMHS parameter” of Table 3.1.2-16 and according to the specification in the clause referred to in the column “mapping”.

Table 3.1.2-16. Generation of unknown address AFTN service message

AFTN Message Part	Component	Action	converted from AMHS parameter	Mapping
Heading	Start-of-Heading Character	G	-	see Annex 10, Vol. II, 4.4.15.1.1
	Transmission Identification	G	-	see Annex 10, Vol. II, 4.4.15.1.1
Address	Alignment Function	G	-	see Annex 10, Vol. II, 4.4.15.2.1
	Priority Indicator	G	-	see 3.1.2.3.5.4.2.4
	Addressee Indicator(s)	G	-	see 3.1.2.3.5.4.2.5
	Alignment Function	G	-	see Annex 10, Vol. II, 4.4.15.2.1
Origin	Filing Time	G	-	see 3.1.2.3.5.4.2.6
	Originator Indicator	G	-	see 3.1.2.3.5.4.2.7
	Priority Alarm	G	-	see Annex 10, Vol. II, 4.4.15.2.2
	Optional Heading Information	X	-	-
	Alignment Function	G	-	see Annex 10, Vol. II, 4.4.15.2.2
	Start-of-Text Character	G	-	see Annex 10, Vol. II, 4.4.15.2.2
Text		T	actual-recipient-name (see Table 3.1.2-17/ Part 1/2.2.1)	see 3.1.2.3.5.4.2.8
Ending	Alignment Function	G	-	see Annex 10, Vol. II, 4.4.15.3.12
	Page-feed sequence	G	-	see Annex 10, Vol. II, 4.4.15.3.12
	End-of-Text Character	G	-	see Annex 10, Vol. II, 4.4.15.3.12

Legend: (see 3.1.1)

G = generated

T = translated

X = excluded (not used)

3.1.2.3.5.4.2.4 The priority indicator component shall take the value of the priority indicator of the subject AFTN message.

3.1.2.3.5.4.2.5 The addressee indicator(s) component shall contain a single AF-Address which is the originator indicator of the subject AFTN message.

3.1.2.3.5.4.2.6 The filing time component, expressed as a date-time group in compliance with Annex 10, Volume II, 4.4.15.2.2.1, shall take the value of the time at which the AFTN service message is generated by the Message Transfer and Control Unit.

3.1.2.3.5.4.2.7 The originator indicator shall be the AFTN Address of the AFTN Component of the AFTN/AMHS Gateway, as specified in 3.1.2.3.2.1.16.

3.1.2.3.5.4.2.8 The value of the message text component shall be structured as follows:

- a) a first line composed as specified in Annex 10, Volume II, 4.4.11.13.3, items 1) to 4), using the origin of the subject AFTN message;
- b) a second line composed as specified in Annex 10, Volume II, 4.4.11.13.3, items 5) and 6), using the first address line of the subject AFTN message; and
- c) the third and following lines as appropriate composed as specified in Annex 10, Volume II, 4.4.11.13.3, items 7) to 9), using the AF-Address(es) translated as specified in 3.1.2.3.5.4.2.9 from the *actual-recipient-name* elements of the *per-recipient-fields* of the Non-Delivery Report which were not stored for action at the control position as the result of 3.1.2.3.5.4.1.2.

3.1.2.3.5.4.2.9 Each *actual-recipient-name* element used to generate an unknown address AFTN service message as specified in item c) of 3.1.2.3.5.4.2.8 above shall be processed for translation into an AF-Address in one of three mutually exclusive manners, after preliminary conversion of the value of all AMHS address attributes from lower case IA5IRV characters, if any, to upper case IA5IRV characters:

- a) allocation of the value of the first element of the *organizational-unit-names* attribute to the AF-Address, if this value is a syntactically valid AF-Address and if the *organization-name* attribute has the value “AFTN”;
- b) determination of an AF-Address matching exactly the MF-Address of the recipient in the User address look-up table maintained in the Message Transfer and Control Unit, if the value of the *organization-name* attribute differs from “AFTN” and if such an exact match can be found; or
- c) if none of the conditions in a) and b) can be met, then:
 - 1) logging of the error situation and reporting to a control position; and
 - 2) storage of the MF-Address and of the non-delivery report for appropriate action at the control position.

3.1.2.3.5.4.3 Use of Report Transfer Envelope and Content parameters

3.1.2.3.5.4.3.1 Each of the elements composing the Report Transfer Envelope and Report Transfer Content of an AMHS report to be converted into an AFTN service message in the Message Transfer and Control Unit shall be processed as specified in the column “action” of Table 3.1.2-17.

3.1.2.3.5.4.3.2 These elements shall be handled according to the specification in the clause referred to in the column “mapping” of Table 3.1.2-17.

Note.— Table 3.1.2-17 is structured as a PRL derived from the ISPICS Proforma included in ISO/IEC ISP 10611-3. The columns “Base” and “ISP” are extracted from ISO/IEC ISP 10611-3, and the column “Basic ATS Message Service” specifies the static capability of an AU for the MT-EoS, i.e. the ability to convey, handle and act in relation with the element. The references to the ISP Profile are indicated in the part titles as AMH11/ref where appropriate.

Table 3.1.2-17. Use of Report Transfer Envelope and Content parameters

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Mapping / Notes
PART 1 : AMH11/A.1.4.3 REPORTTRANSFER						
1	ReportTransferEnvelope	M	M	M	D	-
2	ReportTransferContent	M	M	M	T	see Part 1/2.1 and 2.2
2.1	(per report fields)					
2.1.1	subject-identifier	M	M	M	D	-
2.1.2	subject-intermediate-trace-information	O	M	M	D	-
2.1.3	original-encoded-information-types	M	M	M	D	-
2.1.4	content-type	M	M	M	D	-
2.1.5	content-identifier	M	M	M	D	-
2.1.6	returned-content	O	M-	M-	D	-
2.1.7	additional-information	O	M-	M-	D	-
2.1.8	extensions	M	M	M	D	-
2.2	per-recipient-fields	M	M	M		
2.2.1	actual-recipient-name	M	M	M	T	see 3.1.2.3.5.4.2.8
2.2.2	originally-specified-recipient-number	M	M	M	D	-

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Mapping / Notes
2.2.3	per-recipient-indicators	M	M	M	D	-
2.2.4	last-trace-information	M	M	M	D	-
2.2.5	originally-intended-recipient-name	M	M	M	X	see 3.1.2.3.5.4.1.3
2.2.6	supplementary-information	O	M-	M-	D	-
2.2.7	extensions	M	M	M	D	-

Legend (see 3.1.1) :

- M = mandatory support
- M- = minimal mandatory support
- O = optional support
- D = discarded
- T = translated
- X = excluded

3.1.2.3.5.5 Action upon reception of AMHS Probe

3.1.2.3.5.5.1 Upon reception by the Message Transfer and Control Unit of an AMHS probe which content type is either “interpersonal-messaging-1984” or “interpersonal-messaging-1988”, the received probe shall be processed in one of the following manners, depending on the abstract-value of the current-encoded-information-types, determined as either the abstract-value of the latest *converted-encoded-information-types*, if existing, in the *trace-information* element, or as the abstract-value of the *original-encoded-information-types* element in the Probe Transfer Envelope if the previous does not exist:

- a) processing as specified in 3.1.2.3.5.5.2 if the abstract-value of the current encoded-information-types is “ia5-text” or extended “ia5-text”; or
- b) if the abstract-value differs from built-in “ia5-text” and from extended “ia5-text”:
 - 1) rejection of the probe for all the probe recipients; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in all the *per-recipient-fields* of the report:
 - i) “unable-to-transfer” for the *non-delivery-reason-code*; and
 - ii) “encoded-information-types-unsupported” for the *non-delivery-diagnostic-code*.

3.1.2.3.5.5.2 A probe which was not rejected as the result of 3.1.2.3.5.5.1 shall be processed in one of the following manners:

- a) processing as specified in 3.1.2.3.5.5.3 if the abstract-value of the *implicit-conversion-prohibited* in the *per-message-indicators* element in the Probe Transfer Envelope differs from “prohibited”; or
- b) if the abstract-value of the element is “prohibited”:
 - 1) rejection of the message for all the message recipients; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in all the *per-recipient-fields* of the report:
 - i) “conversion-not-performed” for the *non-delivery-reason-code*;
 - ii) “implicit-conversion-prohibited” for the *non-delivery-diagnostic-code*; and
 - iii) “unable to convert to AFTN” for the *supplementary-information*.

3.1.2.3.5.5.3 A probe which was not rejected as the result of 3.1.2.3.5.5.2 shall be processed in one of three mutually exclusive manners:

- a) if, due to system resource limitation, the value of the element *content-length* in the Probe Transfer Envelope exceeds the conversion capability of the Message Transfer and Control Unit, then:
 - 1) rejection of the message for all the message recipients; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in all the *per-recipient-fields* of the report:
 - i) “unable-to-transfer” for the *non-delivery-reason-code*; and
 - ii) “content-too-long” for the *non-delivery-diagnostic-code*; or
- b) processing as specified in 3.1.2.3.5.5.4 for further conveyance test if the *content-length* does not exceed the conversion capability of the Message Transfer and Control Unit.

Note.— The way to determine the conversion capability of the Message Transfer and Control Unit in terms of message length is a matter local to the AMHS Management Domain operating the AFTN/AMHS Gateway.

3.1.2.3.5.5.4 A probe which was not rejected as the result of 3.1.2.3.5.5.3 shall be processed in one of three mutually exclusive manners, depending on the number of probe recipients towards which the Message Transfer and Control Unit is responsible for conveyance test, and on the AFTN/AMHS Gateway resources:

- a) if this number exceeds 21 probe recipients:
 - 1) attempt to split the probe, internally to the Message Transfer and Control Unit, into several probes, each of them with no more than 21 probe recipients:
 - i) each of the resulting probes having for conveyance test purposes the same *per-probe-fields* in the Probe Transfer Envelope; and
 - ii) only the *per-recipient-fields* elements in the Probe Transfer Envelope varying between the different resulting probes; and
 - 2) processing of each of these probes as specified in 3.1.2.3.5.5.5;
- b) if this number exceeds 21 probe recipients, and if, due to system resource limitation, the splitting attempt made by the gateway as specified in item a) above cannot be properly achieved:
 - 1) rejection of the probe for all the probe recipients; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in all the *per-recipient-fields* of the report:
 - i) “unable-to-transfer” for the *non-delivery-reason-code*;
 - ii) “too-many-recipients” for the *non-delivery-diagnostic-code*; and
 - iii) “unable to convert to AFTN due to number of recipients” for the *supplementary-information*; or
- d) processing as specified in 3.1.2.3.5.5.5, if this number does not exceed 21 probe recipients.

3.1.2.3.5.5.5 A probe which was not rejected as the result of 3.1.2.3.5.5.4 shall be processed in one of the following manners, depending on the ability of the Message Transfer and Control Unit to translate the originator-name element of the Probe Transfer Envelope into an AF-Address:

- a) processing as specified in 3.1.2.3.5.5.6 if either of the following conditions is met:
 - 1) if, after conversion from lower case IA5IRV characters, if any, to upper case IA5IRV characters, the *organization-name* attribute has the value “AFTN” and if the value of the first element of the *organizational-unit-names* is a syntactically valid AF-Address; or

- 2) if, after conversion from lower case IA5IRV characters, if any, to upper case IA5IRV characters, the value of the *organization-name* attribute differs from “AFTN” and if an AF-Address matching exactly the MF-Address of the originator can be found in the User address look-up table maintained in the Message Transfer and Control Unit; or
- b) if none of the conditions 1) or 2) in a) above can be met, then:
 - 1) rejection of the probe for all the probe recipients; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in all the *per-recipient-fields* of the report:
 - i) “unable-to-transfer” for the *non-delivery-reason-code*;
 - ii) “invalid-arguments” for the *non-delivery-diagnostic-code*; and
 - iii) “unable to convert to AFTN due to unrecognized originator O/R address” for the *supplementary-information*.

3.1.2.3.5.5.6 For each probe recipient, a probe which was not rejected as the result of 3.1.2.3.5.5.5 shall be processed in one of the following manners, depending on the ability of the Message Transfer and Control Unit to translate the considered *recipient-name* element of the Probe Transfer Envelope into an AF-Address:

- a) processing as specified in 3.1.2.3.5.5.7 if either of the following conditions is met:
 - 1) if, after conversion from lower case IA5IRV characters, if any, to upper case IA5IRV characters, the *organization-name* attribute has the value “AFTN” and if the value of the first element of the *organizational-unit-names* is a syntactically valid AF-Address; or
 - 2) if, after conversion from lower case IA5IRV characters, if any, to upper case IA5IRV characters, the value of the *organization-name* attribute differs from “AFTN” and if an AF-Address matching exactly the MF-Address of the recipient can be found in the User address look-up table maintained in the Message Transfer and Control Unit; or
- b) if none of the conditions 1) or 2) in a) above can be met, then:
 - 1) rejection of the probe for the considered recipient; and
 - 2) generation of a non-delivery report as specified in 3.1.2.3.5.6 with the following elements taking the following abstract-values in the corresponding *per-recipient-fields* of the report:
 - i) “unable-to-transfer” for the *non-delivery-reason-code*; and
 - ii) “unrecognised-OR-name” for the *non-delivery-diagnostic-code*.

3.1.2.3.5.5.7 For the probe recipients which were not rejected as the result of 3.1.2.3.5.5.6, a delivery-report shall be generated as specified in 3.1.2.3.5.6, if requested, to indicate the successful result of the probe conveyance test.

3.1.2.3.5.6 Generation of AMHS Reports

3.1.2.3.5.6.1 General

3.1.2.3.5.6.1.1 A non-delivery report shall be generated by the Message Transfer and Control Unit:

- a) for each message or probe which was rejected at the AFTN/AMHS Gateway, as the result of the procedures described in 3.1.2.3.5.1.1, 3.1.2.3.5.1.4, 3.1.2.3.5.2 and 3.1.2.3.5.5, either for all the recipients or for certain recipients; and
- b) as the result of the conversion of an unknown address AFTN service message, as specified in 3.1.2.3.4.4.1.6.

3.1.2.3.5.6.1.2 **Recommendation.**— *When the generation of a non-delivery report is required in relation with the rejection at the AFTN/AMHS Gateway of the subject AMHS message for more than one recipient of the subject AMHS message, a single non-delivery report should be generated to report on the rejection for multiple recipients, using several per-recipient-fields elements in the Report Transfer Content.*

3.1.2.3.5.6.1.3 For each AMHS message which was converted by the Message Transfer and Control Unit as the result of the procedures specified in 3.1.2.3.5.2.2 to 3.1.2.3.5.2.4 and then successfully passed to the AFTN Component as specified in 3.1.2.3.5.1.6, a delivery report shall be generated by the Message Transfer and Control Unit for each message recipient of which:

- a) the *originating-MTA-report-request* element has the abstract-value “report” or “audited-report”; or
- b) the *originator-report-request* element has the abstract-value “report”; or
- c) both conditions a) and b) above are met.

3.1.2.3.5.6.1.4 **Recommendation.**— *When the generation of a delivery report is required as specified in 3.1.2.3.5.6.1.3 for more than one recipient of the subject AMHS message, a single delivery report should be generated to report on the conveyance towards multiple recipients, using several per-recipient-fields elements in the Report Transfer Content.*

3.1.2.3.5.6.1.5 When the generation of a delivery report is required in relation with the result of a probe conveyance test as specified in 3.1.2.3.5.5, the clauses 3.1.2.3.5.6.1.3 to 3.1.2.3.5.6.1.4 above shall apply with the difference that the event which triggers the generation of the delivery report is the success of the probe conveyance test.

3.1.2.3.5.6.1.6 A report resulting from the clauses above shall be generated as specified in 3.1.2.3.5.6.2.

3.1.2.3.5.6.2 Generation of Report Transfer Envelope and Content

3.1.2.3.5.6.2.1 Each report resulting from the specification of 3.1.2.3.5.6.1 shall be generated by the Message Transfer and Control Unit, in the form of an AMHS Report Transfer Envelope and Report Transfer Content, composed of elements as specified in the column “action” of Table 3.1.2-18.

3.1.2.3.5.6.2.2 These elements which are classified as “G” or “G2” shall be either generated or conditionally generated according to the specification in the clause referred to in the column “generation action” of Table 3.1.2-18.

Note.— Table 3.1.2-18 is structured as a PRL derived from the ISPICS Proforma included in ISO/IEC ISP 10611-3. The columns “Base” and “ISP” are extracted from ISO/IEC ISP 10611-3, and the column “Basic ATS Message Service” specifies the static capability of an AU in relation with the MT-EoS, i.e. the ability to convey, handle and act in relation with the element. The references to the ISP Profile are indicated in the part titles as AMH11/ref where appropriate.

Table 3.1.2-18. Generation of AMHS Report

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Generation action
PART 1 : AMH11/A.1.4.3 REPORT/TRANSFER						
1	ReportTransferEnvelope	M	M	M	G	see Part 1/1.1-1.4
1.1	report-identifier	M	M	M	G	see 3.1.2.3.5.6.2.3 and Part 2/1
1.2	report-destination-name	M	M	M	G	see 3.1.2.3.5.6.2.6
1.3	trace-information	M	M	M	G	see 3.1.2.3.5.6.2.7
1.4	extensions	M	M	M		see 3.1.2.3.5.6.2.8
1.4.1	message-security-label	O	M-	M-	X	-
1.4.2	originator-and-DL-expansion-history	M	M	M	G2	see 3.1.2.3.5.6.2.9
1.4.3	reporting-DL-name	O	M-	M-	X	-
1.4.4	reporting-MTA-certificate	O	M-	M-	X	-
1.4.5	report-origin-authentication-check	O	M-	M-	X	-
1.4.6	internal-trace-information	M	M	M	G	see 3.1.2.3.5.6.2.10

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Generation action
2	ReportTransferContent	M	M	M	G	see Part1/2.1 and 2.2
2.1	(per report fields)					
2.1.1	subject-identifier	M	M	M	G	see 3.1.2.3.5.6.2.11
2.1.2	subject-intermediate-trace- information	O	M	M	G2	see 3.1.2.3.5.6.2.12
2.1.3	original-encoded-information -types	M	M	M	G	see 3.1.2.3.5.6.2.13
2.1.4	content-type	M	M	M	G	see 3.1.2.3.5.6.2.14
2.1.5	content-identifier	M	M	M	G2	see 3.1.2.3.5.6.2.15
2.1.6	returned-content	O	M-	M-	G2	see 3.1.2.3.5.6.2.16
2.1.7	additional-information	O	M-	M-	X	-
2.1.8	extensions	M	M	M		see 3.1.2.3.5.6.2.8
2.1.8.1	content-correlator	M	M	M	G2	see 3.1.2.3.5.6.2.17
2.2	per-recipient-fields	M	M	M		see Part1/2.2.1-2.2.7
2.2.1	actual-recipient-name	M	M	M	G	see 3.1.2.3.5.6.2.18
2.2.2	originally-specified-recipient- number	M	M	M	G	see 3.1.2.3.5.6.2.19
2.2.3	per-recipient-indicators	M	M	M	G	see 3.1.2.3.5.6.2.20
2.2.4	last-trace-information	M	M	M	G	see Part 2/7
2.2.5	originally-intended-recipient- name	M	M	M	G2	see 3.1.2.3.5.6.2.26
2.2.6	supplementary-information	O	M-	M-	G2	see 3.1.2.3.5.6.2.27
2.2.7	extensions	M	M	M		see 3.1.2.3.5.6.2.8
2.2.7.1	redirection-history	M	M	M	G2	see 3.1.2.3.5.6.2.28

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Generation action
2.2.7.2	physical-forwarding-address	O	M-	M-	X	-
2.2.7.3	recipient-certificate	O	M-	M-	X	-
2.2.7.4	proof-of-delivery	O	M-	M-	X	-
PART 2 : AMH11/A.1.5 COMMON DATA TYPES						
1	MTSIdentifier					
1.1	global-domain-identifier	M	M	M	G	see 3.1.2.3.5.6.2.4 and Part 2/2
1.2	local-identifier	M	M	M	G	see 3.1.2.3.5.6.2.5
2	GlobalDomainIdentifier					
2.1	country-name	M	M	M		see 3.1.2.3.5.6.2.29
2.2	administration-domain-name	M	M	M		see 3.1.2.3.5.6.2.30
2.3	private-domain-identifier	M	M	M		see 3.1.2.3.5.6.2.31
6	TraceInformation					
6.1	TraceInformationElement	M	M	M	G	see Part 2/6.1.1 and 6.1.2
6.1.1	global-domain-identifier	M	M	M	G	see 3.1.2.3.5.6.2.32 and Part 2/2
6.1.2	domain-supplied-information	M	M	M	G	see Part 2/6.1.2.1-6.1.2.4
6.1.2.1	arrival-time	M	M	M	G	see 3.1.2.3.5.6.2.33
6.1.2.2	routing-action	M	M	M	G	see Part 2/6.1.2.2.1 and 6.1.2.2.2
6.1.2.2.1	relayed	M	M	M	G	see 3.1.2.3.5.6.2.34
6.1.2.2.2	rerouted	O	C1	C1	X	-

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Generation action
6.1.2.3	attempted-domain	O	C1	C1	X	-
6.1.2.4	(additional actions)					
6.1.2.4.1	deferred-time	M	C2	C2	X	-
6.1.2.4.2	converted-encoded- information-types	O	M-	M-	X	-
6.1.2.4.3	other-actions	O	M-	M-	X	-
6.1.2.4.3.1	redirected	O	M-	M-	X	-
6.1.2.4.3.2	dl-operation	O	M-	M-	X	-
7	LastTraceInformation					
7.1	arrival-time	M	M	M	G	see 3.1.2.3.5.6.2.21
7.2	converted-encoded- information-types	M	M	M	G2	see 3.1.2.3.5.6.2.22
7.3	report-type	M	M	M	G	see Part 2/7.3.1 and 7.3.2
7.3.1	delivery	M	M	M	G2	see Part 2/7.3.1.1 and 7.3.1.2
7.3.1.1	message-delivery-time	M	M	M	G	see 3.1.2.3.5.6.2.23
7.3.1.2	type-of-MTS-user	M	M	M	G	see 3.1.2.3.5.6.2.24
7.3.2	non-delivery	M	M	M	G2	see Part 2/7.3.2.1 and 7.3.2.2
7.3.2.1	non-delivery-reason-code	M	M	M	G	see 3.1.2.3.5.6.2.25
7.3.2.2	non-delivery-diagnostic-code	M	M	M	G	see 3.1.2.3.5.6.2.25

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Generation action
PART 3 : AMH11/A.1.6 EXTENSION DATA TYPES						
1	ExtensionField					
1.1	type	M	M	M	G	see Part 3/1.1.1 and 1.1.2
1.1.1	standard-extension	M	M	M	G	see 3.1.2.3.5.6.2.8
1.1.2	private-extension	O	M-	M-	X	-
1.2	criticality	M	M	M	G	see 3.1.2.3.5.6.2.8
1.3	value	M	M	M	G	see 3.1.2.3.5.6.2.8
5	InternalTraceInformation					
5.1	global-domain-identifier	M	M	M	G	see 3.1.2.3.5.6.2.32
5.2	mta-name	M	M	M	G	see 3.1.2.3.5.6.2.35
5.3	mta-supplied-information	M	M	M	G	see Part 3/5.3.1-5.3.4
5.3.1	arrival-time	M	M	M	G	see 3.1.2.3.5.6.2.33
5.3.2	routing-action	M	M	M	G	see Part 3/5.3.2.1-5.3.2.2
5.3.2.1	relayed	M	M	M	G	see 3.1.2.3.5.6.2.34
5.3.2.2	rerouted	O	C1	C1	X	-
5.3.3	attempted	O	C1	C1	X	-
5.3.4	(additional actions)					
5.3.4.1	deferred-time	M	C2	C2	X	-
5.3.4.2	converted-encoded-information-types	O	M-	M-	X	-
5.3.4.3	other-actions	O	M-	M-	X	-

Ref	Element	Base	ISP	Basic ATS Mess. Service	Action	Generation action
5.3.4.3.1	redirected	O	M-	M-	X	-
5.3.4.3.2	dl-operation	O	M-	M-	X	-

Legend (see 3.1.1) :

- M = mandatory support
- M- = minimal mandatory support
- O = optional support
- I = out of scope
- = not applicable
- C1 = if rerouting is supported then M else M-
- C2 = if deferred delivery is supported then M else M-
- G = generated
- G2 = conditionally generated
- X = excluded (not used)

3.1.2.3.5.6.2.3 The element *report-identifier* in the Report Transfer Envelope shall:

- a) be generated locally so as to ensure that it distinguishes the report from all other messages, probes or reports generated in the AMHS, as specified in ISO/IEC 10021-4, 12.2.1.3.1.1; and
- b) be composed as specified in Table 3.1.2-18/Part 2/1.

3.1.2.3.5.6.2.4 The element *global-domain-identifier* in the *report-identifier*, or in the *trace-information*, or in the *internal-trace-information* shall:

- a) identify the AMHS Management Domain operating the AFTN/AMHS Gateway; and
- b) be composed as specified in Table 3.1.2-18/Part 2/2.

3.1.2.3.5.6.2.5 The element *local-identifier* in the *report-identifier* shall be generated locally so as to ensure that it distinguishes the report from all other messages, probes or reports generated in the AMHS Management Domain operating the AFTN/AMHS Gateway.

3.1.2.3.5.6.2.6 The *report-destination-name* element in the Report Transfer Envelope shall be one of the following:

- a) the last OR-name in the *DL-expansion-history* element, if present, of the subject AMHS message as specified in Table 3.1.2-12/Part 1/1.1.11.11; or
- b) the *originator-name* of the subject AMHS message, as specified in Table 3.1.2-12/Part 1/1.1.2, if there is no *DL-expansion-history* element in the subject AMHS message.

3.1.2.3.5.6.2.7 The first *trace-information-element* in the *trace-information* of the Report Transfer Envelope shall be generated as specified in Table 3.1.2-18/Part 2/6.

3.1.2.3.5.6.2.8 Only extensions of type “standard-extension” as defined in the base standards shall be used, as further specified in the classification of Table 3.1.2-18.

3.1.2.3.5.6.2.9 If a *DL-expansion-history* element as specified in Table 3.1.2-12/Part 1/1.1.11.11 was present in the subject AMHS message, the *originator-and-DL-expansion-history* element shall be generated as the sequence of the *originator-name* of the subject AMHS message, as specified in Table 3.1.2-12/Part 1/1.1.2, and of the aforementioned *DL-expansion-history* element of the subject AMHS message.

3.1.2.3.5.6.2.10 The first *internal-trace-information-element* in the *internal-trace-information* of the Report Transfer Envelope shall be generated as specified in Table 3.1.2-18/Part 3/5.

3.1.2.3.5.6.2.11 The *subject-identifier* element in the Report Transfer Content shall take the value of the *message-identifier* element of the subject AMHS message as specified in Table 3.1.2-12/Part 1/1.1.1.

3.1.2.3.5.6.2.12 The *subject-intermediate-trace-information* element in the Report Transfer Content shall take the value which the *trace-information* element of the subject AMHS message as specified in Table 3.1.2-12/Part 1/1.1.10 had when the subject AMHS message entered the AMHS Management Domain operating the Message Transfer and Control Unit, if and only if the *originating-MTA-report-request* element in the *per-recipient-indicators* of all the subject AMHS message recipients in the subject Message Transfer Envelope has the abstract-value “audited-report”.

3.1.2.3.5.6.2.13 The *original-encoded-information-types* element in the Report Transfer Content shall take the value of the *original-encoded-information-types* element of the subject AMHS message as specified in Table 3.1.2-12/Part 1/1.1.3.

3.1.2.3.5.6.2.14 The *content-type* element in the Report Transfer Content shall take the value of the *content-type* element of the subject AMHS message as specified in Table 3.1.2-12/Part 1/1.1.4.

3.1.2.3.5.6.2.15 The *content-identifier* element in the Report Transfer Content shall either:

- a) take the value of the *content-identifier* element of the subject AMHS message as specified in Table 3.1.2-12/Part 1/1.1.5, if present; or
- b) be omitted in the report if there is no such element in the subject AMHS message.

3.1.2.3.5.6.2.16 The *returned-content* element in the Report Transfer Content shall optionally take the value of the *content* of the subject AMHS message, if and only if the *content-return-request* element in the *per-message-indicators* of the subject AMHS message in the subject Message Transfer Envelope has the abstract-value “content-return-requested”.

Note.— The Message Transfer and Control Unit is not mandated to implement the Return Of Content (RoC) Optional Functional Group as defined in ISO/IEC ISP 10611-1.

3.1.2.3.5.6.2.17 The *content-correlator* element in the Report Transfer Content shall either:

- a) take the value of the *content-correlator* element of the subject AMHS message as specified in Table 3.1.2-12/Part 1/1.1.11.10, if present; or
- b) be omitted in the report if there is no such element in the subject AMHS message.

3.1.2.3.5.6.2.18 The *actual-recipient-name* element in a *per-recipient-fields* element of the Report Transfer Content shall take the value of the corresponding *recipient-name* element in the *per-recipient-fields* of the subject AMHS message as specified in Table 3.1.2-12/Part 1/1.2.1.

3.1.2.3.5.6.2.19 The *originally-specified-recipient-number* element in a *per-recipient-fields* element of the Report Transfer Content shall take the value of the corresponding *originally-specified-recipient-number* element in the *per-recipient-fields* of the subject AMHS message as specified in Table 3.1.2-12/Part 1/1.2.2.

3.1.2.3.5.6.2.20 The *per-recipient-indicators* element in a *per-recipient-fields* element of the Report Transfer Content shall take the value of the corresponding *per-recipient-indicators* element in the *per-recipient-fields* of the subject AMHS message as specified in Table 3.1.2-12/Part 1/1.2.3.

3.1.2.3.5.6.2.21 The *arrival-time* element in the *last-trace-information* of a *per-recipient-fields* element shall take the value of the time at which the subject AMHS message entered the AMHS Management Domain operating the AFTN/AMHS Gateway, as found in the last *trace-information-element* of the subject AMHS message, as specified in Table 3.1.2-12/Part 2/6.1.2.1.

3.1.2.3.5.6.2.22 The *converted-encoded-information-types* element in the *last-trace-information* of a *per-recipient-fields* element shall either:

- a) take the last value of the *converted-encoded-information-types* element in the *trace-information* of the subject AMHS message, as specified in Table 3.1.2-12/Part 2/6.1.2.4.2, if this element exists; or
- b) be omitted in the report, if no such element is present in the *trace-information* of the subject AMHS message.

3.1.2.3.5.6.2.23 If the report is a delivery-report, the *message-delivery-time* element in the *last-trace-information* of a *per-recipient-fields* element shall be the time at which the subject AMHS message has been successfully passed to the AFTN Component by the Message Transfer and Control Unit.

3.1.2.3.5.6.2.24 If the report is a delivery-report, the *type-of-MTS-user* element in the *last-trace-information* of a *per-recipient-fields* element shall take the abstract-value “other”.

3.1.2.3.5.6.2.25 If the report is a non-delivery-report, the *non-delivery-reason-code* and *non-delivery-diagnostic-code* elements in the *last-trace-information* of a *per-recipient-fields* element shall take the abstract-values specified in the clause which caused the generation of the report.

3.1.2.3.5.6.2.26 The *originally-intended-recipient-name* element in a *per-recipient-fields* element shall either:

- a) take the value of the first O/R name found in the *redirection-history* element of the subject AMHS message, if present, as specified in Table 3.1.2-12/Part 1/1.2.5.13; or
- b) be omitted in the report if there is no *redirection-history* element in the subject AMHS message.

3.1.2.3.5.6.2.27 The *supplementary-information* element in a *per-recipient-fields* element shall take one of the following values:

- a) the value “This report only indicates successful (potential) conversion to AFTN, not delivery to a recipient” if the report is a delivery-report; or
- b) the value, if any, specified in the clause which caused the generation of the report if it is a non-delivery-report.

3.1.2.3.5.6.2.28 The *redirection-history* element in a *per-recipient-fields* element shall either:

- a) take the value of the *redirection-history* element of the subject AMHS message, if present, as specified in Table 3.1.2-12/Part 1/1.2.5.13; or
- b) be omitted in the report if there is no *redirection-history* element in the subject AMHS message.

3.1.2.3.5.6.2.29 The element *country-name* in the *global-domain-identifier* element of the *MTS-identifier* and of the first *trace-information-element* shall:

- a) be part of the identification of the AMHS Management Domain operating the AFTN/AMHS Gateway by taking one of the following values:
 - 1) the two-character alphabetical country-indicator as specified in ISO 3166 for the country, or for one of the countries, where the AMHS Management Domain has been registered, if the AMHS Management Domain has been subject to national or multi-national registration; or
 - 2) a two-character alphabetical indicator dedicated to an international organization, if the AMHS Management Domain has been subject to international registration as defined in ITU-T Recommendation X.666; and
- b) be encoded as a Printable String.

3.1.2.3.5.6.2.30 The element *administration-domain-name* in the *global-domain-identifier* element of the *MTS-identifier* and of the first *trace-information-element* shall:

- a) be part of the identification of the AMHS Management Domain operating the AFTN/AMHS Gateway by taking one of the following values, depending on its status:
 - 1) the name of the ADMD under which the AMHS Management Domain has been registered, either nationally or internationally, if the AMHS Management Domain operates as an ADMD;
 - 2) the name of the ADMD to which the AMHS Management Domain is connected, if the AMHS Management Domain operates as a PRMD; or
 - 3) the value single-space if the AMHS Management Domain operates as a PRMD and is unique with regard to the *country-name* identifying the area where it is registered, either nationally or internationally; and
- b) be encoded as a Printable String.

3.1.2.3.5.6.2.31 The element *private-domain-identifier* in the *global-domain-identifier* element of the *MTS-identifier* and of the first *trace-information-element* shall be handled in one of the following manners, depending on the status under which the AMHS Management Domain operates:

- a) generation of the element, with the value of the name of the PRMD, encoded as a Printable String, if the AMHS Management Domain operates as an PRMD; or
- b) omission in the *global-domain-identifier* if the AMHS Management Domain operates as an ADMD.

3.1.2.3.5.6.2.32 The element *global-domain-identifier* in the *trace-information* or in the *internal-trace-information* shall:

- a) identify the AMHS Management Domain operating the AFTN/AMHS Gateway; and
- b) be composed as specified in Table 3.1.2-18 / Part 2/2.

3.1.2.3.5.6.2.33 The element *arrival-time* in the first element of *trace-information* or of *internal-trace-information* shall take the semantic value of the time when the report was generated by the Message Transfer and Control Unit for conveyance in the AMHS.

3.1.2.3.5.6.2.34 The element *routing-action* in the first element of *trace-information* or of *internal-trace-information* shall take the abstract-value “relayed”.

3.1.2.3.5.6.2.35 The element *mta-name* in the first element of *internal-trace-information* shall be the *mta-name* assigned to the Message Transfer and Control Unit included in the AFTN/AMHS Gateway.

Note.— The structure of the mta-name of the Message Transfer and Control Unit included in an AFTN/AMHS Gateway within an AMHS Management Domain is a matter of policy internal to the AMHS Management Domain.

3.1.3 ATN PASS-THROUGH SERVICE

3.1.3.1 System level provisions

3.1.3.1.1 The ATN Pass-Through Service shall provide a message environment for the exchange of IA-5 encoded AFTN messages over the ATN Internet Communications Service and with the AFTN via the AFTN/ATN Type A gateway.

Note 1.— This service does not provide classical store and forward messages services such as found in the AFTN and the ATS Message Service, nor is it visible to users at AFTN stations.

Note 2.— As a matter of organisations' policy, the implementation of the ATS Message Service may be deferred. In order to take early advantage of the enhanced connectivity provided by the ATN, ATS Organisations with such a policy may implement and operate in the interim the ATN Pass-Through Service. This service provides connectivity for the AFTN traffic as presently defined in Annex 10, Volume II, through the ATN. The interoperability between the ATS Message Service and the ATN Pass-Through Service is a local implementation matter.

3.1.3.1.2 **Recommendation.**— *ATS Organisations which choose to implement the ATN Pass-Through Service should plan to implement the ATS Message Service at the earliest possible time.*

3.1.3.1.3 **Recommendation.**— *ATS Organisations which choose to implement the ATN Pass-Through Service should provide the interoperability facilities to the ATS Message Service implementations.*

3.1.3.1.4 AFTN/ATN Type A Gateway users

The AFTN/ATN Type A Gateway users shall consist of AFTN stations (as defined in Annex 10, Volume II) exchanging AFTN messages.

3.1.3.1.5 AFTN/ATN Type A Gateway model

If an AFTN/ATN Type A Gateway is connected to an AFTN Centre which is capable of using only ITA-2 format, the AFTN Component of the gateway shall convert messages to/from the IA-5 format.

Note.— An ATS organisation may choose to connect an AFTN/ATN Type A Gateway to the AFTN only via its AFTN Centre. In this case, some requirements placed on the AFTN Component may not have to be fulfilled, provided that the AFTN Centre and AFTN/ATN Type A Gateway together fulfill all requirements.

3.1.3.1.5.1 AFTN/ATN Type A Gateway information model

The AFTN/ATN Type A Gateway information elements shall consist only of AFTN messages.

3.1.3.1.5.2 Security and management models

Recommendation.— *Security should be obtained by procedural means rather than by technical features inherent in the ATN Pass-Through Service.*

Note 1.— The security at each AFTN/ATN Type A Gateway is deemed a local issue to be addressed by the ATS organisation in charge of the system.

Note 2.— Management is limited to the logging provisions which are defined for the AFTN/ATN Type A Gateway. No provision is made for retrieval or exchange of this information, which is deemed a local issue to be addressed by the authority in charge of the system.

3.1.3.1.6 AFTN/ATN Type A Gateway System configurations

3.1.3.1.6.1 The minimal set of systems implemented and operated by an ATS organisation for the ATN Pass-Through Service shall be one AFTN/ATN Type A Gateway system.

3.1.3.1.6.2 The minimal set of communications circuits implemented by an ATS organisation operating an AFTN/ATN Type A Gateway shall be:

- a) when integrated with an AFTN centre, access to one ATN subnetwork;
- b) when not integrated with an AFTN centre, one AFTN circuit utilizing a code and byte independent procedure and access to one ATN subnetwork; or
- c) when not integrated with an AFTN centre, one AFTN circuit utilizing any Annex 10, Volume I controlled or Volume II uncontrolled circuit procedure and access to one ATN subnetwork.

Note.— The effect of selecting either 3.1.3.1.6.2 a) or b) is the elimination of the requirement for the AFTN/ATN Type A Gateway to implement the manual teletypewriter procedures, such as service message procedures, channel-check and transmission identification procedures, and code conversion procedures contained in Annex 10, Volume II.

3.1.3.1.7 AFTN/ATN Type A Gateway System naming principles

Naming for each AFTN/ATN Type A Gateway system shall consist of an AP-title set and an AE-qualifier set, as specified in 4.3.2.

3.1.3.1.8 AFTN/ATN Type A Gateway System addressing principles

3.1.3.1.8.1 There shall be two address forms used in the AFTN/ATN Type A Gateway System:

- a) an AFTN address comprising an AFTN addressee indicator as specified in Annex 10, Volume II, 4.4.3.1.2 and 4.4.14.2; and
- b) an ATN end-system address comprising a facility designation, as specified in 4.

3.1.3.1.8.2 A facility designation shall be assigned to each AFTN/ATN Type A Gateway.

3.1.3.1.9 Routing principles

Routing of messages shall be provided by the AFTN Centres to which the AFTN/ATN Type A Gateway is connected.

3.1.3.1.10 Processing of communication failure

If, for any reason, the Message Transfer and Control Unit is unable to accept AFTN messages passed by the AFTN Component, then the AFTN Component shall handle this situation in compliance with the provisions of Annex 10, Volume II, 4.4.1.5.2.3.

Note.— Such a condition may be caused by the inability of the Message Transfer and Control Unit to pass messages to the ATN Component.

3.1.3.2 ATN Pass-Through Service Specification

3.1.3.2.1 An AFTN/ATN Type A Gateway shall consist of the following three logical components:

- a) AFTN component;
- b) ATN component; and
- c) Message transfer and control unit.

3.1.3.2.2 The three logical components shall interact according to the architecture specified in 4.

3.1.3.2.3 For both the configurations specified in 3.1.3.1.6.2 a) and b), the ATN Pass-Through Service shall be totally transparent for AFTN messages and AFTN service messages to the users of the service, except when applying the procedures for address stripping.

3.1.3.2.4 For the configuration specified in 3.1.3.1.6.2 c), the AFTN/ATN Type A Gateways shall handle the AFTN procedures as specified in Annex 10, Volume II.

3.1.3.3 AFTN/ATN Type A Gateway Specification

3.1.3.3.1 AFTN component

3.1.3.3.1.1 The AFTN component shall handle the interface to the AFTN and provide an interface to the Message Transfer and Control Unit.

3.1.3.3.1.2 The AFTN component shall implement:

- a) all the applicable requirements of Annex 10, Volume II, in a manner so as to be indistinguishable from an operational AFTN station by the AFTN Centre to which the gateway is connected; and
- b) additional requirements which are not placed on AFTN stations by Annex 10, Volume II but which are necessary due to the AFTN Component requirements pertaining to an AFTN/ATN Type A Gateway.

3.1.3.3.1.3 The AFTN component shall incorporate an AFTN procedure handler that provides all of the AFTN functions prescribed for the interface to the AFTN.

3.1.3.3.1.4 The AFTN Component shall isolate all AFTN procedures from the Message Transfer and Control Unit Component.

Note.— The AFTN procedure handler includes the procedures for managing the order of AFTN messages based on the transmission priority specified. Using the AFTN procedure handler for managing priority eliminates the need for the Message Transfer and Control Unit to manage message priorities.

3.1.3.3.1.5 The AFTN Component of an AFTN/ATN Type A Gateway shall perform short term retention of all messages transmitted towards the AFTN in a manner equivalent to that specified for an AFTN communication centre in Annex 10, Volume II, 4.4.1.7 to provide recovery from communication protocol errors.

3.1.3.3.1.6 The AFTN Component shall perform long-term retention of all AFTN messages, in their entirety, that it generates, for a period of at least thirty days.

3.1.3.3.1.7 The AFTN Component shall perform long-term retention of the heading, address and origin parts of all messages received from the Message Transfer and Control Unit and the action taken thereon, for a period of at least thirty days.

3.1.3.3.2 ATN component

3.1.3.3.2.1 The ATN Component shall implement the procedures required of an ATN End System as specified by the ATS Message protocol stack Type A.

3.1.3.3.2.2 ATN Component service

3.1.3.3.2.2.1 The ATN Component service shall consist of a single service primitive between it and the Message Transfer and Control Unit, the GA-Data request and indication.

Table 3.1.3-1. ATN Component Service

GA-Data Service Primitive	Req	Ind
User Data	M	M(=)
Called Address	M	
Calling Address	M	M(=)
Priority (transmission)	U	U(=)

3.1.3.3.2.2.2 The User Data parameter shall contain the IA5 form of a complete AFTN message, as defined in Annex 10, Volume II.

3.1.3.3.2.2.3 The Called Address parameter shall contain the ATN-end system id of the destination AFTN/ATN Type A Gateway consisting of the 8-character facility designation as defined in 4.

3.1.3.3.2.2.4 The Calling Address parameter shall contain the ATN-end system id of the AFTN/ATN Type A Gateway consisting of the 8-character facility designation as defined in 4.

3.1.3.3.2.2.5 The AFTN Priority parameter, if present, shall contain the AFTN priority indicator of the AFTN message, as defined in Annex 10, Volume II.

3.1.3.3.2.3 The ATS Message protocol stack Type A shall consist of protocols and procedures specified in 4; and consisting of:

- a) the ATN Component Control Function, which incorporates the Control Function of the Upper Layer Communication Service as specified in 4.3.3 and the additional provisions specified in 3.1.3.3.2.4;
- b) the Dialogue Service as specified in 4.2, consisting of:
 - 1) the Association Control Service Element,
 - 2) the Presentation Efficiency enhancements, and
 - 3) the Session Efficiency enhancements
- c) the Application Layer Naming and Context Definition as specified in 4.3.2; and
- d) the ATN Communication Services requirements as specified in 5.

3.1.3.3.2.4 ATN Component Control Function

Note.— The specification of the ATN Component Control Function (CF) does not constrain the implementation of the ATN Component as long as the latter exhibits the external behaviour of the CF as specified.

3.1.3.3.2.4.1 The ATN Component control function (CF) shall map the GA-Data requests and indications to and from the Dialogue Service as specified in 4.

3.1.3.3.2.4.2 Upon receipt of a GA-Data request, the CF shall determine if a dialogue exists with the destination ATN End-System by examining the Called Address parameter.

3.1.3.3.2.4.3 If a dialogue does not exist, the CF shall formulate a D-START-request primitive.

3.1.3.3.2.4.4 The parameters of the D-START-request shall be set according to Table 3.1.3-2.

Table 3.1.3-2. D-START-request/indication Parameters

D-START-request Parameter	GA-Data Request Parameter	Value
Called Peer ID	Called Address	atn-facility designation
Calling Peer ID	Calling Address	atn-facility designation
DS-User Version Number	-----	1
Security Requirements	-----	<not used>
QOS Parameters Routing class Priority Residual Error Rate	----- AFTN Priority -----	“ATSC: No Traffic Type Policy Preference” see Table 3.1.3-7 abstract-value “high”
User Data	-----	<not used>

3.1.3.3.2.4.5 Upon receipt of an D-START-indication, the CF shall determine if the parameters values are as indicated in Table 3.1.3-2.

3.1.3.3.2.4.6 If the parameters received in a D-START indication are acceptable and sufficient resources available to support the association, the CF shall accept the association by sending a D-START-response, in which the parameters are set according to Table 3.1.3-3 with the Result parameter set to the abstract-value “accepted”.

Note.— The actual definition of “acceptable values” for the parameters of the D-START-indication is a local matter.

3.1.3.3.2.4.7 If the parameters received in a D-START indication are unacceptable or there are insufficient resources available to support the association, the CF shall reject the association by sending an D-START-response, in which the parameters are set according to Table 3.1.3-3, with the Result parameter set to the abstract-value “rejected (permanent)” in the case of invalid parameters and set to the abstract-value “rejected (transient)” if there are insufficient resources.

Table 3.1.3-3. D-START-response/confirmation Parameters

D-START-response Parameter	GA - Data Indication Parameter	Value
DS-User Version	-----	1
Security Requirements	-----	<not used>
QOS (routing class, priority, and residual error rate)	-----	<not used>
User Data		<not used>
Result	-----	“accepted” “rejected (permanent)” “rejected (transient)”

Note 1.— The actual definition of “unacceptable values” for the parameters of the D-START-indication is a local matter.

Note 2.— The use of a security policy (such as only accepting associations from particular remote ATN end systems) to limit acceptance of associations is a local matter.

3.1.3.3.2.4.8 If the result parameter in the D-START-confirmation is set to the abstract-value “rejected (transient)”, the ATN Component CF shall:

- a) re-attempt (a locally defined number of times) the establishment of a dialogue with the same gateway; and
- b) if subsequent to the procedure defined in a) a dialogue still cannot be established, attempt the establishment of a dialogue with a prioritised list of gateways which are defined as being alternates to the one which has been determined as unreachable.

3.1.3.3.2.4.9 If the result parameter in the D-START-confirmation is set to the abstract-value “rejected (permanent)”, the ATN Component CF shall attempt (a locally defined number of times) the establishment of a dialogue with a prioritised list of gateways which are defined as being alternates to the one which has been determined as unreachable.

3.1.3.3.2.4.10 If any of the QOS parameters in the D-START-confirmation is unacceptable, the ATN Component CF shall perform the following:

- a) abort the dialogue as specified in 3.1.3.3.2.4.19; and

- b) attempt (a locally number of times) the establishment of a dialogue with a prioritised list of gateways which are defined as being alternates to the one which has been determined as unreachable with an acceptable QOS.

Note.— The actual definition of “unacceptable values” for the parameters of the D-START-confirmation is a local matter.

3.1.3.3.2.4.11 If subsequent to the procedures defined in 3.1.3.3.2.4.8 to 3.1.3.3.2.4.10 a dialogue still cannot be established, the ATN Component CF shall:

- a) log the error situation;
- b) store the message for further processing and
- c) ensure that the message is not discarded.

Note.— The processing to be performed in application of this clause is a local matter.

3.1.3.3.2.4.12 Upon the completion of the dialogue set-up, or in the case of using an existing dialogue, the CF shall formulate a D-DATA request by taking the data in the User Data parameter in the GA-Data-request and encoding it as the user data field in the D-DATA-request.

3.1.3.3.2.4.13 The data received in the User Data parameter of the GA-Data-request is the complete AFTN message, which shall be passed transparently to the destination system.

3.1.3.3.2.4.14 Upon the receipt of a D-DATA-indication, the CF shall extract the user data and place it in the User Data parameter of the GA-Data-indication.

3.1.3.3.2.4.15 If the CF does not have any data to send over a dialogue for a time period t_1 , it shall release the dialogue by formulating an D-END-request.

Note.— The time period to wait before releasing a dialogue is a local matter to be determined by cost and expected data traffic.

3.1.3.3.2.4.16 The parameters of the D-END-request shall be set according to Table 3.1.3-4.

Table 3.1.3-4. D-END-request Parameters

D-END-request Parameter	Type	Value
User Data		<not used>

3.1.3.3.2.4.17 Upon receiving an D-END-indication, the CF shall release the dialogue as soon as it no longer has any data to send (over that dialogue) by formulating a D-END-response.

3.1.3.3.2.4.18 The parameters of the D-END-response shall be set according to Table 3.1.3-5.

Table 3.1.3-5. D-END-response Parameters

D-END-response Parameter	Type	Value
Result	INT	abstract-value “accepted”
User Data	----	<not used>

3.1.3.3.2.4.19 For immediate termination of the dialogue, the D-ABORT parameters shall be set according to Table 3.1.3-6.

Note.— The conditions under which this primitive is used is a local matter.

Table 3.1.3-6. D-ABORT-request Parameters

D-ABORT-request Parameter	Type	Value
Originator	-----	<not used>
User Data	-----	<not used>

3.1.3.3.2.4.20 Upon receipt of a D-ABORT-indication the dialogue shall be considered to have failed.

Note.— The recovery mechanisms, if any, to be applied are a matter of local implementation choices.

3.1.3.3.2.4.21 Upon receipt of a D-P-ABORT-indication the dialogue shall be terminated.

Note.— The recovery mechanisms, if any, to be applied are a matter of local implementation choices.

3.1.3.3.2.5 Dialogue Service QOS (Priority)

3.1.3.3.2.5.1 For transmission of messages across the ATN, the AFTN priority indicators, as found in Annex 10, Volume II, 4.4.1.2, shall map to Dialogue Service QOS (Priorities) in accordance with Table 3.1.3-7.

Table 3.1.3-7. AFTN/ATN Priority Mapping

AFTN Priority Indicator	Dialogue Service Quality of Service (Priority) Parameter
SS	distress communications
DD	urgent communications
FF	high priority flight safety messages
GG	flight regularity communications
KK	aeronautical administrative messages

3.1.3.3.2.5.2 The ATN component shall process incoming and outgoing messages according to the priority of the message.

3.1.3.3.3 Message Transfer and Control Unit Component

Note.— The Message Transfer and Control Unit Component provides a bi-directional conversion facility between the AFTN component and the ATN component and consists of:

- a) *a set of general functions as specified in 3.1.3.3.3.1;*
- b) *a set of AFTN to ATN mapping functions as specified in 3.1.3.3.3.3;*
- c) *a set of ATN to AFTN mapping functions as specified in 3.1.3.3.3.4;*
- d) *a set of interface requirements between the Message Transfer and Control Unit Component and the ATN Component as specified in 3.1.3.3.3.5; and*
- e) *a set of interface requirements between the Message Transfer and Control Unit Component and the AFTN Component as specified in 3.1.3.3.3.6.*

3.1.3.3.3.1 General functions

The Message Transfer and Control Unit of an AFTN/ATN Type A Gateway shall log all messages and information related to the following events that have occurred at its interfaces with the ATN Component and with the AFTN Component, and in its internal procedures:

- a) the messages transferred out (to the ATN Component);
- b) the messages transferred in (from the ATN Component);
- c) the AFTN messages conveyed out (to the AFTN Component);
- d) the AFTN messages conveyed in (from the AFTN Component);
- e) the AFTN service messages indicating unknown addressee indicator conveyed out (to the AFTN Component).

Note.— This requirement is not intended to fulfill the 30 day message requirements for an AFTN station.

3.1.3.3.3.2 Determination of gateway address

3.1.3.3.3.2.1 The Message Transfer and Control Unit Component shall maintain an address mapping function which maps between an AFTN addressee indicator and the ATN address of the AFTN/ATN Type A Gateway via which the addressee may be reached.

3.1.3.3.3.2.2 The address mapping function shall, at a minimum, provide the following mappings:

- a) a map from an entire AFTN address to an ATN address,

- b) a map from sets of AFTN addresses based on a portion of the AFTN address to a single ATN address.

Note.— All AFTN address Indicators are treated as explicit addresses, including predetermined address indicators (PDAIS), thus a single AFTN address can only map to a single ATN address.

3.1.3.3.3.2.3 Recommendation.— *The address mapping function should provide a default mapping of an AFTN Addressee Indicator to an alternate gateway ATN address when the primary gateway ATN address is not in service.*

3.1.3.3.3.3 AFTN to ATN mapping

3.1.3.3.3.3.1 Upon the reception by the Message Transfer and Control Unit of a message passed from the AFTN Component, it shall examine the AFTN Address Indicators to determine the onward routing requirements of the message over the ATN Internet.

3.1.3.3.3.3.2 Prior to delivery of the message to the ATN Component, the Message Transfer and Control Unit Component shall apply the address stripping procedures defined in Annex 10, Volume II, 4.4.8 to omit from the address any AFTN Address Indicators not related to the selected ATN address and provide for message replication if more than one ATN address is required.

Note.— In applying the procedures of 3.1.3.3.3.3.2 the Message Transfer and Control Unit Component provides sufficient copies of the message to reach each ATN address obtained by applying the procedures of 3.1.3.3.3.2.1. In most cases, a set of AFTN addresses will map to a single ATN address (the address of the corresponding ATN Gateway).

3.1.3.3.3.3.3 The Message Transfer and Control Unit shall send an appropriate service message to the AFTN originator indicator advising of an unknown address indicator according to the following:

- a) the abbreviation SVC,
- b) the procedure signal ADS,
- c) the alignment function,
- d) the indication UNKNOWN,
- e) the unknown address indicator(s),
- f) the end-of-text signal.

3.1.3.3.3.4 ATN to AFTN mapping

3.1.3.3.3.4.1 Upon the reception by the Message Transfer and Control Unit of a GA-Data-Indication passed from the ATN Component, the message shall be extracted from the User Data parameter.

3.1.3.3.3.4.2 The extracted message shall be passed unmodified to the AFTN Component.

3.1.3.3.3.5 Interface between the ATN Component and the Message Control Unit Component

3.1.3.3.3.5.1 The interface between the ATN Component and the Message Control Unit Component shall be according to the ATN Component service as specified in 3.1.3.3.2.2.

3.1.3.3.3.5.2 To send an AFTN message across the ATN, the Message Control Unit Component shall invoke a GA-Data-request primitive to the ATN Component.

Note.— The requirement to invoke the GA-Data-request to the ATN component is not intended to constrain an implementation. The requirement is to pass the required information to the ATN component in a manner consistent with the logical service.

3.1.3.3.3.5.3 The AFTN message, forwarded by the Message Transfer and Control Unit, shall comprise the User Data parameter.

3.1.3.3.3.5.4 The called address parameter in the GA-Data-request shall be the facility designation of the destination ATN-end system.

3.1.3.3.3.5.5 The calling address parameter in the GA-Data-request shall be the local facility designation of the AFTN/ATN Type A Gateway.

3.1.3.3.3.5.6 The AFTN priority parameter in the GA-Data-request shall be set according to the value of the AFTN priority indicator of the message.

3.1.3.3.3.5.7 Upon receipt of a D-DATA-indication primitive, the ATN Component shall invoke a GA-Data-indication to the Message Transfer and Control Unit Component.

Note.— The requirement to invoke an GA-Data-indication primitive to the Message Transfer and Control Unit Component is not intended to constrain an implementation. The requirement is to pass the required information with the Message Transfer and Control Unit Component in a manner consistent with the logical service.

3.1.3.3.3.5.8 The AFTN message, as found in the User Data parameter of the D-DATA-indication, shall comprise the User Data parameter of the GA-Data-indication.

3.1.3.3.3.5.9 The calling address parameter in the GA-Data-indication shall be the facility designation of the AFTN/ATN Type A Gateway which initiated the GA-Data-request, as found in the D-START-indication Calling Address parameter.

3.1.3.3.3.5.10 The AFTN priority parameter, if present in the GA-Data-indication, shall be derived, using Table 3.1.3-7, from the value of the QOS (priority) parameter of the corresponding D-START-indication.

3.1.3.3.3.6 Interface between the AFTN Component and the Message Control Unit Component

3.1.3.3.3.6.1 All AFTN messages or service messages passed by the AFTN Component to the Message Transfer and Control Unit shall be transferred in the order received.

3.1.3.3.3.6.2 An AFTN message or service message passed by the Message Transfer and Control Unit to the AFTN Component shall be transferred in the order received.

3.2 ATS INTERFACILITY DATA COMMUNICATIONS

3.2.1 INTRODUCTION

The AIDC application exchanges information between ATS Units (ATSUs) for support of critical Air Traffic Control (ATC) functions, such as notification of flights approaching a Flight Information Region (FIR) boundary, coordination of boundary conditions and transfer of control and communications authority.

AIDC is strictly an ATC application for exchanging tactical control information between ATS units, not with other offices or facilities.

Structure of this document

- a) 3.2.1: INTRODUCTION identifies the document's structure, the functions of the AIDC application and a description of the AIDC functional model.
- b) 3.2.2: GENERAL REQUIREMENTS identifies the version of the AIDC application and the Upper Layer requirements for the AIDC application.
- c) 3.2.3: THE AIDC-AE ABSTRACT SERVICE describes the AIDC-AE service and the associated primitives provided to the user of the AIDC service.
- d) 3.2.4: THE AIDC-ASE SERVICE describes the AIDC-ASE service and the associated primitives.
- e) 3.2.5: THE AIDC CONTROL FUNCTION describes the Control Function (CF) mapping of the AIDC-user invoked primitives, the AIDC-ASE service, the ACSE service and the service provided by the communications service provider.
- f) 3.2.6: THE AIDC-ASE PROTOCOL DEFINITION describes the exchanges of messages allowed by the AIDC protocol, as well as time constraints and AIDC-ASE protocol descriptions.
- g) 3.2.7: FORMAL DEFINITIONS contains the ASN.1 abstract syntax for the AIDC-AE.
- h) 3.2.8: COMMUNICATION REQUIREMENTS contains the requirements that the AIDC application imposes on the underlying communication system.
- i) 3.2.9: AIDC USER REQUIREMENTS defines the requirements that the user of the AIDC service must meet.
- j) 3.2.10: SEQUENCE DIAGRAMS AND PRIMITIVE SEQUENCING

3.2.1.1 General

3.2.1.1.1 **Recommendation.**— *AIDC is an ATN application which should be employed by two Air Traffic Service (ATS) units when exchanging Air Traffic Control (ATC) information for an active flight.*

3.2.1.2 AIDC Functional Descriptions

Flight Notification: — This function allows the Controlling ATS Unit (C-ATSU) to notify the Downstream ATS Unit (D-ATSU) of a flight's cleared profile some time before the flight enters the D-ATSU's area of interest. This function may be initiated a multiple number of times for the same flight, depending on the number and type of changes made to the flight's cleared profile.

Flight Coordination: — This function allows the C-ATSU to coordinate the conditions of transfer for a flight with a D-ATSU.

Transfer of Control: — This function allows the C-ATSU to transfer control authority for a flight to the Receiving ATS Unit (R-ATSU) and allows the R-ATSU to accept the control authority for the flight.

Transfer of Communications: — This function allows one of the following to take place:

- a) the C-ATSU to transfer the control authority and communications authority for a flight to the R-ATSU and the R-ATSU to accept the control authority and communications authority for the flight; or
- b) the R-ATSU to take the control authority and communications authority for a flight.

Transfer of Surveillance Data: — This function allows an ATSU1 to transfer surveillance data to an ATSU2.

General Information Interchange: — This function allows an ATSU1 to exchange general flight related data including free text messages (i.e. unstructured) with an ATSU2.

3.2.1.3 The AIDC Functional Model

Figure 3.2.1-1 shows the functional model of the AIDC Application.

The functional elements identified in this model are the following :

- a) the AIDC-User,
- b) the AIDC Application Entity (AIDC-AE) Service Interface,
- c) the AIDC-AE,
- d) the AIDC Control Function (CF),
- e) the AIDC Application Service Element (AIDC-ASE) Service Interface,
- f) the AIDC-ASE,
- g) the Association Control Service Element, and
- h) the ATN Service Provider Interface.

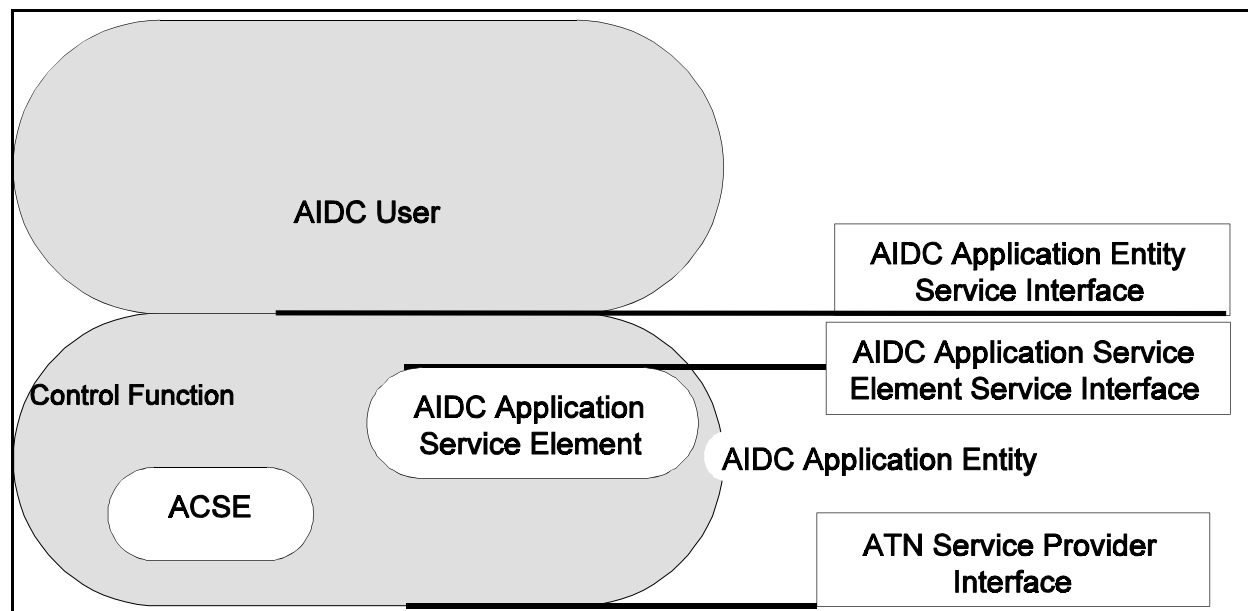


Figure 3.2.1-1: Functional Model of the AIDC Application

The AIDC-User represents the operational part of the AIDC system. This user does not perform the communication functions but relies on a communication service provided by the AIDC-AE through the AIDC-AE service interface.

The AIDC-ASE is the element in the communication system which executes the AIDC specific protocol. In other words, it enforces the AIDC specific primitive sequencing actions, timer management, and error handling.

The AIDC-AE CF is responsible for mapping primitives received from one element to another within the AIDC functional model.

3.2.1.4 Modelling Conventions

In modelling the AIDC application, each service description includes a table listing both the service primitives and the parameters of the service.

For a given primitive, the presence of each parameter is described by one of the following values in the parameter tables:

blank	not applicable;
C	conditional upon some predicate explained in the text;
C(=)	conditional upon the value of the parameter to the left being present. In addition, the value of the parameter is equal to the value of the parameter to the left;
M	mandatory;
M(=)	mandatory. In addition, the value of the parameter is equal to the value of the parameter to the left;
U	user option.

The following abbreviations are used in this part:

Req — request; data is input by the user initiating the service.

Ind — indication; data is indicated by the service to its respective user.

Rsp — response; data is input by the service user.

Cnf — confirmation; data is confirmed by the service to its respective user.

3.2.2 GENERAL REQUIREMENTS

3.2.2.1 AIDC-AE Version Number

3.2.2.1.1 For the first version of the AIDC application, the AIDC-AE version number shall be set to 1 (one).

3.2.2.2 Upper Layer Requirements

3.2.2.2.1 The AIDC application shall utilise the Upper Layer Architecture as defined in Sub-Volume IV.

Note.— The basis of the Upper Layer Architecture is the Application Layer, consisting of an Application Entity (AE) formed by an Application Service Element (ASE), an Association Control Service Element (ACSE) and a Control Function (CF), using the efficiency enhancements of the Presentation and Session Layers

3.2.2.2.2 The AIDC application shall use the following aspects contained in Sub-Volume IV:

- a) the Naming, Addressing, Presentation Context Identification and Registration from 4.3.2;
- b) the APRL for Session from 4.4;
- c) the APRL for Presentation from 4.5;
- d) the APRL for ACSE as specified in 4.6.

3.2.3 THE AIDC-AE ABSTRACT SERVICE

Note.— The following defines the abstract service interface used by an AIDC-User to access the AIDC-AE services.

3.2.3.1 Standard Parameters

Note.— The following service primitive parameters are defined here rather than repeated for each service definition.

3.2.3.1.1 Called ICAO Facility Designation

3.2.3.1.1.1 The *Called ICAO Facility Designation* parameter shall be provided by the AIDC-User.

3.2.3.1.1.2 The *Called ICAO Facility Designation* parameter shall be the called ATS system's ICAO four-letter location indicator or the ICAO eight-letter combined location indicator, three letters designator and an additional letter.

3.2.3.1.2 Calling ICAO Facility Designation

3.2.3.1.2.1 The *Calling ICAO Facility Designation* parameter shall be provided by the AIDC-AE.

3.2.3.1.2.2 The *Calling ICAO Facility Designation* parameter shall be the calling ATS system's ICAO four-letter location indicator or the ICAO eight-letter combined location indicator, three letters designator and an additional letter.

3.2.3.1.3 Message Number

3.2.3.1.3.1 The *Message Number* parameter shall be provided by the AIDC-User except when invoking the User-abort or User-confirmation request primitives.

3.2.3.1.3.2 The *Message Number* parameter value shall consist of a unique identifier for reference in the User-confirmation.

3.2.3.1.3.3 The *Message Number* parameter shall conform to the ASN.1 abstract syntax of *MessageNumber*.

3.2.3.2 AIDC-User Service

3.2.3.2.1 The AIDC service shall exhibit external behaviour consistent with having implemented an AIDC-AE with the following abstract services, making them available to the AIDC-User:

- a) User-confirmation service as defined in 3.2.3.4.
- b) Notify service as defined in 3.2.3.5.1
- c) Coordinate-start service as defined in 3.2.3.6.4
- d) Coordinate-end service as defined in 3.2.3.6.5

- e) Coordinate-negotiate service as defined in 3.2.3.6.6
- f) Coordinate-standby service as defined in 3.2.3.6.7
- g) Transfer-initiate service as defined in 3.2.3.7.3
- h) Transfer-request service as defined in 3.2.3.7.4
- i) Transfer-conditions-proposal service as defined in 3.2.3.7.5
- j) Transfer-conditions-accept service as defined in 3.2.3.7.6
- k) Transfer-control service as defined in 3.2.3.7.7
- l) Transfer-communication service as defined in 3.2.3.7.8
- m) Transfer-communication-assume service as defined in 3.2.3.7.9
- n) Info-transfer service as defined in 3.2.3.8.1
- o) End service as defined in 3.2.3.9.1
- p) User-abort service as defined in 3.2.3.9.2
- q) Provider-abort service as defined in 3.2.3.9.3

3.2.3.3 Service Primitive Sequencing

3.2.3.3.1 The AIDC-User Service shall consist of three regimes, one asynchronous service, and a set of termination services.

3.2.3.3.2 The three regimes shall occur in a sequence:

- a) Notifying regime,
- b) Coordinating regime, and
- c) Transferring regime.

3.2.3.3.3 The Coordinating regime shall be allowed to be re-entered at the end of the Transferring regime.

3.2.3.3.4 **Recommendation.**— *The Notifying regime should consist of the Notify service.*

3.2.3.3.5 **Recommendation.**— *The Coordinating regime should consist of the:*

- a) *Coordinate-start service,*
- b) *Coordinate-negotiate service,*

- c) *Coordinate-standby service, and*
- d) *Coordinate-end service.*

3.2.3.3.6 **Recommendation.**— *The Transferring regime should consist of the:*

- a) *Transfer-initiate service,*
- b) *Transfer-conditions-proposal service,*
- c) *Transfer-conditions-accept service,*
- d) *Transfer-request service,*
- e) *Transfer-control service,*
- f) *Transfer-communication service, and*
- g) *Transfer-communication-assume service.*

3.2.3.4 User Confirmation Service

3.2.3.4.1 Upon the receipt of any primitive indication containing an Information parameter (e.g. Notification Information, Coordinate Start Information), the AIDC-User shall validate the value of this parameter and respond with a User-confirmation request primitive.

3.2.3.4.2 The User-confirmation service shall be an unconfirmed service.

3.2.3.4.3 Table 3.2.3-1 specifies the parameters that shall be passed when the primitives of the User-confirmation service are invoked.

Table 3.2.3-1: User-confirmation Service Primitive Parameters

Parameter Name	Req	Ind
Result	M	M(=)
Reason	U	C(=)
Referenced Number	M	M(=)

3.2.3.4.4 Result

3.2.3.4.4.1 The *Result* parameter shall be provided by the AIDC-User.

3.2.3.4.4.2 The *Result* parameter shall conform to the ASN.1 abstract syntax of *Result*.

3.2.3.4.5 Reason

3.2.3.4.5.1 **Recommendation.**— *The Reason parameter should be provided by the AIDC-User when the Response parameter has the abstract value of “rejected”.*

3.2.3.4.5.2 The Reason parameter shall conform to the ASN.1 abstract syntax *ApplicationErrorData*.

3.2.3.4.6 Referenced Number

3.2.3.4.6.1 The *Referenced Number* parameter shall be provided by the AIDC-User.

3.2.3.4.6.2 The *Referenced Number* parameter shall contain the Message Number of the message being confirmed.

3.2.3.4.6.3 The *Referenced Number* parameter shall conform to the ASN.1 abstract syntax of *MessageNumber*.

3.2.3.5 Notifying Regime

3.2.3.5.1 Notify Service

Note.— *The purpose of the Notify service is to allow a C-ATSU to update the information a D-ATSU maintains on a flight that is expected to enter its area of interest at some future time.*

3.2.3.5.1.1 Service Primitives

3.2.3.5.1.1.1 The Notify service shall be an unconfirmed service.

3.2.3.5.1.1.2 Table 3.2.3-2 specifies the parameters that shall be passed when the primitives of the Notify service are invoked.

Table 3.2.3-2: Notify Service Primitives Parameters

Parameter Name	Req	Ind
Called ICAO Facility Designation	M	
Calling ICAO Facility Designation		M
Notification Information	M	M(=)
Message Number	M	M(=)

3.2.3.5.1.1.3 Notification Information

3.2.3.5.1.1.3.1 The Notification Information parameter shall be provided by the AIDC-User when invoking the Notify request primitive.

3.2.3.5.1.1.3.2 The Notification Information parameter shall conform to the ASN.1 abstract syntax *Notify*.

3.2.3.6 Coordinating Regime

- 3.2.3.6.1 The Coordinating regime shall begin with the invocation of a Coordinate-start primitive.
- 3.2.3.6.2 The Coordinating regime shall end with the invocation of a Coordinate-end primitive.
- 3.2.3.6.3 Upon entering the Coordinating regime, further use of the Notify service shall be prohibited.
- 3.2.3.6.4 Coordinate-start Service

Note 1.— The purpose of the Coordinate-start service is to allow an ATSU1 to begin the coordination of, or update, the conditions of transfer of a flight with an ATSU2.

Note 2.— If rejected, any currently agreed coordination conditions remain in affect.

3.2.3.6.4.1 Service Primitives

- 3.2.3.6.4.1.1 The Coordinate-start service shall be an unconfirmed service.
- 3.2.3.6.4.1.2 Table 3.2.3-3 specifies the parameters that shall be passed when the primitives of the Coordinate-start service are invoked.

Table 3.2.3-3: Coordinate-start Service primitive Parameters

Parameter Name	Req	Ind
Called ICAO Facility Designation	M	
Calling ICAO Facility Designation		M
Coordinate Start Information	M	M(=)
Message Number	M	M(=)

3.2.3.6.4.1.3 Coordinate Start Information

- 3.2.3.6.4.1.3.1 The *Coordinate Start Information* parameter shall be provided by the AIDC-User.
- 3.2.3.6.4.1.3.2 The *Coordinate Start Information* parameter shall conform to the ASN.1 abstract syntax of *CoordinateInitial* or *CoordinateUpdate*.

3.2.3.6.5 Coordinate-end Service

Note 1.— The purpose of the Coordinate-end service is to allow an ATSU1 to notify an ATSU2 that the ATSU1 either accepts or rejects the coordination conditions proposed.

Note 2.— The successful completion of this service ends the coordinating regime for a flight.

3.2.3.6.5.1 Service Primitives

3.2.3.6.5.1.1 The Coordinate-end service shall be an unconfirmed service.

3.2.3.6.5.1.2 The Coordinate-end service primitives shall only be invoked while in the Coordinating regime.

3.2.3.6.5.1.3 Table 3.2.3-4 specifies the parameters that shall be passed when the primitives of the Coordinate-end service are invoked.

Table 3.2.3-4: Coordinate-end Service Primitive Parameters

Parameter Name	Req	Ind
Coordinate End Information	M	M(=)
Result	M	
Message Number	M	M(=)

3.2.3.6.5.1.4 Coordinate End Information

3.2.3.6.5.1.4.1 The *Coordinate End Information* parameter shall be provided by the AIDC-User.

3.2.3.6.5.1.4.2 The *Coordinate End Information* parameter shall conform to the ASN.1 abstract syntax of *CoordinateAccept* or *CoordinateReject*.

3.2.3.6.5.1.5 Result

3.2.3.6.5.1.5.1 The *Result* parameter shall be provided by the AIDC-User.

3.2.3.6.5.1.5.2 The *Result* parameter shall conform to the ASN.1 abstract syntax of *Result*.

3.2.3.6.6 Coordinate-negotiate Service

Note 1.— The purpose of the Coordinate-negotiate service is to allow an ATSU1 to negotiate modifications to a flight's existing coordination conditions with an ATSU2.

Note 2.— If rejected, any currently agreed coordination conditions remain in affect.

3.2.3.6.6.1 Service Primitives

3.2.3.6.6.1.1 The Coordinate-negotiate service shall be an unconfirmed service.

3.2.3.6.6.1.2 The Coordinate-negotiate service primitives shall only be invoked while in the Coordinating regime.

3.2.3.6.1.3 Table 3.2.3-5 specifies the parameters that shall be passed when the primitives of the Coordinate-negotiate service are invoked.

Table 3.2.3-5: Coordinate-negotiate Service Primitive Parameters

Parameter Name	Req	Ind
Coordinate Negotiate Information	M	M(=)
Message Number	M	M(=)

3.2.3.6.1.4 Coordinate Negotiate Information

3.2.3.6.1.4.1 The *Coordinate Negotiate Information* parameter shall be provided by the AIDC-User.

3.2.3.6.1.4.2 The *Coordinate Negotiate Information* parameter shall conform to the ASN.1 abstract syntax *CoordinateNegotiate*.

3.2.3.6.7 Coordinate-standby Service

Note 1.— The purpose of the Coordinate-standby service is to allow an ATSU1 to notify an ATSU2 that the coordination dialogue between the ATSUs is being temporarily suspended.

Note 2.— Each invocation of this service effectively extends, by a defined amount, the time before a user response to a coordinate service indication is given.

3.2.3.6.7.1 Service Primitives

3.2.3.6.7.1.1 The Coordinate-standby service shall be an unconfirmed service.

3.2.3.6.7.1.2 The Coordinate-standby service primitives shall only be invoked while in the Coordinating regime.

3.2.3.6.7.1.3 Table 3.2.3-6 specifies the parameters that shall be passed when the primitives of the Coordinate-standby service are invoked.

Table 3.2.3-6: Coordinate-standby Service Primitive Parameters

Parameter Name	Req	Ind
Coordinate Standby Information	M	M(=)
Message Number	M	M(=)

3.2.3.6.7.1.4 Coordinate Standby Information

3.2.3.6.7.1.4.1 The *Coordinate Standby Information* parameter shall be provided by the AIDC-User.

3.2.3.6.7.1.4.2 The *Coordinate Standby Information* parameter shall conform to the ASN.1 abstract syntax *CoordinateStandby*.

3.2.3.7 Transferring regime

3.2.3.7.1 The Transferring regime shall be entered after the completion of the Coordinating regime.

3.2.3.7.2 The Transferring regime shall begin with the invocation of any of the following:

- a) the Transfer-initiate service;
- b) the Transfer-request service; or
- c) the Transfer-control service.

3.2.3.7.3 Transfer-initiate Service

Note. — The purpose of the Transfer-initiate service is to allow a C-ATSU to initiate the transfer phase for a flight by passing executive control information to an R-ATSU.

3.2.3.7.3.1 Service Primitives

3.2.3.7.3.1.1 The Transfer-initiate service shall be an unconfirmed service.

3.2.3.7.3.1.2 Table 3.2.3-7 specifies the parameters that shall be passed when the primitives of the Transfer-initiate service are invoked.

Table 3.2.3-7: Transfer-initiate Service Primitive Parameters

Parameter Name	Req	Ind
Transfer Initiate Information	M	M(=)
Message Number	M	M(=)

3.2.3.7.3.1.3 Transfer Initiate Information

3.2.3.7.3.1.3.1 The *Transfer Initiate Information* parameter shall be provided by the AIDC-User.

3.2.3.7.3.1.3.2 The *Transfer Initiate Information* parameter shall conform to the ASN.1 abstract syntax *TransferInitiate*.

3.2.3.7.4 Transfer-request Service

Note.— The purpose of the Transfer-request service is to allow an R-ATSU to request control and communications authority for a flight from a C-ATSU.

3.2.3.7.4.1 Service Primitives

3.2.3.7.4.1.1 The Transfer-request service shall be an unconfirmed service.

3.2.3.7.4.1.2 Table 3.2.3-8 specifies the parameters that shall be passed when the primitives of the Transfer-request service are invoked.

Table 3.2.3-8: Transfer-request Service Primitive Parameters

Parameter Name	Req	Ind
Transfer Request Information	M	M(=)
Message Number	M	M(=)

3.2.3.7.4.1.3 Transfer Request Information

3.2.3.7.4.1.3.1 The *Transfer Request Information* parameter shall be provided by the AIDC-User.

3.2.3.7.4.1.3.2 The *Transfer Request Information* parameter shall conform to the ASN.1 abstract syntax *TransferRequest*.

3.2.3.7.5 Transfer-conditions-proposal Service

Note.— The purpose of the Transfer-conditions-proposal service is to allow a C-ATSU to propose the conditions under which control authority for a flight can be given to an R-ATSU.

3.2.3.7.5.1 Service Primitives

3.2.3.7.5.1.1 The Transfer-conditions-proposal service shall be an unconfirmed service.

3.2.3.7.5.1.2 The Transfer-conditions-proposal service primitives shall only be invoked after the successful completion of the Transfer-initiate service.

3.2.3.7.5.1.3 Table 3.2.3-9 specifies the parameters that shall be passed when the primitives of the Transfer-conditions-proposal service are invoked.

Table 3.2.3-9: Transfer-conditions-proposal Service Primitive Parameters

Parameter Name	Req	Ind
Transfer Conditions Proposal Information	M	M(=)
Message Number	M	M(=)

3.2.3.7.5.1.4 Transfer Conditions Proposal Information

3.2.3.7.5.1.4.1 The *Transfer Conditions Proposal Information* parameter is provided by the AIDC-User.

3.2.3.7.5.1.4.2 The *Transfer Conditions Proposal Information* parameter shall conform to the ASN.1 abstract syntax *TransferConditionsProposal*.

3.2.3.7.6 Transfer-conditions-accept Service

Note 1.— The purpose of the Transfer-conditions-accept service is to allow an R-ATSU to indicate that it is willing to accept control conditions proposed for a flight by the C-ATSU.

Note 2.— This service, when used, is only used in response to the Transfer-conditions-proposal service.

3.2.3.7.6.1 Service Primitives

3.2.3.7.6.1.1 The Transfer-conditions-accept service shall be an unconfirmed service.

3.2.3.7.6.1.2 The Transfer-conditions-accept service primitives shall only be invoked after the successful completion of the Transfer-conditions-proposal service.

3.2.3.7.6.1.3 Table 3.2.3-10 specifies the parameters that shall be passed when the primitives of the Transfer-conditions-accept service are invoked.

Table 3.2.3-10: Transfer-conditions-accept Service Primitive Parameters

Parameter Name	Req	Ind
Transfer Conditions Accept Information	M	M(=)
Message Number	M	M(=)

3.2.3.7.6.1.4 Transfer Conditions Accept Information

3.2.3.7.6.1.4.1 The *Transfer Conditions Accept Information* parameter shall be provided by the AIDC-User.

3.2.3.7.6.1.4.2 The *Transfer Conditions Accept Information* parameter shall conform to the ASN.1 abstract syntax *TransferConditionsAccept*.

3.2.3.7.7 Transfer-control Service

Note.— The purpose of the Transfer-control service is to allow a C-ATSU to indicate that it wants to relinquish control authority for a flight to an R-ATSU.

3.2.3.7.7.1 Service Primitives

3.2.3.7.7.1.1 The Transfer-control service shall be a confirmed service.

3.2.3.7.7.1.2 Table 3.2.3-11 specifies the parameters that shall be passed when the primitives of the Transfer-control service are invoked.

Table 3.2.3-11: Transfer-control Service Primitive Parameters

Parameter Name	Req	Ind	Rsp	Cnf
Transfer Control Information	M	M(=)	M	M(=)
Result			M	M(=)
Message Number	M	M(=)	M	M(=)

3.2.3.7.7.1.3 Transfer Control Information

3.2.3.7.7.1.3.1 The *Transfer Control Information* parameter shall be provided by the C-ATSU AIDC-User when invoking the Transfer-control request primitive.

3.2.3.7.7.1.3.2 The *Transfer Control Information* parameter shall conform to the ASN.1 abstract syntax *TransferControl*, when the C-ATSU AIDC-User invokes the Transfer-control request primitive.

3.2.3.7.7.1.3.3 The *Transfer Control Information* parameter shall be provided by the R-ATSU AIDC-User when invoking the Transfer-control response service primitive.

3.2.3.7.7.1.3.4 The *Transfer Control Information* parameter shall conform to the ASN.1 abstract syntax *TransferControlData*, when the R-ATSU AIDC-User invokes the Transfer-control response primitive.

3.2.3.7.7.1.4 Result

3.2.3.7.7.1.4.1 The *Result* parameter shall be provided by the C-ATSU AIDC-User.

3.2.3.7.7.1.4.2 The *Result* parameter shall conform to the ASN.1 abstract syntax *Result*.

3.2.3.7.7.1.4.3 When the *Result* parameter value is set to “accept”, the Transfer Control Information parameter shall conform to the ASN.1 abstract syntax *TransferControlAssume*.

3.2.3.7.7.1.4.4 When the *Result* parameter value is set to “reject”, the Transfer Control Information parameter shall conform to the ASN.1 abstract syntax *TransferControlReject*.

3.2.3.7.8 Transfer-communication Service

Note.— The purpose of the Transfer-communication service is to allow a C-ATSU to indicate that it, is relinquishing communication authority for a flight to an R-ATSU.

3.2.3.7.8.1 Service Primitives

3.2.3.7.8.1.1 The Transfer-communication service shall be an unconfirmed service.

3.2.3.7.8.1.2 The Transfer-communication service primitives shall only be invoked after the successful completion of the Transfer-initiate service.

3.2.3.7.8.1.3 Table 3.2.3-12 specifies the parameters that shall be passed when the primitives of the Transfer-communication service are invoked.

Table 3.2.3-12: Transfer-communication Service Primitive Parameters

Parameter Name	Req	Ind
Transfer Communication Information	M	M(=)
Message Number	M	M(=)

3.2.3.7.8.1.4 Transfer Communication Assume Information

3.2.3.7.8.1.4.1 The *Transfer Communication Information* parameter shall be provided by the AIDC-User.

3.2.3.7.8.1.4.2 The *Transfer Communication Information* parameter shall conform to the ASN.1 abstract syntax *TransferCommunication*.

3.2.3.7.9 Transfer-communication-assume Service

Note.— The purpose of the Transfer-communication-assume service is to allow an R-ATSU to indicate to a C-ATSU that communication with a flight has been established.

3.2.3.7.9.1 Service Primitives

3.2.3.7.9.1.1 The Transfer-communication-assume service shall be an unconfirmed service.

3.2.3.7.9.1.2 The Transfer-communication-assume service primitives shall only be invoked after the successful completion of the Transfer-initiate service.

3.2.3.7.9.1.3 Table 3.2.3-13 specifies the parameters that shall be passed when the primitives of the Transfer-communication-assume service are invoked.

Table 3.2.3-13: Transfer-communication-assume Service Primitive Parameters

Parameter Name	Req	Ind
Transfer Communication Assume Information	M	M(=)
Message Number	M	M(=)

3.2.3.7.9.1.4 Transfer Communication Assume Information

3.2.3.7.9.1.4.1 The *Transfer Communication Assume Information* parameter shall be provided by the AIDC-User.

3.2.3.7.9.1.4.2 The *Transfer Communication Assume Information* parameter shall conform to the ASN.1 abstract syntax *TransferCommunicationAssume*.

3.2.3.8 Asynchronous Services

3.2.3.8.1 Info-transfer Service

Note 1.— The Info-transfer service permits an ATSU1 to transmit general executive data, surveillance data, general free text data, or emergency free text data, or to point-out a flight to an ATSU2.

Note 2.— The Info-transfer service may be invoked by the C-ATSU even when no regime has been established.

3.2.3.8.1.1 Service Primitives

3.2.3.8.1.1.1 It shall be possible, for any ATSU to invoke the primitives of the Info-transfer service at any time after the initial invocation of primitives of the Notify service or after the initial invocation of the Coordinate-start service primitives when the Notify service is not used.

3.2.3.8.1.1.2 It shall be possible for the C-ATSU only, to invoke the primitives of the Info-transfer service before invoking any other service primitives.

3.2.3.8.1.2 The Info-transfer service shall be an unconfirmed service.

3.2.3.8.1.3 Table 3.2.3-14 specifies the parameters that shall be passed when the primitives of the Info-transfer service are invoked.

Table 3.2.3-14: Info-transfer Service Primitive Parameters

Parameter Name	Req	Ind
Called ICAO Facility Designation	C	
Calling ICAO Facility Designation		C
Information	M	M(=)
Message Number	M	M(=)

3.2.3.8.1.4 Called ICAO Facility Designation

3.2.3.8.1.4.1 The *Called ICAO Facility Designation* parameter shall be supplied, as specified in 3.2.3.1.1, only when the Info-transfer request primitive is invoked and no regime has yet been established by the AIDC-user.

3.2.3.8.1.5 Calling ICAO Facility Designation

3.2.3.8.1.5.1 The *Calling ICAO Facility Designation* parameter shall be supplied, as specified in 3.2.3.1.2, only when the Info-transfer indication primitive corresponds to a request primitive in which the Called ICAO Facility Designation was supplied.

3.2.3.8.1.6 Information

3.2.3.8.1.6.1 The *Information* parameter shall be provided by the AIDC-User.

3.2.3.8.1.6.2 The *Information* parameter shall conform to the ASN.1 abstract syntax of any of the following:

- a) *GeneralExecutiveData*,
- b) *GeneralPoint*,
- c) *SurveillanceData*,
- d) *GeneralFreeText*, or
- e) *EmergencyFreeText*.

3.2.3.9 Termination Services

3.2.3.9.1 End Service

Note.— The purpose of the End service is to allow an ATSU1 to indicate to an ATSU2 that it is in the process of:

- a) *cancelling the notification for a flight and terminating the AIDC service; or*
- b) *cancelling the coordination for a flight and terminating the AIDC service; or*
- c) *terminating the AIDC service.*

3.2.3.9.1.1 Service Primitives

3.2.3.9.1.1.1 The End service shall be an unconfirmed service.

3.2.3.9.1.1.2 Table 3.2.3-15 specifies the parameters that shall be passed when the primitives of the End service are invoked.

Table 3.2.3-15: End Service primitive Parameters

Parameter Name	Req	Ind
Cancel Information	C	C(=)
Message Number	M	M(=)

3.2.3.9.1.1.3 Cancel Information

3.2.3.9.1.1.3.1 The *Cancel Information* parameter shall be provided by the AIDC-User when cancelling the notification or coordination for a flight.

3.2.3.9.1.1.3.2 The *Cancel Information* parameter shall conform to the ASN.1 abstract syntax of *Cancel*.

3.2.3.9.2 User-abort Service

Note 1.— The purpose of the User-abort service is to allow an AIDC-User to immediately terminate the AIDC service.

Note 2.— The User-abort service may be used for both operational and technical reasons.

3.2.3.9.2.1 Service Primitives

3.2.3.9.2.1.1 The User-abort service shall be an unconfirmed service.

3.2.3.9.2.1.2 It shall be possible to invoke the primitives of the User-abort service at any time.

3.2.3.9.2.1.3 The User-abort service primitives shall have no parameters.

3.2.3.9.3 Provider-abort Service

Note.— The purpose of the Provider-abort service is to provide the capability for the AIDC-service provider to inform the AIDC-User that it can no longer provide the AIDC service.

3.2.3.9.3.1 Service Primitives

3.2.3.9.3.1.1 The Provider-abort service shall be an unconfirmed service.

3.2.3.9.3.1.2 The primitives of the Provider-abort service shall be invoked by the AIDC-AE service-provider.

3.2.3.9.3.1.3 Table 3.2.3-16 specifies the parameter that shall be passed when the primitive of the Provider-abort service is invoked.

Table 3.2.3-16: Provider-abort Service primitive Parameters

Parameter Name	Ind
Provider Abort Reason	M
Result Source	C

3.2.3.9.3.1.4 Provider Abort Reason

3.2.3.9.3.1.4.1 The *Provider Abort Reason* parameter shall be provided by the AIDC-AE service provider.

3.2.3.9.3.1.4.2 The *Provider Abort Reason* parameter shall conform to the ASN.1 abstract syntax *ProviderAbortReason*.

3.2.3.9.3.1.5 Result Source

3.2.3.9.3.1.5.1 The *Result Source* parameter shall optionally be provided by the AIDC-AE service provider, only when the *Provider Abort Reason* parameter has one of the abstract values “rejectedpermanent” or “rejectedtransient”.

3.2.3.9.3.1.5.2 The *Result Source* parameter shall conform to the abstract syntax of the A-ASSOCIATE *Result Source* parameter, as defined in ISO/IEC 8649.

3.2.4 THE AIDC-ASE ABSTRACT SERVICE

Note.— The following defines the abstract service interface used by an AIDC-ASE user to access the AIDC-ASE services and the services assumed to be supporting the AIDC-ASE.

3.2.4.1 Standard Parameters

Note.— The following service primitive parameters are defined here rather than repeated.

3.2.4.1.1 User Data

3.2.4.1.1.1 The *User Data* parameter, if provided, shall contain data provided by the user of the AIDC-ASE service.

Note.— The *User Data* parameter conforms to one of the following ASN.1 abstract syntaxes:

- a) *Notify*,
- b) *CoordinateInitial*,
- c) *CoordinateUpdate*,
- d) *CoordinateAccept*,
- e) *CoordinateReject*,
- f) *CoordinateNegotiate*,
- g) *CoordinateStandby*,
- h) *TransferInitiate*,
- i) *TransferRequest*,
- j) *TransferConditionsProposal*,
- k) *TransferConditionsAccept*,
- l) *TransferControl*,
- m) *TransferControlAssume*,
- n) *TransferControlReject*,
- o) *TransferCommunication*,
- p) *TransferCommunicationAssume*,
- q) *GeneralExecutiveData*,

- r) *GeneralPoint*,
- s) *SurveillanceData*,
- t) *GeneralFreeText*,
- u) *EmergencyFreeText*,
- v) *Cancel*.

3.2.4.1.2 Msg Number

3.2.4.1.2.1 The *Msg Number* parameter shall be provided by the AIDC-ASE user except when invoking the AIDC-usr-abort or AIDC-ucf request primitives.

3.2.4.1.2.2 The *Msg Number* parameter shall consist of a unique identifier for reference in the AIDC-ucf primitives.

Note.— The Msg Number parameter conforms to the ASN.1 abstract syntax MessageNumber.

3.2.4.2 AIDC-ASE Services

3.2.4.2.1 List of AIDC-ASE Services

3.2.4.2.1.1 An implementation of the AIDC-AE shall exhibit behaviour consistent with having implemented an AIDC-ASE with the following abstract services:

- a) AIDC-ucf service as defined in 3.2.4.2.2
- b) AIDC-nfy service as defined in 3.2.4.2.3
- c) AIDC-crd-start service as defined in 3.2.4.2.4
- d) AIDC-crd-end service as defined in 3.2.4.2.5
- e) AIDC-crd-ngtt service as defined in 3.2.4.2.6
- f) AIDC-crd-stndby service as defined in 3.2.4.2.7
- g) AIDC-tfr-init service as defined in 3.2.4.2.8
- h) AIDC-tfr-rqst service as defined in 3.2.4.2.9
- i) AIDC-tfr-prpsl service as defined in 3.2.4.2.10
- j) AIDC-tfr-accept service as defined in 3.2.4.2.11
- k) AIDC-tfr-cntrl service as defined in 3.2.4.2.12

- l) AIDC-tfr-comm service as defined in 3.2.4.2.13
- m) AIDC-tfr-comm-assm service as defined in 3.2.4.2.14
- n) AIDC-inf-tfr service as defined in 3.2.4.2.15
- o) AIDC-end service as defined in 3.2.4.2.16
- p) AIDC-usr-abrt service as defined in 3.2.4.2.17
- q) AIDC-pvd-abrt service as defined in 3.2.4.2.18.

3.2.4.2.2 AIDC-ucf Service

3.2.4.2.2.1 Service Primitives

3.2.4.2.2.1.1 The AIDC-ucf service shall be an unconfirmed service.

3.2.4.2.2.1.2 Table 3.2.4-1 specifies the parameters that shall be passed when the primitives of the AIDC-ucf service are invoked.

Table 3.2.4-1: AIDC-ucf Service Primitive Parameters

Parameter Name	Req	Ind
Result	M	M(=)
Reason	U	C(=)
Reference ID	M	M(=)

3.2.4.2.2.1.3 Result

3.2.4.2.2.1.3.1 The *Result* parameter shall be provided by the AIDC-ASE user.

3.2.4.2.2.1.3.2 The *Result* parameter shall indicate the acceptance or rejection of the service primitive.

Note.— The Result parameter conforms to the ASN.1 abstract syntax Result.

3.2.4.2.2.1.4 Reason

3.2.4.2.2.1.4.1 The *Reason* parameter shall be provided by the AIDC-ASE user.

Note.— The Reason parameter conforms to the ASN.1 abstract syntax ApplicationErrorData.

3.2.4.2.2.1.5 Reference ID

3.2.4.2.2.1.5.1 The *Reference ID* parameter shall be provided by the AIDC-ASE user.

3.2.4.2.2.1.5.2 The *Reference ID* parameter shall contain the Msg Number of the service that is being accepted or rejected.

Note.— The Reference ID parameter conforms to the ASN.1 abstract syntax MessageNumber.

3.2.4.2.3 AIDC-nfy Service

3.2.4.2.3.1 Service Primitives

3.2.4.2.3.1.1 The AIDC-nfy service shall be an unconfirmed service.

3.2.4.2.3.1.2 Table 3.2.4-2 specifies the parameters that shall be passed when the primitives of the AIDC-nfy service are invoked.

Table 3.2.4-2: AIDC-nfy Service Primitive Parameters

Parameter Name	Req	Ind
User Data	U	C(=)
Msg Number	M	M(=)

3.2.4.2.4 AIDC-crd-start Service.

3.2.4.2.4.1 Service Primitives

3.2.4.2.4.1.1 The AIDC-crd-start service shall be an unconfirmed service.

3.2.4.2.4.1.2 Table 3.2.4-3 specifies the parameters that shall be passed when the primitives of the AIDC-crd-start service are invoked.

Table 3.2.4-3: AIDC-crd-start Service Primitive Parameters

Parameter Name	Req	Ind
User Data	U	C(=)
Msg Number	M	M(=)

3.2.4.2.5 AIDC-crd-end Service

3.2.4.2.5.1 Service Primitives

3.2.4.2.5.1.1 The AIDC-crd-end service shall be an unconfirmed service.

3.2.4.2.5.1.2 Table 3.2.4-4 specifies the parameters that shall be passed when the primitives of the AIDC-crd-end service are invoked.

Table 3.2.4-4: AIDC-crd-end Service Primitive Parameters

Parameter Name	Req	Ind
User Data	U	C(=)
Result	M	M(=)
Msg Number	M	M(=)

3.2.4.2.5.1.3 Result

3.2.4.2.5.1.3.1 The *Result* parameter shall be provided by the AIDC-ASE user.

3.2.4.2.5.1.3.2 The *Result* parameter shall be used to indicate acceptance or rejection of the ending of the Coordinating Regime.

Note.— The Result parameter conforms to the ASN.1 abstract syntax Result.

3.2.4.2.6 AIDC-crd-ngtt Service

3.2.4.2.6.1 Service Primitives

3.2.4.2.6.1.1 The AIDC-crd-ngtt service shall be an unconfirmed service.

3.2.4.2.6.1.2 Table 3.2.4-5 specifies the parameters that shall be passed when the primitives of the AIDC-crd-ngtt service are invoked.

Table 3.2.4-5: AIDC-crd-ngtt Service Primitive Parameters

Parameter Name	Req	Ind
User Data	U	C(=)
Msg Number		

3.2.4.2.7 AIDC-crd-standby Service

3.2.4.2.7.1 Service Primitives

3.2.4.2.7.1.1 The AIDC-crd-standby service shall be an unconfirmed service.

3.2.4.2.7.1.2 Table 3.2.4-6 specifies the parameters that shall be passed when the primitives of the AIDC-crd-standby service are invoked.

Table 3.2.4-6: AIDC-crd-stndby Service Primitive Parameters

Parameter Name	Req	Ind
User Data	U	C(=)
Msg Number	M	M(=)

3.2.4.2.8 AIDC-tfr-init Service

3.2.4.2.8.1 Service Primitives

3.2.4.2.8.1.1 The AIDC-tfr-init service shall be an unconfirmed service.

3.2.4.2.8.1.2 Table 3.2.4-7 specifies the parameters that shall be passed when the primitives of the AIDC-tfr-init service are invoked.

Table 3.2.4-7: AIDC-tfr-init Service Primitive Parameters

Parameter Name	Req	Ind
User Data	U	C(=)
Msg Number	M	M(=)

3.2.4.2.9 AIDC-tfr-rqst Service

3.2.4.2.9.1 Service Primitives

3.2.4.2.9.1.1 The AIDC-tfr-rqst service shall be an unconfirmed service.

3.2.4.2.9.1.2 Table 3.2.4-8 specifies the parameters that shall be passed when the primitives of the AIDC-tfr-rqst service are invoked.

Table 3.2.4-8: AIDC-tfr-rqst Service Primitive Parameters

Parameter Name	Req	Ind
User Data	U	C(=)
Msg Number	M	M(=)

3.2.4.2.10 AIDC-tfr-prpsl Service

3.2.4.2.10.1 Service Primitives

3.2.4.2.10.1.1 The AIDC-tfr-prpsl service shall be an unconfirmed service.

3.2.4.2.10.1.2 Table 3.2.4-9 specifies the parameters that shall be passed when the primitives of the AIDC-tfr-prpsl service are invoked.

Table 3.2.4-9: AIDC-tfr-prpsl Service Primitive Parameters

Parameter Name	Req	Ind
User Data	U	C(=)
Msg Number	M	M(=)

3.2.4.2.11 AIDC-tfr-accept Service

3.2.4.2.11.1 Service Primitives

3.2.4.2.11.1.1 The AIDC-tfr-accept service shall be an unconfirmed service.

3.2.4.2.11.1.2 Table 3.2.4-10 specifies the parameters that shall be passed when the primitives of the AIDC-tfr-accept service are invoked.

Table 3.2.4-10: AIDC-tfr-accept Service Primitive Parameters

Parameter Name	Req	Ind
User Data	U	C(=)
Msg Number	M	M(=)

3.2.4.2.12 AIDC-tfr-cntrl Service

3.2.4.2.12.1 Service Primitives

3.2.4.2.12.1.1 The AIDC-tfr-cntrl service shall be a confirmed service.

3.2.4.2.12.1.2 Table 3.2.4-11 below specifies the parameters that shall be passed when the primitives of the AIDC-tfr-cntrl service are invoked.

Table 3.2.4-11: AIDC-tfr-cntrl Service Primitive Parameters

Parameter Name	Req	Ind	Rsp	Cnf
User Data	U	C(=)	U	C(=)
Result			M	M(=)
Msg Number	M	M(=)	M	M(=)

3.2.4.2.12.1.3 Result

3.2.4.2.12.1.3.1 The *Result* parameter, shall be provided by the AIDC-ASE user.

Note.— The Result parameter conforms to the ASN.1 abstract syntax Result.

3.2.4.2.13 AIDC-tfr-comm Service

3.2.4.2.13.1 Service Primitives

3.2.4.2.13.1.1 The AIDC-tfr-comm service shall be an unconfirmed service.

3.2.4.2.13.1.2 Table 3.2.4-12 specifies the parameters that shall be passed when the primitives of the AIDC-tfr-comm service are invoked.

Table 3.2.4-12: AIDC-tfr-comm Service Primitive Parameters

Parameter Name	Req	Ind
User Data	U	C(=)
Msg Number	M	M(=)

3.2.4.2.14 AIDC-tfr-comm-asm Service

3.2.4.2.14.1 Service Primitives

3.2.4.2.14.1.1 The AIDC-tfr-comm-asm service shall be an unconfirmed service.

3.2.4.2.14.1.2 Table 3.2.4-13 specifies the parameters that shall be passed when the primitives of the AIDC-tfr-comm-asm service are invoked.

Table 3.2.4-13: AIDC-tfr-comm-asm Service Primitive Parameters

Parameter Name	Req	Ind
User Data	U	C(=)
Msg Number	M	M(=)

3.2.4.2.15 AIDC-inf-tfr Service

3.2.4.2.15.1 Service Primitives

3.2.4.2.15.1.1 The AIDC-inf-tfr service shall be an unconfirmed service.

3.2.4.2.15.1.2 Table 3.2.4-14 specifies the parameters that shall be passed when the primitives of the AIDC-inf-tfr service are invoked.

Table 3.2.4-14: AIDC-inf-tfr Service Primitive Parameters

Parameter Name	Req	Ind
User Data	U	C(=)
Msg Number	M	M(=)

3.2.4.2.16 AIDC-end Service

3.2.4.2.16.1 Service Primitives

3.2.4.2.16.1.1 The AIDC-end service shall be an unconfirmed service.

3.2.4.2.16.1.2 Table 3.2.4-15 specifies the parameters that shall be passed when the primitives of the AIDC-end service are invoked.

Table 3.2.4-15: AIDC-end Service Primitive Parameters

Parameter Name	Req	Ind
User Data	U	C(=)
Msg Number	M	M(=)

3.2.4.2.17 AIDC-usr-abrt Service

3.2.4.2.17.1 Service Primitives

3.2.4.2.17.1.1 The AIDC-usr-abrt service shall be an unconfirmed service.

3.2.4.2.17.1.2 The AIDC-usr-abrt service primitives shall have no parameters.

3.2.4.2.18 AIDC-pvd-abrt Service

3.2.4.2.18.1 The AIDC-pvd-abrt service shall be an AIDC-ASE service-provider initiated service.

3.2.4.2.18.2 Service Primitive

3.2.4.2.18.2.1 The AIDC-pvd-abrt service shall be an unconfirmed service.

3.2.4.2.18.2.2 Table 3.2.4-16 specifies the parameter that shall be passed when the primitive of the AIDC-pvd-abrt service is invoked.

Table 3.2.4-16: AIDC-pvd-abrt Service Primitive Parameter

Parameter Name	Ind
Abort Reason	M

3.2.4.2.18.2.3 AbortReason

3.2.4.2.18.2.3.1 The *Abort Reason* parameter shall be provided by the AIDC-ASE.

3.2.4.2.18.2.3.2 The *Abort Reason* parameter shall be used to identify the reason for the abort.

Note.— The *Abort Reason* parameter conforms to the ASN.1 abstract syntax *ProviderAbortReason*.

3.2.4.3 Services Supporting the AIDC-ASE

Note.— The AIDC-ASE may be incorporated in any application entity that provides the following services.

3.2.4.3.1 List of Supporting Services

3.2.4.3.1.1 An implementation of the AIDC-AE shall exhibit the behaviour consistent with having implemented an AIDC-ASE supported by the following abstract services:

- a) AIDC-DATA service as defined in 3.2.4.3.2
- b) AIDC-ABORT service as defined in 3.2.4.3.3
- c) AIDC-P-ABORT service as defined in 3.2.4.3.4

3.2.4.3.2 AIDC-DATA Service

3.2.4.3.2.1 Service Primitives

3.2.4.3.2.1.1 The AIDC-DATA service shall be an unconfirmed service.

3.2.4.3.2.1.2 Table 3.2.4-17 specifies the parameter that shall be passed when the primitives of the AIDC-DATA service are invoked.

Table 3.2.4-17: AIDC-DATA Service Primitive Parameters

Parameter Name	Req	Ind
AIDC Data	U	C(=)

3.2.4.3.2.1.3 AIDC Data

3.2.4.3.2.1.3.1 The *AIDC Data* parameter, if any, shall be provided by the AIDC-ASE.

3.2.4.3.2.1.3.2 The *AIDC Data* parameter shall conform to the ASN.1 syntax *AIDC-APDU*.

3.2.4.3.3 AIDC-ABORT Service

3.2.4.3.3.1 Service Primitives

3.2.4.3.3.1.1 The AIDC-ABORT service shall be an unconfirmed service.

3.2.4.3.3.1.2 Table 3.2.4-18 specifies the parameter that shall be passed when the primitives of the AIDC-ABORT service primitives are invoked.

Table 3.1.4-18: AIDC-ABORT Service Primitive Parameters

Parameter Name	Req	Ind
Abort Reason	U	C(=)

3.2.4.3.3.2 Abort Reason

3.2.4.3.3.2.1 The *Abort Reason* parameter, if any, shall be provided by the AIDC-ASE.

3.2.4.3.3.2.2 The *Abort Reason* parameter shall conform to the ASN.1 syntax of *ProviderAbortReason*.

3.2.4.3.4 AIDC-P-ABORT Service

3.2.4.3.4.1 Service Primitives

3.2.4.3.4.1.1 The AIDC-P-ABORT service shall be an unconfirmed service.

3.2.4.3.4.1.2 The AIDC-P-ABORT service primitives shall have no parameters.

3.2.5 THE AIDC CONTROL FUNCTION

3.2.5.1 Recommendation. — *An implementation of the AIDC-AE should behave as if there existed a Control Function as defined below.*

Note.— *The following specifies the AIDC Control Function (CF) in terms of state definitions, and service mappings. The sequence diagrams for the various services are shown in 3.2.10.*

3.2.5.1.1 With the permissible exception of abort service primitives, the AIDC-AE shall process service primitives in the order in which they are received: this ensures that the AE will, with the exception of aborts, guarantee message sequencing.

3.2.5.2 AIDC-AE CF State Definitions

3.2.5.2.1 The AIDC-AE CF shall be in one of the following states at a given time:

- a) *Null (STA0)* – This is the state of the CF when there is no association in existence.
- b) *Association Pending (STA1)* – The CF enters this state when the AIDC-User has invoked a Notify or Coordinate-start request primitive or an indication has been received that the peer has made a request to establish an association.
- c) *Data Transfer (STA2)* – The CF enters this state once the establishment phase is complete. An association has successfully been established and the communicating partners are free to send and receive data.
- d) *Release Pending (STA3)* – The CF enters this state when either the AIDC-User has requested the termination of the AIDC service or an indication has been received that the peer has made a request to terminate the association.

3.2.5.3 AIDC-AE CF Service Mappings

Note.— *Figure 3.2.5-1 indicates which parts of this document specifies the behaviour of the CF in response to events at the various service interfaces.*

3.2.5.3.1 AIDC-User Services Primitives Submitted to the CF

Note.— *The following specifies the actions of the CF in response to events which occur at the upper service boundary of the AIDC-AE: specifically, request and response primitives generated by the AIDC-User.*

3.2.5.3.1.1 Implicit Association

3.2.5.3.1.1.1 The invocation of an Info-transfer request primitive, or a Notify request primitive or a Coordinate-start request primitive shall implicitly cause the establishment of an association, if one does not exist, between two peer AIDC-AEs.

Note.— *For a given flight, the handling of double associations between two peer AIDC-AEs is not managed by the AIDC-CF.*

3.2.5.3.1.1.2 The association establishment and release between peer AIDC-AEs shall be performed by invoking the primitives of ACSE.

3.2.5.3.1.1.3 Upon the receipt of an Info-transfer request primitive, a Notify request primitive or Coordinate-start request primitive, the CF shall:

- a) construct the Calling AP Title from the local ICAO Facility Designation according to the specification in 4.3.2;
- b) invoke an A-ASSOCIATE Request primitive with the parameters specified in Table 3.2.5-1; and
- c) enter the ASSOCIATION PENDING state.

Note.—To construct the Calling AP Title, the CF is assumed to have local knowledge of the ICAO Facility Designation of the AIDC-AE which it defines.

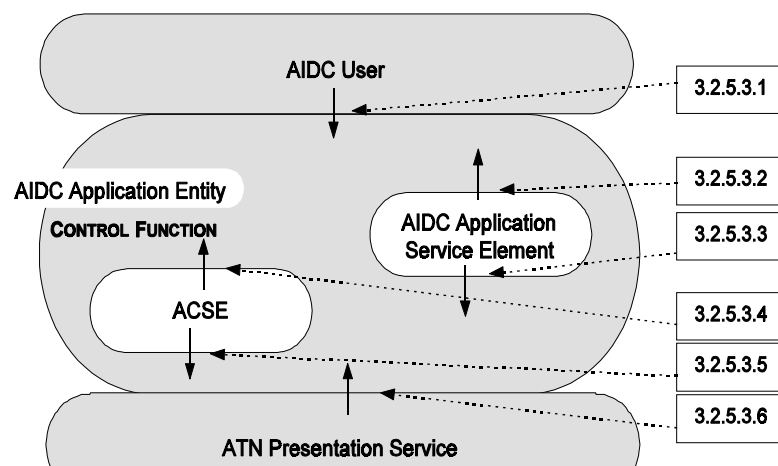


Figure 3.2.5-1: Elements of the AIDC-AE

Table 3.2.5-1: A-ASSOCIATE Request Primitive Parameters

A-ASSOCIATE Request Parameter	Value
Mode	Not used (default value)
Application Context Name	{iso (1) identified-organization (3) icao (27) atn-ac (3) 1 }
Application Context Name List	Not used
Calling AP Title	{iso (1) identified-organization (3) icao (27) atn-end-system-ground (2) <end-system-id> (n) operational (0) }
Calling AE Qualifier	idc (6)
Calling AP Invocation-identifier	Not used
Calling AE Invocation-identifier	Not used
Called AP Title	Not used
Called AE Qualifier	Not used
Called AP Invocation-identifier	Not used
Called AE Invocation-identifier	Not used
ACSE Requirements	Not used
Authentication-mechanism Name	Not used
Authentication-value	Not used
User Information	Not used
Calling Presentation Address	Local Implementation
Called Presentation Address	Local Implementation
Presentation Context Definition List	Not used
Default Presentation Context Name	Not used
Quality of Service	See 3.2.8.2
Presentation Requirements	Not used (default value)
Session Requirements	No Orderly Release (NOR), Duplex
Initial Synchronization Point Serial No	Not used
Initial Assignment of Tokens	Not used
Session-connection Identifier	Not used

3.2.5.3.1.2 User-confirmation Request Primitive

3.2.5.3.1.2.1 When Invoked

3.2.5.3.1.2.1.1 It shall be valid to invoke the User-confirmation request primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.1.2.1.2 **Recommendation.**— *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, and the User-confirmation request primitive is invoked, then appropriate error recovery action should be taken.*

3.2.5.3.1.2.2 Action Upon Invocation

3.2.5.3.1.2.2.1 The CF shall invoke an AIDC-ucf request primitive with:

- a) the User-confirmation *Result* parameter as the AIDC-ucf *Result* parameter;
- b) the User-confirmation *Referenced Number* parameter as the AIDC-ucf *Reference ID* parameter; and
- c) if present, the User-confirmation *Reason* parameter as the AIDC-ucf *Reason* parameter.

3.2.5.3.1.3 Notify Request Primitive

3.2.5.3.1.3.1 When Invoked

3.2.5.3.1.3.1.1 It shall be valid to invoke the Notify request primitive when the CF is in the NULL state, or the DATA TRANSFER state.

3.2.5.3.1.3.1.2 **Recommendation.**— *If the CF is in the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.1.3.2 Action Upon Invocation

3.2.5.3.1.3.2.1 If in the NULL state the CF shall:

- a) store the value of the Notify request primitive parameters;
- b) set the Boolean variable **nfy-assc**; and
- c) perform the implicit association function as specified in 3.2.5.3.1.1.

3.2.5.3.1.3.2.2 If in the DATA TRANSFER state, the CF shall invoke an AIDC-nfy request primitive with:

- a) the Notify request primitive *Notification Information* parameter as the AIDC-nfy request primitive *User Data* parameter; and
- b) the Notify request primitive *Message Number* parameter as the AIDC-nfy request primitive *Msg Number* parameter.

3.2.5.3.1.4 Coordinate-start Request Primitive

3.2.5.3.1.4.1 When Invoked

3.2.5.3.1.4.1.1 It shall be valid to invoke the Coordinate-start request primitive when the CF is in the NULL state, or the DATA TRANSFER state.

3.2.5.3.1.4.1.2 **Recommendation.** — *If the CF is in the ASSOCIATION PENDING state, or the RELEASE PENDING state then appropriate error recovery action should be taken.*

3.2.5.3.1.4.2 Action Upon Invocation

3.2.5.3.1.4.2.1 If in the NULL state the CF shall:

- a) store the value of the Coordinate-start request primitive parameters;
- b) set the Boolean variable **crd-assc**; and
- c) perform the implicit association function as specified in 3.2.5.3.1.1.

3.2.5.3.1.4.2.2 If in the DATA TRANSFER state, the CF shall invoke an AIDC-crd-start request primitive with:

- a) the Coordinate-start request primitive *Coordination Start Information* parameter as the AIDC-crd-start request primitive *User Data* parameter; and
- b) the Coordinate-start request primitive *Message Number* parameter as the AIDC-crd-start request primitive *Msg Number* parameter.

3.2.5.3.1.5 Coordinate-end Request Primitive

3.2.5.3.1.5.1 When Invoked

3.2.5.3.1.5.1.1 It shall be valid to invoke the Coordinate-end request primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.1.5.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.1.5.2 Action Upon Invocation

3.2.5.3.1.5.2.1 Upon the receipt of a Coordinate-end request primitive, the CF shall invoke an AIDC-crd-end request primitive with:

- a) the Coordinate-end request primitive *Coordinate End Information* parameter as the AIDC-crd-end request primitive *User Data* parameter;
- b) the Coordinate-end request primitive *Result* parameter as the AIDC-crd-end request primitive *Result* parameter; and

- c) the Coordinate-end request primitive *Message Number* parameter as the AIDC-crd-end request primitive *Msg Number* parameter.

3.2.5.3.1.6 Coordinate-negotiate Request Primitive

3.2.5.3.1.6.1 When Invoked

3.2.5.3.1.6.1.1 It shall be valid to invoke the Coordinate-negotiate request primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.1.6.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.1.6.2 Action Upon Invocation

3.2.5.3.1.6.2.1 Upon the receipt of a Coordinate-negotiate request primitive, the CF shall invoke an AIDC-crd-ngtt request primitive with:

- a) the Coordinate-negotiate request primitive *Coordinate Negotiate Information* parameter as the AIDC-crd-ngtt request primitive *User Data* parameter; and
- b) the Coordinate-negotiate request primitive *Message Number* parameter as the AIDC-crd-ngtt request primitive *Msg Number* parameter.

3.2.5.3.1.7 Coordinate-standby Request Primitive

3.2.5.3.1.7.1 When Invoked

3.2.5.3.1.7.1.1 It shall be valid to invoke the Coordinate-standby request primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.1.7.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.1.7.2 Action Upon Invocation

3.2.5.3.1.7.2.1 Upon the receipt of a Coordinate-standby request primitive, the CF shall invoke an AIDC-crd-stndby request primitive with:

- a) the Coordinate-standby request primitive *Coordinate Standby Information* parameter as the AIDC-crd-stndby request primitive *User Data* parameter; and
- b) the Coordinate-standby request primitive *Message Number* parameter as the AIDC-crd-stndby request primitive *Msg Number* parameter.

3.2.5.3.1.8 Transfer-initiate Request Primitive

3.2.5.3.1.8.1 When Invoked

3.2.5.3.1.8.1.1 It shall be valid to invoke the Transfer-initiate request primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.1.8.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.1.8.2 Action Upon Invocation

3.2.5.3.1.8.2.1 Upon the receipt of a Transfer-initiate request primitive, the CF shall invoke an AIDC-tfr-init request primitive with:

- a) the Transfer-initiate request primitive *Transfer Initiate Information* parameter as the AIDC-tfr-init request primitive *User Data* parameter; and
- b) the Transfer-initiate request primitive *Message Number* parameter as the AIDC-tfr-init request primitive *Msg Number* parameter.

3.2.5.3.1.9 Transfer-request Request Primitive

3.2.5.3.1.9.1 When Invoked

3.2.5.3.1.9.1.1 It shall be valid to invoke the Transfer-request request primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.1.9.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.1.9.2 Action Upon Invocation

3.2.5.3.1.9.2.1 Upon the receipt of a Transfer-request request primitive, the CF shall invoke an AIDC-tfr-rqst request primitive with:

- a) the Transfer-request request primitive *Transfer Request Information* parameter as the AIDC-tfr-rqst request primitive *User Data* parameter; and
- b) the Transfer-request request primitive *Message Number* parameter as the AIDC-tfr-rqst request primitive *Msg Number* parameter.

3.2.5.3.1.10 Transfer-conditions-proposal Request Primitive

3.2.5.3.1.10.1 When Invoked

3.2.5.3.1.10.1.1 It shall be valid to invoke the Transfer-conditions-proposal request primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.1.10.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.1.10.2 Action Upon Invocation

3.2.5.3.1.10.2.1 Upon the receipt of a Transfer-conditions-proposal request primitive, the CF shall invoke an AIDC-tfr-prpsl request primitive with:

- a) the Transfer-conditions-proposal request primitive *Transfer Conditions Proposal Information* parameter as the AIDC-tfr-prpsl request primitive *User Data* parameter; and
- b) the Transfer-conditions-proposal request primitive *Message Number* parameter as the AIDC-tfr-prpsl request primitive *Msg Number* parameter.

3.2.5.3.1.11 Transfer-conditions-accept Request Primitive

3.2.5.3.1.11.1 When Invoked

3.2.5.3.1.11.1.1 It shall be valid to invoke the Transfer-conditions-accept request primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.1.11.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.1.11.2 Action Upon Invocation

3.2.5.3.1.11.2.1 Upon the receipt of a Transfer-conditions-accept request primitive, the CF shall invoke an AIDC-tfr-accept request primitive with:

- a) the Transfer-conditions-accept request primitive *Transfer Conditions Accept Information* parameter as the AIDC-tfr-accept request primitive *User Data* parameter; and
- b) the Transfer-conditions-accept request primitive *Message Number* parameter as the AIDC-tfr-accept request primitive *Msg Number* parameter.

3.2.5.3.1.12 Transfer-control Request Primitive

3.2.5.3.1.12.1 When Invoked

3.2.5.3.1.12.1.1 It shall be valid to invoke the Transfer-control request primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.1.12.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.1.12.2 Action Upon Invocation

3.2.5.3.1.12.2.1 Upon the receipt of a Transfer-control request primitive, the CF shall invoke an AIDC-tfr-cntrl request primitive with:

- a) the Transfer-control request primitive *Transfer Control Information* parameter as the AIDC-tfr-cntrl request primitive *User Data* parameter; and
- b) the Transfer-control request primitive *Message Number* parameter as the AIDC-tfr-cntrl request primitive *Msg Number* parameter.

3.2.5.3.1.13 Transfer-control Response Primitive

3.2.5.3.1.13.1 When Invoked

3.2.5.3.1.13.1.1 It shall be valid to invoke the Transfer-control response primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.1.13.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.1.13.2 Action Upon Invocation

3.2.5.3.1.13.2.1 Upon the receipt of a Transfer-control response primitive, the CF shall invoke an AIDC-tfr-cntrl response primitive with:

- a) the Transfer-control response primitive *Transfer Control Information* parameter as the AIDC-tfr-cntrl response primitive *User Data* parameter;
- b) the Transfer-control response primitive *Result* parameter as the AIDC-tfr-cntrl response primitive *Result* parameter; and
- c) the Transfer-control response primitive *Message Number* parameter as the AIDC-tfr-cntrl response primitive *Msg Number* parameter.

3.2.5.3.1.14 Transfer-communication Request Primitive

3.2.5.3.1.14.1 When Invoked

3.2.5.3.1.14.1.1 It shall be valid to invoke the Transfer-communication request primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.1.14.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.1.14.2 Action Upon Invocation

3.2.5.3.1.14.2.1 Upon the receipt of a Transfer-communication request primitive, the CF shall invoke an AIDC-tfr-comm request primitive with:

- a) the Transfer-communication request primitive *Transfer Communication Information* parameter as the AIDC-tfr-comm request primitive *User Data* parameter; and
- b) the Transfer-communication request primitive *Message Number* parameter as the AIDC-tfr-comm request primitive *Msg Number* parameter.

3.2.5.3.1.15 Transfer-communication-assume Request Primitive

3.2.5.3.1.15.1 When Invoked

3.2.5.3.1.15.1.1 It shall be valid to invoke the Transfer-communication-assume request primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.1.15.1.2 **Recommendation.**— *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.1.15.2 Action Upon Invocation

3.2.5.3.1.15.2.1 Upon the receipt of a Transfer-communication-assume request primitive, the CF shall invoke an AIDC-tfr-comm-assm request primitive with:

- a) the Transfer-communication request primitive *Transfer Communication Assume Information* parameter as the AIDC-tfr-comm-assm request primitive *User Data* parameter; and
- b) the Transfer-communication request primitive *Message Number* parameter as the AIDC-tfr-comm-assm request primitive *Msg Number* parameter.

3.2.5.3.1.16 Info-transfer Request Primitive

3.2.5.3.1.16.1 When Invoked

3.2.5.3.1.16.1.1 It shall be valid to invoke the Info-transfer request primitive when the CF is in the NULL state, or the DATA TRANSFER state.

3.2.5.3.1.16.1.2 **Recommendation.**— *If the CF is in the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.1.16.2 Action Upon Invocation

3.2.5.3.1.16.2.1 If in the NULL state the CF shall:

- a) store the value of the Info-transfer request primitive parameter;

- b) set the Boolean variable **inf-assc**; and
- c) perform the implicit association function as specified in 3.2.5.3.1.1.

3.2.5.3.1.16.2.2 If in the DATA TRANSFER state, the CF shall invoke an AIDC-inf-tfr request primitive with:

- a) the Info-transfer request primitive *Information* parameter as the AIDC-inf-tfr request primitive *User Data* parameter; and
- b) the Info-transfer request primitive *Message Number* parameter as the AIDC-inf-tfr request primitive *Msg Number* parameter.

3.2.5.3.1.17 End Request Primitive

3.2.5.3.1.17.1 When Invoked

3.2.5.3.1.17.1.1 It shall be valid to invoke the End request primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.1.17.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.1.17.2 Action Upon Invocation

3.2.5.3.1.17.2.1 Upon the receipt of a End request primitive, the CF shall invoke an AIDC-end request primitive with:

- a) the End request primitive *Cancel Information* parameter, if present, as the AIDC-end request primitive *User Data* parameter; and
- b) the End request primitive *Message Number* parameter as the AIDC-end request primitive *Msg Number* parameter.

3.2.5.3.1.18 User-abort Request Primitive

3.2.5.3.1.18.1 When Invoked

3.2.5.3.1.18.1.1 It shall be valid to invoke the User-abort request primitive when the CF is in the ASSOCIATION PENDING state, or the DATA TRANSFER state, or the RELEASE PENDING state.

3.2.5.3.1.18.1.2 **Recommendation.** — *If the CF is in the NULL state, then appropriate error recovery action should be taken.*

3.2.5.3.1.18.2 Action Upon Invocation

3.2.5.3.1.18.2.1 Upon the receipt of a User-abort request primitive in either the DATA TRANSFER or the RELEASE PENDING state, the CF shall:

- a) invoke an AIDC-usr-abrt request primitive; and
- b) enter or remain in, the RELEASE PENDING state.

3.2.5.3.1.18.2.2 Upon the receipt of a User-abort request primitive in the ASSOCIATION PENDING state, the CF shall:

- a) invoke an A-ABORT request primitive; and
- b) enter the RELEASE PENDING state.

3.2.5.3.2 AIDC-ASE Service Primitives Delivered to the CF

Note.— The following specifies the actions of the CF in response to events which occur at the upper service boundary of the AIDC-ASE: specifically, indication and confirmation primitives which are generated by the AIDC-ASE protocol machine.

3.2.5.3.2.1 AIDC-ucf Indication Primitive

3.2.5.3.2.1.1 When Invoked

3.2.5.3.2.1.1.1 It shall be valid to invoke the AIDC-ucf indication primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.2.1.1.2 **Recommendation.**— *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.2.1.2 Action Upon Invocation

3.2.5.3.2.1.2.1 Upon the receipt of a AIDC-ucf indication primitive, the CF shall invoke a User-confirmation indication primitive with:

- a) the AIDC-ucf indication primitive *Result* parameter as the User-confirmation indication primitive *Result* parameter;
- b) if present, the AIDC-ucf indication primitive *Reason* parameter as the User-confirmation primitive *Reason* parameter; and
- c) the AIDC-ucf indication *Reference ID* parameter as the User-confirmation indication *Referenced Number* parameter.

3.2.5.3.2.2 AIDC-nfy Indication Primitive

3.2.5.3.2.2.1 When Invoked

3.2.5.3.2.2.1.1 It shall be valid to invoke the AIDC-nfy indication primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.2.2.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.2.2.2 Action Upon Invocation

3.2.5.3.2.2.2.1 Upon the receipt of a AIDC-nfy indication primitive, the CF shall invoke the Notify indication primitive with:

- a) the stored Calling ICAO Facility Designation as the Notify indication primitive *Calling ICAO Facility Designation* parameter;
- b) the AIDC-nfy indication primitive *User Data* parameter as the Notify indication primitive *Notification Information* parameter; and
- c) the AIDC-nfy indication primitive *Msg Number* parameter as the Notify indication primitive *Message Number* parameter.

3.2.5.3.2.3 AIDC-crd-start Indication Primitive

3.2.5.3.2.3.1 When Invoked

3.2.5.3.2.3.1.1 It shall be valid to invoke the AIDC-crd-start indication primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.2.3.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.2.3.2 Action Upon Invocation

3.2.5.3.2.3.2.1 Upon the receipt of a AIDC-crd-start indication primitive, the CF shall invoke the Coordinate-start indication primitive with:

- a) the stored Calling ICAO Facility Designation as the Coordinate-start indication primitive *Calling ICAO Facility Designation* parameter;
- b) the AIDC-crd-start indication primitive *User Data* parameter as the Coordinate-start indication primitive *Coordinate Start Information* parameter; and
- c) the AIDC-crd-start indication primitive *Msg Number* parameter as the Coordinate-start indication primitive *Message Number* parameter.

3.2.5.3.2.4 AIDC-crd-end Indication Primitive

3.2.5.3.2.4.1 When Invoked

3.2.5.3.2.4.1.1 It shall be valid to invoke the AIDC-crd-end indication primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.2.4.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.2.4.2 Action Upon Invocation

3.2.5.3.2.4.2.1 Upon the receipt of a AIDC-crd-end indication primitive the CF shall invoke a Coordinate-end indication primitive with:

- a) the AIDC-crd-end indication primitive *User Data* parameter as the Coordinate-end indication primitive *Coordinate End Information* parameter; and
- b) the AIDC-crd-end indication primitive *Msg Number* parameter as the Coordinate-end indication primitive *Message Number* parameter.

3.2.5.3.2.5 AIDC-crd-ngtt Indication Primitive

3.2.5.3.2.5.1 When Invoked

3.2.5.3.2.5.1.1 It shall be valid to invoke the AIDC-crd-ngtt indication primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.2.5.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.2.5.2 Action Upon Invocation

3.2.5.3.2.5.2.1 Upon the receipt of a AIDC-crd-ngtt indication primitive the CF shall invoke a Coordinate-negotiate indication primitive with:

- a) the AIDC-crd-ngtt indication primitive *User Data* parameter as the Coordinate-negotiate indication primitive *Coordinate Negotiate Information* parameter; and
- b) the AIDC-crd-ngtt indication primitive *Msg Number* parameter as the Coordinate-negotiate indication primitive *Message Number* parameter.

3.2.5.3.2.6 AIDC-crd-stndby Indication Primitive

3.2.5.3.2.6.1 When Invoked

3.2.5.3.2.6.1.1 It shall be valid to invoke the AIDC-crd-stndby indication primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.2.6.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.2.6.2 Action Upon Invocation

3.2.5.3.2.6.2.1 Upon the receipt of a AIDC-crd-standby indication primitive the CF shall invoke a Coordinate-standby indication primitive with:

- a) the AIDC-crd-standby indication primitive *User Data* parameter as the Coordinate-standby indication primitive *Coordinate Standby Information* parameter; and
- b) the AIDC-crd-standby indication primitive *Msg Number* parameter as the Coordinate-standby indication primitive *Message Number* parameter.

3.2.5.3.2.7 AIDC-tfr-init Indication Primitive

3.2.5.3.2.7.1 When Invoked

3.2.5.3.2.7.1.1 It shall be valid to invoke the AIDC-tfr-init indication primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.2.7.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.2.7.2 Action Upon Invocation

3.2.5.3.2.7.2.1 Upon the receipt of a AIDC-tfr-init indication primitive the CF shall invoke a Transfer-initiate indication primitive with:

- a) the AIDC-tfr-init indication primitive *User Data* parameter as the Transfer-initiate indication primitive *Transfer Initiate Information* parameter; and
- b) the AIDC-tfr-init indication primitive *Msg Number* parameter as the Transfer-initiate indication primitive *Message Number* parameter.

3.2.5.3.2.8 AIDC-tfr-rqst Indication Primitive

3.2.5.3.2.8.1 When Invoked

3.2.5.3.2.8.1.1 It shall be valid to invoke the AIDC-tfr-rqst indication primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.2.8.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.2.8.2 Action Upon Invocation

3.2.5.3.2.8.2.1 Upon the receipt of a AIDC-tfr-rqst indication primitive the CF shall invoke a Transfer-request indication primitive with:

- a) the AIDC-tfr-rqst indication primitive *User Data* parameter as the Transfer-request indication primitive *Transfer Request Information* parameter; and
- b) the AIDC-tfr-rqst indication primitive *Msg Number* parameter as the Transfer-request indication primitive *Message Number* parameter.

3.2.5.3.2.9 AIDC-tfr-prpsl Indication Primitive

3.2.5.3.2.9.1 When Invoked

3.2.5.3.2.9.1.1 It shall be valid to invoke the AIDC-tfr-prpsl indication primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.2.9.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.2.9.2 Action Upon Invocation

3.2.5.3.2.9.2.1 Upon the receipt of a AIDC-tfr-prpsl indication primitive the CF shall invoke a Transfer-conditions-proposal indication primitive with:

- a) the AIDC-tfr-prpsl indication primitive *User Data* parameter as the Transfer-conditions-proposal indication primitive *Transfer Conditions Proposal Information* parameter; and
- b) the AIDC-tfr-prpsl indication primitive *Msg Number* parameter as the Transfer-conditions-proposal indication primitive *Message Number* parameter.

3.2.5.3.2.10 AIDC-tfr-accept Indication Primitive

3.2.5.3.2.10.1 When Invoked

3.2.5.3.2.10.1.1 It shall be valid to invoke the AIDC-tfr-accept indication primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.2.10.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.2.10.2 Action Upon Invocation

3.2.5.3.2.10.2.1 Upon the receipt of a AIDC-tfr-accept indication primitive the CF shall invoke a Transfer-conditions-accept indication primitive with:

- a) the AIDC-tfr-accept indication primitive *User Data* parameter as the Transfer-conditions-accept indication primitive *Transfer Conditions Accept Information* parameter; and
- b) the AIDC-tfr-accept indication primitive *Msg Number* parameter as the Transfer-conditions-accept indication primitive *Message Number* parameter.

3.2.5.3.2.11 AIDC-tfr-cntrl Indication Primitive

3.2.5.3.2.11.1 When Invoked

3.2.5.3.2.11.1.1 It shall be valid to invoke the AIDC-tfr-cntrl indication primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.2.11.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.2.11.2 Action Upon Invocation

3.2.5.3.2.11.2.1 Upon the receipt of a AIDC-tfr-cntrl indication primitive the CF shall invoke a Transfer-control indication primitive with:

- a) the AIDC-tfr-cntrl indication primitive *User Data* parameter as the Transfer-control indication primitive *Transfer Control Information* parameter; and
- b) the AIDC-tfr-cntrl indication primitive *Msg Number* parameter as the Transfer-control indication primitive *Message Number* parameter.

3.2.5.3.2.12 AIDC-tfr-cntrl Confirmation Primitive

3.2.5.3.2.12.1 When Invoked

3.2.5.3.2.12.1.1 It shall be valid to invoke the AIDC-tfr-cntrl confirmation primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.2.12.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.2.12.2 Action Upon Invocation

3.2.5.3.2.12.2.1 Upon the receipt of a AIDC-tfr-cntrl confirmation primitive the CF shall invoke a Transfer-control confirmation primitive with:

- a) the AIDC-tfr-cntrl confirmation primitive *User Data* parameter as the Transfer-control confirmation primitive *Transfer Control Information* parameter;
- b) the AIDC-tfr-cntrl confirmation primitive *Result* parameter as the Transfer-control confirmation primitive *Result* parameter; and
- c) the AIDC-tfr-cntrl confirmation primitive *Msg Number* parameter as the Transfer-control confirmation primitive *Message Number* parameter.

3.2.5.3.2.13 AIDC-tfr-comm Indication Primitive

3.2.5.3.2.13.1 When Invoked

3.2.5.3.2.13.1.1 It shall be valid to invoke the AIDC-tfr-comm indication primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.2.13.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.2.13.2 Action Upon Invocation

3.2.5.3.2.13.2.1 Upon the receipt of a AIDC-tfr-comm indication primitive the CF shall invoke a Transfer-communication indication primitive with:

- a) the AIDC-tfr-comm indication primitive *User Data* parameter as the Transfer-communication indication primitive *Transfer Communication Information* parameter; and
- b) the AIDC-tfr-comm indication primitive *Msg Number* parameter as the Transfer-communication indication primitive *Message Number* parameter.

3.2.5.3.2.14 AIDC-tfr-comm-asm Indication Primitive

3.2.5.3.2.14.1 When Invoked

3.2.5.3.2.14.1.1 It shall be valid to invoke the AIDC-tfr-comm-asm indication primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.2.14.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.2.14.2 Action Upon Invocation

3.2.5.3.2.14.2.1 Upon the receipt of a AIDC-tfr-comm-asm indication primitive the CF shall invoke a Transfer-communication-assume indication primitive with:

- a) the AIDC-tfr-comm-asm indication primitive *User Data* parameter as the Transfer-communication-assume indication primitive *Transfer Communication Assume Information* parameter; and
- b) the AIDC-tfr-comm-asm indication primitive *Msg Number* parameter as the Transfer-communication-assume indication primitive *Message Number* parameter.

3.2.5.3.2.15 AIDC-inf-tfr Indication Primitive

3.2.5.3.2.15.1 When Invoked

3.2.5.3.2.15.1.1 It shall be valid to invoke the AIDC-inf-tfr indication primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.2.15.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.2.15.2 Action Upon Invocation

3.2.5.3.2.15.2.1 Upon the receipt of a AIDC-inf-tfr indication primitive the CF shall invoke a Info-transfer indication primitive with:

- a) the AIDC-inf-tfr indication primitive *User Data* parameter as the Info-transfer indication primitive *Information* parameter; and
- b) the AIDC-inf-tfr indication primitive *Msg Number* parameter as the Info-transfer indication primitive *Message Number* parameter.

3.2.5.3.2.16 AIDC-end Indication Primitive

3.2.5.3.2.16.1 When Invoked

3.2.5.3.2.16.1.1 It shall be valid to invoke the AIDC-end indication primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.2.16.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.2.16.2 Action Upon Invocation

3.2.5.3.2.16.2.1 Upon the receipt of a AIDC-end indication primitive the CF shall invoke a End indication primitive with:

- a) the AIDC-end indication primitive *User Data* parameter, if present, as the End indication primitive *Cancel Information* parameter; and
- b) the AIDC-end indication primitive *Msg Number* parameter as the End indication primitive *Message Number* parameter.

3.2.5.3.2.17 AIDC-usr-abrt Indication Primitive

3.2.5.3.2.17.1 When Invoked

3.2.5.3.2.17.1.1 It shall be valid to invoke the AIDC-usr-abrt indication primitive when the CF is in the ASSOCIATION PENDING state, or the DATA TRANSFER state, or the RELEASE PENDING state.

3.2.5.3.2.17.1.2 **Recommendation.** — *If the CF is in the NULL state, then appropriate error recovery action should be taken.*

3.2.5.3.2.17.2 Action Upon Invocation

3.2.5.3.2.17.2.1 Upon the receipt of a AIDC-usr-abrt indication primitive the CF shall invoke a User-abort indication primitive and enter the NULL state.

3.2.5.3.2.18 AIDC-pvd-abrt Indication Primitive

3.2.5.3.2.18.1 When Invoked

3.2.5.3.2.18.1.1 It shall be valid to invoke the AIDC-pvd-abrt indication primitive when the CF is in the DATA TRANSFER state, or the RELEASE PENDING state.

3.2.5.3.2.18.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.2.18.2 Action Upon Invocation

3.2.5.3.2.18.2.1 Upon the receipt of a AIDC-pvd-abrt indication primitive the CF shall:

- a) if the CF is in the DATA TRANSFER state:
 - 1) invoke a Provider-abort indication primitive with the Provider-abort *Provider Abort Reason* parameter having the value delivered in the AIDC-pvd-abrt *Abort Reason* parameter; and
 - 2) enter the RELEASE PENDING state.

- b) if the CF is in the RELEASE PENDING state invoke a Provider-abort indication primitive with the Provider-abort *Provider Abort Reason* parameter having the abstract value delivered in the AIDC-pvd-abrt *Abort Reason* parameter.

3.2.5.3.3 AIDC-ASE Service Primitives Submitted to the CF

Note.— The following specifies the actions of the CF in response to events which occur at the lower service boundary of the AIDC-ASE: specifically, request primitives which are generated by the AIDC-ASE protocol machine.

3.2.5.3.3.1 AIDC-DATA Request primitive

3.2.5.3.3.1.1 When Invoked

3.2.5.3.3.1.1.1 It shall be valid to invoke the AIDC-DATA request primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.3.1.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.3.1.2 Action Upon Invocation

3.2.5.3.3.1.2.1 Upon the receipt of a AIDC-DATA request primitive the CF shall:

- a) encode the AIDC-DATA Request *User Data* parameter using the definition of presentation-user-data in 4.3.2.6; and
- b) invoke a P-DATA request primitive with the resulting encoding from a) above, as the *User Data* parameter.

3.2.5.3.3.2 AIDC-ABORT Request Primitive

3.2.5.3.3.2.1 When Invoked

3.2.5.3.3.2.1.1 It shall be valid to invoke the AIDC-ABORT request primitive when the CF is in the DATA TRANSFER state, or the RELEASE PENDING state.

3.2.5.3.3.2.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.3.2.2 Action Upon Invocation

3.2.5.3.3.2.2.1 Upon the receipt of a AIDC-ABORT request primitive the CF shall:

- a) if the CF is in the DATA TRANSFER state:
 - 1) invoke an A-ABORT request primitive with parameters as follows:
 - i) if no AIDC-ABORT *Abort Reason* parameter is present, the A-ABORT *AbortSource* parameter set to abstract value “acse-service-user” and the A-ABORT *Diagnostic* value set to the abstract value “no-reason-given”;
 - ii) if the AIDC-ABORT *Abort Reason* parameter has either one of the abstract values “protocolerror” or “timerexpired”, the A-ABORT *AbortSource* parameter set to abstract value “acse-service-user” and the A-ABORT *Diagnostic* value set to the abstract value “protocol-error”;
 - iii) otherwise, the A-ABORT *AbortSource* parameter set to the abstract value “acse-service-user” and no A-ABORT *Diagnostic* value; and
 - 2) enter the RELEASE PENDING state;
- b) if the CF is in the RELEASE PENDING state, invoke an A-ABORT request primitive with no parameters.

3.2.5.3.4 ACSE Service Primitives Delivered to the CF

Note.— The following specifies the action of the CF in response to events which occur at the upper service boundary of ACSE: specifically, indication and confirmation primitives which are generated by the ACSE Protocol Machine (ACPM).

3.2.5.3.4.1 A-ASSOCIATE Indication Primitive

3.2.5.3.4.1.1 When Invoked

3.2.5.3.4.1.1.1 It shall be valid to invoke the A-ASSOCIATE indication primitive when the CF is in the ASSOCIATION PENDING state.

3.2.5.3.4.1.1.2 **Recommendation.** — *If the CF is in the NULL state or the DATA TRANSFER state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.4.1.2 Action Upon Invocation

3.2.5.3.4.1.2.1 Upon the receipt of an A-ASSOCIATE indication primitive, the CF shall:

- a) extract and store the encoded Calling ICAO Facility Designation from *Calling AP Title* parameter; and
- b) invoke an A-ASSOCIATE response primitive with the parameters as shown in Table 3.2.5-2.

Table 3.2.5-2: A-ASSOCIATE Response Primitive Parameters

Parameter	Value
Mode	Not used (default value)
Application Context Name	{ iso (1) identified-organization (3) icao (27) atn-ac (3) 1 }
Application Context Name List	Not used
Responding AP Title	Not used
Responding AE Qualifier	Not used
Responding AP Invocation-identifier	Not used
Responding AE Invocation-identifier	Not used
ACSE Requirements	Not used
Authentication-mechanism Name	Not used
Authentication-value	Not used
User Information	Not used
Result	Not used
Diagnostic	Not used
Responding Presentation Address	Local Implementation
Presentation Context Definition Result List	Not used
Default Presentation Context Result	Not used
Quality of Service	Not used
Presentation Requirements	Not used (default value)
Session Requirements	No Orderly Release (NOR), Duplex
Initial Synchronization Point Serial No	Not used

Parameter	Value
Initial Assignment of Tokens	Not used
Session-connection Identifier	Not used

3.2.5.3.4.2 A-ASSOCIATE Confirmation Primitive

3.2.5.3.4.2.1 When Invoked

3.2.5.3.4.2.1.1 It shall be valid to invoke the A-ASSOCIATE confirmation primitive when the CF is in the ASSOCIATION PENDING state.

3.2.5.3.4.2.1.2 **Recommendation.** — *If the CF is in the NULL state, or the DATA TRANSFER state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.4.2.2 Action Upon Invocation

3.2.5.3.4.2.2.1 Upon the receipt of an A-ASSOCIATE confirmation primitive, the CF shall:

- a) if the *Result* parameter has the abstract value “accepted”, then:
 - 1) if the **nfy-assc** variable is set:- invoke an AIDC-nfy request primitive with:
 - i) the stored *Notification Information* as the *User Data* parameter;
 - ii) the stored *Message Number* parameter as the *Msg Number* parameter; and
 - iii) enter the DATA TRANSFER state; or
 - 2) if the **crd-assc** variable is set:- invoke an AIDC-crd-start request primitive with:
 - i) the stored *Coordinate Start Information* as the *User Data* parameter;
 - ii) the stored *Message Number* parameter as the *Msg Number* parameter; and
 - iii) enter the DATA TRANSFER state.
 - 3) if the **inf-assc** variable is set:- invoke an AIDC-inf-tfr request primitive with:
 - i) the stored *Info-transfer Information* as the *User Data* parameter;
 - ii) the stored *Message Number* parameter as the *Msg Number* parameter; and

- iii) enter the DATA TRANSFER state.
- b) if the *Result* parameter has the abstract value “rejected (permanent)” or “rejected (transient)”, then:
 - 1) invoke a Provider-abort indication primitive with the A-ASSOCIATE confirmation primitive *Result Source* parameter as the Provider-abort indication primitive *Provider Abort Reason* parameter set to the abstract value “rejectedpermanent” or “rejectedtransient” corresponding to the value of the A-ASSOCIATE indication *Result* parameter; and
 - 2) enter the NULL state.

3.2.5.3.4.3 A-ABORT Indication primitive

3.2.5.3.4.3.1 When Invoked

3.2.5.3.4.3.1.1 It shall be valid to invoke the A-ABORT indication primitive when the CF is in the ASSOCIATION PENDING state, or the DATA TRANSFER state, or the RELEASE PENDING state.

3.2.5.3.4.3.1.2 **Recommendation.** — *If the CF is in the NULL state, then appropriate error recovery action should be taken.*

3.2.5.3.4.3.2 Action Upon Invocation

3.2.5.3.4.3.2.1 Upon the receipt of an A-ABORT indication primitive, the CF shall:

- a) if the CF is in the ASSOCIATION PENDING state, or the DATA TRANSFER state:
 - 1) invoke a Provider-abort indication primitive as follows:
 - i) in each case, ignore any delivered A-ABORT *UserInformation* parameter value;
 - ii) if the A-ABORT *AbortSource* parameter has the abstract value “acse-service-user” and the A-ABORT *Diagnostic* parameter has the abstract value “protocol-error”, set the *Provider Abort Reason* parameter to the abstract value “protocolerror”;
 - iii) if the A-ABORT *AbortSource* parameter has the abstract value “acse-service-provider”, set the *Provider Abort Reason* parameter to the abstract value “providererror”;
 - iv) otherwise, set the *Provider Abort Reason* parameter to the abstract value “undefinederror”; and
 - 2) enter the NULL state.

- b) if the CF is in the DATA TRANSFER state:
 - 1) invoke an AIDC-ABORT indication primitive as follows:
 - i) in each case, ignore any delivered A-ABORT *UserInformation* parameter value;
 - ii) if the A-ABORT *AbortSource* parameter has the abstract value “acse-service-user” and the A-ABORT *Diagnostic* parameter has the abstract value “no-reason-given”, omit the AIDC-ABORT *AbortReason* parameter;
 - iii) if the A-ABORT *Diagnostic* parameter has the abstract value “protocol-error”, set the AIDC-ABORT *AbortReason* parameter to the abstract value “protocolerror”;
 - iv) otherwise, set the AIDC-ABORT *Abort Reason* parameter to the abstract value “undefinederror”; and
 - 2) enter the RELEASE PENDING state.
- c) if the CF is in the RELEASE PENDING state, enter the NULL state.

3.2.5.3.4.4 A-P-ABORT Indication Primitive

3.2.5.3.4.4.1 When Invoked

3.2.5.3.4.4.1.1 It shall be valid to invoke the A-P-ABORT indication primitive when the CF is in the ASSOCIATION PENDING state, or the DATA TRANSFER state, or the RELEASE PENDING state.

3.2.5.3.4.4.1.2 **Recommendation.** — *If the CF is in the NULL state, then appropriate error recovery action should be taken.*

3.2.5.3.4.4.2 Action Upon Invocation

3.2.5.3.4.4.2.1 Upon the receipt of an A-P-ABORT indication primitive, the CF shall:

- a) if in the ASSOCIATION PENDING state:
 - 1) invoke a Provider-abort indication with abstract value “providererror” as the value of the Provider-abort indication primitive *Provider Abort Reason* parameter, and discard any *ProviderReason* parameter in the A-P-ABORT indication; and
 - 2) enter the NULL state.

- b) if in the DATA TRANSFER state, or the RELEASE PENDING state:
 - 1) invoke an AIDC-P-ABORT indication primitive, and discard any *Provider Reason* parameter in the A-P-ABORT indication; and
 - 2) enter the RELEASE PENDING state.

3.2.5.3.5 ACSE Service Primitives Submitted to the CF

Note 1.— The following specifies the actions of the CF in response to events at the lower service boundary of ACSE: specifically, request and response primitives generated by the ACPM.

Note 2.— ACSE (Edition 2) mandates the mapping between ACSE and the underlying Presentation service provider. Invocations of Presentation service primitives by ACSE are “intercepted” by the CF and re-mapped to the “actual” Presentation service as appropriate.

3.2.5.3.5.1 P-CONNECT Request Primitive

3.2.5.3.5.1.1 When Invoked

3.2.5.3.5.1.1.1 It shall be valid to invoke the P-CONNECT request primitive when the CF is in the ASSOCIATION PENDING state.

3.2.5.3.5.1.1.2 **Recommendation.** — *If the CF is in the NULL state, or the DATA TRANSFER state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.5.1.2 Action Upon Invocation

3.2.5.3.5.1.2.1 Upon the receipt of a P-CONNECT request primitive, the CF shall invoke the equivalent Presentation service primitive of the ATN service provider.

3.2.5.3.5.2 P-CONNECT Response Primitive

3.2.5.3.5.2.1 When Invoked

3.2.5.3.5.2.1.1 It shall be valid to invoke the P-CONNECT response primitive when the CF is in the ASSOCIATION PENDING state.

3.2.5.3.5.2.1.2 **Recommendation.** — *If the CF is in the DATA TRANSFER state, or the RELEASE PENDING state, or the NULL state, then appropriate error recovery action should be taken.*

3.2.5.3.5.2.2 Action Upon Invocation

3.2.5.3.5.2.2.1 Upon the receipt of a P-CONNECT response primitive accepting the proposed connection, the CF shall:

- a) transparently invoke the equivalent presentation service primitive; and
- b) enter the DATA TRANSFER state.

3.2.5.3.5.2.2.2 Upon the receipt of a P-CONNECT response primitive rejecting the proposed connection, the CF shall:

- a) transparently invoke the equivalent presentation service primitive; and
- b) enter the NULL state.

3.2.5.3.5.3 P-U-ABORT Request Primitive

3.2.5.3.5.3.1 When Invoked

3.2.5.3.5.3.1.1 It shall be valid to invoke the P-U-ABORT request primitive when the CF is in the ASSOCIATION PENDING state, or the DATA TRANSFER state, or the RELEASE PENDING state.

3.2.5.3.5.3.1.2 **Recommendation.** — *If the CF is in the NULL state, then appropriate error recovery action should be taken.*

3.2.5.3.5.3.2 Action Upon Invocation

3.2.5.3.5.3.2.1 Upon the receipt of a P-U-ABORT request primitive, the CF shall:

- a) if the P-U-Abort request user data parameter is present, and the CF is in the DATA TRANSFER state:
 - 1) encode the presentation user data as indicated in 4.3.2.6 with the P-U-ABORT user data parameter (an ACSE ABRT APDU) as the presentation data value and presentation context identifier value corresponding to “acse-apdu”; and
 - 2) invoke a P-DATA Request primitive with the resulting encoding as the value of the UserData parameter;
- b) otherwise, invoke a P-U-ABORT request primitive without any parameters;
- c) in either case, enter the NULL state.

Note.— The invocation of the P-U-ABORT request primitive will abort the connection to the underlying ATN Service Provider.

3.2.5.3.6 Presentation Service Primitives Delivered to the CF

Note 1.— The following specifies the actions of the CF in response to events which occur at the lower service boundary of the AIDC-AE: specifically, indication and confirmation primitives which are delivered by the Presentation service.

Note 2.— ACSE (Edition 2) mandates the mapping between ACSE and the underlying Presentation service provider. Presentation service primitives are “intercepted” by the CF.

3.2.5.3.6.1 P-CONNECT Indication Primitive

3.2.5.3.6.1.1 When Invoked

3.2.5.3.6.1.1.1 It shall be valid to invoke the P-CONNECT indication primitive when the CF is in the NULL state.

3.2.5.3.6.1.1.2 **Recommendation.** — *If the CF is in the ASSOCIATION PENDING state, or the DATA TRANSFER state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.6.1.2 Action Upon Invocation

3.2.5.3.6.1.2.1 Upon the receipt of a P-CONNECT indication primitive, the CF shall:

- a) enter the ASSOCIATION PENDING state; and
- b) invoke the equivalent Presentation service primitive at the lower ACSE service boundary.

3.2.5.3.6.2 P-CONNECT Confirmation Primitive

3.2.5.3.6.2.1 When Invoked

3.2.5.3.6.2.1.1 It shall be valid to invoke the P-CONNECT confirmation primitive when the CF is in the ASSOCIATION PENDING state.

3.2.5.3.6.2.1.2 **Recommendation.** — *If the CF is in the NULL state, or the DATA TRANSFER state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.6.2.2 Action Upon Invocation

3.2.5.3.6.2.2.1 Upon the receipt of a P-CONNECT confirmation primitive, the CF shall invoke the equivalent Presentation service primitive at the lower ACSE service boundary.

3.2.5.3.6.3 P-U-ABORT Indication Primitive

3.2.5.3.6.3.1 When Invoked

3.2.5.3.6.3.1.1 It shall be valid to invoke the P-U-ABORT indication primitive when the CF is in the ASSOCIATION PENDING state, or the DATA TRANSFER state, or the RELEASE PENDING state.

3.2.5.3.6.3.1.2 **Recommendation.** — *If the CF is in the NULL state, then appropriate error recovery action should be taken.*

3.2.5.3.6.3.2 Action Upon Invocation

3.2.5.3.6.3.2.1 Upon the receipt of a P-U-ABORT indication primitive, the CF shall invoke the equivalent Presentation service primitive at the lower ACSE service boundary.

3.2.5.3.6.4 P-P-ABORT Indication Primitive

3.2.5.3.6.4.1 When Invoked

3.2.5.3.6.4.1.1 It shall be valid to invoke the P-P-ABORT indication primitive when the CF is in the ASSOCIATION PENDING state, or the DATA TRANSFER state, or the RELEASE PENDING state.

3.2.5.3.6.4.1.2 **Recommendation.** — *If the CF is in the NULL state, then appropriate error recovery action should be taken.*

3.2.5.3.6.4.2 Action Upon Invocation

3.2.5.3.6.4.2.1 Upon the receipt of a P-P-ABORT indication primitive, the CF shall invoke the corresponding Presentation service primitive at the lower ACSE service boundary.

3.2.5.3.6.5 P-DATA Indication Primitive

3.2.5.3.6.5.1 When Invoked

3.2.5.3.6.5.1.1 It shall be valid to invoke the P-DATA indication primitive when the CF is in the DATA TRANSFER state.

3.2.5.3.6.5.1.2 **Recommendation.** — *If the CF is in the NULL state, or the ASSOCIATION PENDING state, or the RELEASE PENDING state, then appropriate error recovery action should be taken.*

3.2.5.3.6.5.2 Action Upon Invocation

3.2.5.3.6.5.2.1 Upon the receipt of a P-DATA indication primitive, the CF shall:

- a) decode the presentation-user-data as indicated in 4.3.2.6 to determine the destination ASE of the APDU, and extract the Presentation data value.
- b) if the presentation context identifier has the abstract value “user-ase”, invoke an AIDC-DATA indication primitive with the extracted AIDC-APDU as the AIDC-DATA *AIDC Data* parameter;
- c) if the presentation context identifier has the abstract value “acse-ase” and the ACSE-APDU has the syntax *ABRT-apdu*, invoke a P-U-Abort indication primitive with the *UserData* parameter containing the received APDU.

Note.— *ABRT is the only ACSE APDU that is transmitted using P-DAT by the AIDC CF.*

3.2.5.4 AIDC-CF State Table

3.2.5.4.1 The AIDC-AE shall behave as if it has a CF which functions in accordance with the following state table.

Note.— Table 3.2.5-3 shows the state transitions and actions performed by the AIDC-CF in response to service primitives submitted to the AIDC-CF. The source of the service primitive invocation is shown in column one of the Table 3.2.5-3 and the service primitive invocations are shown in the second column.

3.2.5.4.2 In the event of discrepancies between the state Table 3.2.5-3 and the text above, the text shall take precedence.

3.2.5.4.3 Each cell in the state Table 3.2.5-3 shows:

- a) the action, if any, which the CF shall perform; and
- b) the new state that the CF shall enter.

3.2.5.4.4 Blank cells shall indicate error conditions.

3.2.5.4.4.1 The error handling shall result in the association being aborted, if one exists.

3.2.5.4.4.2 **Recommendation.**— *The AIDC-User should be notified when an association is aborted.*

Table 3.2.5-3: AIDC CF State Table

Source ↓	State → Event ↓	STA0 NULL	STA1 ASSOCIATION PENDING	STA2 DATA TRANSFER	STA3 RELEASE PENDING
From AIDC-User	Notify Req	- Store Notification Information - set nfy-assc - A-ASSOCIATE Req ⇒STA1		AIDC-nfy Req ⇒STA2	
	Coordinate-start Req	- Store Coordination Start Information - set crd-assc - A-ASSOCIATE Req ⇒STA1		AIDC-crd-start Req ⇒STA2	
	Info-transfer Req	- Store Info-transfer information - set inf-assc - A-ASSOCIATE Req ⇒STA1		AIDC-inf-tfr Req ⇒STA2	
	End Req			- set end = Message Number - AIDC-end Req ⇒STA2	

Source ↓	State → Event ↓	STA0 NULL	STA1 ASSOCIATION PENDING	STA2 DATA TRANSFER	STA3 RELEASE PENDING
	User-abort Req		• A-ABORT Req ⇒STA3	• AIDC-usr-abrt Req ⇒STA3	• AIDC-usr-abrt Req ⇒STA3
	All other Req			• equivalent AIDC-ASE Req ⇒STA2	
From AIDC-ASE (upper)	AIDC-ucf Ind			• User- confirmation Ind ⇒STA2	
	AIDC-usr-abrt Ind		• User-abort Ind ⇒STA0	• User-abort Ind ⇒STA0	• User-abort Ind ⇒STA0
	AIDC-pvd-abrt Ind			• Provider-abort Ind ⇒STA3	• Provider-abort Ind ⇒STA3
	All other Ind			• equivalent AIDC-User service invocation ⇒STA2	
From AIDC-ASE (lower)	AIDC-DATA Req			• P-DATA Req (APDU) ⇒STA2	
	AIDC-ABORT Req			• A-ABORT Req ⇒STA3	• A-ABORT Req ⇒STA3
From ACSE (upper)	A-ASSOCIATE Ind		• A-ASSOCIATE Rsp+ ⇒STA1		
	A-ASSOCIATE Cnf+		if nfy-assc • AIDC-nfy Req or if crd-assc • AIDC-crd-start Req or if inf-assc • AIDC-inf-tfr Req ⇒STA2		
	A-ASSOCIATE Cnf-		• Provider-abort Ind ⇒STA0		
	A-ABORT Ind		• Provider-abort Ind ⇒STA0	• AIDC-ABORT Ind ⇒STA3	⇒STA0
	A-P-ABORT Ind		• Provider-abort Ind ⇒STA0	• AIDC-P- ABORT Ind ⇒STA3	• AIDC-P- ABORT Ind ⇒STA3

Source ↓	State → Event ↓	STA0 NULL	STA1 ASSOCIATION PENDING	STA2 DATA TRANSFER	STA3 RELEASE PENDING
From ACSE (lower)	P-CONNECT Req		• P-CONNECT Req ⇒STA1		
	P-CONNECT Rsp+		• P-CONNECT Rsp+ ⇒STA2		
	P-CONNECT Rsp-		• P-CONNECT Rsp- ⇒STA0		
	P-U-ABORT Req		• P-U-ABORT Req ⇒STA0	• P-DATA Req with ACSE ABRT APDU as UserData and “acse-ase” as presentation context identifier value ⇒STA0	• P-U-ABORT Req ⇒STA0
From ATN Service Provider	P-CONNECT Ind	• P-CONNECT Ind ⇒STA1			
	P-CONNECT Cnf+/-		• P-CONNECT Cnf ⇒STA1		
	P-DATA Ind			- if presentation context identifier = “user-ase” { • P-DATA Ind ⇒ STA2} - if presentation context identifier = “acse-ase” { • P-U-Abort Ind with ABRT APDU as UserData ⇒ STA2}	
	P-U-ABORT Ind		• P-U-ABORT Ind ⇒STA 1	• P-U-ABORT Ind ⇒STA2	• P-U-ABORT Ind ⇒STA3
	P-P-ABORT Ind		• P-P-ABORT Ind ⇒STA1	• P-P-ABORT Ind ⇒STA2	• P-P-ABORT Ind ⇒STA3

3.2.6 THE AIDC-ASE PROTOCOL DEFINITION

Note.— The following specifies the AIDC-ASE protocol.

3.2.6.1 AIDC-ASE Protocol Description

3.2.6.1.1 Only if requirements are described for an AIDC-ASE primitive when an AIDC-ASE is in a particular state, shall the invocation of that primitive be permitted while the AIDC-ASE is in that state.

3.2.6.1.2 If no requirements are described for the arrival of an APDU when the AIDC-ASE is in a particular state, then exception handling procedures as described in 3.2.6.2 shall apply.

3.2.6.1.3 **Recommendation.**— *Appropriate error recovery action should be taken, upon the arrival of an unexpected APDU.*

3.2.6.1.4 Predicates

Note.— The AIDC-ASE protocol has one predicate, P1, defined. This predicate conditions the use of the AIDC-tfr-accept service.

3.2.6.1.4.1 When the predicate P1 is set true, then the AIDC-tfr-accept service shall be used in response to the AIDC-tfr-prpsl service.

3.2.6.1.4.2 When the predicate P1 is set false, the invocation of the AIDC-tfr-accept service shall be prohibited.

3.2.6.1.5 AIDC-nfy Request Primitive

3.2.6.1.5.1 When Invoked

3.2.6.1.5.1.1 It shall be valid to invoke the AIDC-nfy request primitive when the AIDC-ASE protocol machine is in the IDLE state, or the NOTIFY state.

3.2.6.1.5.2 Action Upon Invocation

3.2.6.1.5.2.1 Upon the receipt of an AIDC-nfy request primitive the AIDC-ASE shall:

a) If in the IDLE state or the NOTIFY state:

- 1) create an AIDC-nfy-apdu based on the *User Data* parameter, and the *Msg Number* parameter;
- 2) invoke an AIDC-DATA request primitive with the AIDC-nfy-apdu as the *AIDC Data* parameter;
- 3) if running, stop timer t_{2IN} ;
- 4) if in the NOTIFY state, then stop the timer t_{2NC} ;

- 5) start the timer t_C ; and
 - 6) set the variable $vs_1 = \text{notify}$ and $vs_2 = \text{Msg Number}$.
- b) If the AIDC-ASE protocol machine is any other state:
- 1) invoke an AIDC-pvd-abrt indication primitive with the *AbortReason* parameter set to the abstract value: “protocolerror”;
 - 2) invoke an AIDC-ABORT request primitive with the *AbortReason* set to abstract value “protocolerror”; and
 - 3) enter the IDLE state.

3.2.6.1.6 AIDC-DATA Indication Primitive with an AIDC-nfy-apdu

3.2.6.1.6.1 When Invoked

3.2.6.1.6.1.1 It shall be valid to invoke the AIDC-DATA indication primitive with an AIDC-nfy-apdu when the AIDC-ASE protocol machine is in the IDLE state, or the NOTIFY state.

3.2.6.1.6.2 Action Upon Invocation

3.2.6.1.6.2.1 Upon the receipt of an AIDC-DATA indication primitive with an AIDC-nfy-apdu the AIDC-ASE shall:

- a) extract the *User Data* parameter, and the *Msg Number* parameter from the AIDC-nfy-apdu;
- b) invoke an AIDC-nfy indication primitive with the extracted parameters in a) above as the AIDC-nfy indication primitive *User Data* parameter, and the AIDC-nfy indication *Msg Number* parameter;
- c) if running stop the timer t_{1IN} ;
- d) if running, stop the timer t_{2R} ;
- e) if in the NOTIFY state, then stop the timer t_{1NC} ; and
- f) set the variable $vr_1 = \text{notify}$ and $vr_2 = \text{Msg Number}$.

3.2.6.1.7 AIDC-crd-start Request Primitive

3.2.6.1.7.1 When Invoked

3.2.6.1.7.1.1 It shall be valid to invoke the AIDC-crd-start request primitive when the AIDC-ASE protocol machine is in the IDLE state, or the NOTIFY state, or the COORDINATED state, or the TRANSFERRED state.

3.2.6.1.7.2 Action Upon Invocation

3.2.6.1.7.2.1 Upon the receipt of an AIDC-crd-start request primitive the AIDC-ASE shall:

- a) create an AIDC-crd-start-apdu based on the *User Data* parameter, and the *Msg Number* parameter;
- b) invoke an AIDC-DATA request primitive with the AIDC-crd-start-apdu as the *AIDC Data* parameter;
- c) if running, stop timer t_{2IN} ;
- d) if in the NOTIFY state, stop timer T_{1CT} ;
- e) if in the COORDINATED state, then:
 - 1) if running, stop timer t_{1CT} ;
 - 2) if running, stop timer t_{2CT} ;
- f) if in the TRANSFERRED state, then stop the timer t_{TE} ;
- g) start the timer t_C ; and
- h) set the variable $vs_1 = \text{back}$ and $vs_2 = \text{Msg Number}$.

3.2.6.1.8 AIDC-DATA Indication Primitive with an AIDC-crd-start-apdu

3.2.6.1.8.1 When Invoked

3.2.6.1.8.1.1 It shall be valid to invoke the AIDC-DATA indication primitive with an AIDC-crd-start-apdu when the AIDC-ASE protocol machine is in the IDLE state, or the NOTIFY state, or the COORDINATED state, or the TRANSFERRED state.

3.2.6.1.8.2 Action Upon Invocation

3.2.6.1.8.2.1 Upon the receipt of a AIDC-DATA indication primitive with an AIDC-crd-start-apdu the AIDC ASE shall:

- a) extract the *User Data* parameter, and the *Msg Number* parameter from the AIDC-crd-start-apdu;
- b) invoke an AIDC-crd-start indication primitive with the extracted parameters in a) above as the AIDC-crd-start indication *Msg Number* parameter and the AIDC-crd-start indication primitive *User Data* parameter;
- c) if running, stop timer t_{1IN} ;
- d) if running, stop the timer t_{2R} ;

- e) if in the NOTIFY state, then stop the timer t_{INC} ;
- f) if in the COORDINATED state, then:
 - 1) if running, stop timer t_{2CT} ;
 - 2) if running, stop the timer t_{1CT} ;
- g) if in the TRANSFERRED state, then stop the timer t_{TE} ; and
- h) set the variable $vr_1 = \text{back}$ and $vr_2 = \text{Msg Number}$.

3.2.6.1.9 AIDC-crd-end Request Primitive

3.2.6.1.9.1 When Invoked

3.2.6.1.9.1.1 It shall be valid to invoke the AIDC-crd-end request primitive when the AIDC-ASE protocol machine is in the NEGOTIATING state, or the RE-NEGOTIATING state, or the BACKWARD COORDINATING state.

3.2.6.1.9.2 Action Upon Invocation

3.2.6.1.9.2.1 Upon the receipt of an AIDC-crd-end request primitive the AIDC-ASE shall:

- a) create an AIDC-crd-end-apdu based on the *User Data* parameter, the *Result* parameter and the *Msg Number* parameter;
- b) invoke a AIDC-DATA request primitive with the AIDC-crd-end-apdu as the *AIDC Data* parameter;
- c) start the timer t_C ; and
- d) if in the NEGOTIATING state, then
 - 1) set the variable $vs_1 = \text{coord-end}$;
 - 2) if the AIDC-crd-end *Result* parameter has the value “accept” then set the variable $vse = \text{accept}$; and
 - 3) if the AIDC-crd-end *Result* parameter has the value “reject” then set the variable $vse = \text{reject}$.
- e) if in the RE-NEGOTIATING state, set the variable $vs_1 = \text{coord-end}$;
- f) if in the BACKWARD COORDINATING state, set the variable $vs_1 = \text{back-end}$; and
- g) set the variable $vs_2 = \text{Msg Number}$.

3.2.6.1.10 AIDC-DATA Indication Primitive with an AIDC-crd-end-apdu

3.2.6.1.10.1 When Invoked

3.2.6.1.10.1.1 It shall be valid to invoke the AIDC-DATA indication primitive with an AIDC-crd-end-apdu when the AIDC-ASE protocol machine is in the NEGOTIATING state, or the RE-NEGOTIATING state, or the BACKWARD COORDINATING state.

3.2.6.1.10.2 Action Upon Invocation

3.2.6.1.10.2.1 Upon the receipt of an AIDC-DATA indication primitive with an AIDC-crd-end-apdu the AIDC-ASE shall:

- a) extract the *User Data* parameter, the *Result* parameter and the *Msg Number* parameter from the AIDC-crd-end-apdu;
- b) invoke an AIDC-crd-end indication primitive with the extracted parameters in a) above as the AIDC-crd-end indication *User Data* parameter, the AIDC-crd-end indication *Result* parameter and the AIDC-crd-end indication primitive *Msg Number* parameter respectively;
- c) if running, stop the timer t_{1R} ;
- d) if running, stop the timer t_{2R} ;
- e) if running, stop the timer t_s ; and
- f) if in the NEGOTIATING state, then:
 - 1) set the variable $vr_1 = \text{coord-end}$;
 - 2) if the AIDC-crd-end *Result* parameter has the value “accept” then set the variable $vre = \text{accept}$; and
- g) if in the RE-NEGOTIATING state, set the variable $vr_1 = \text{coord-end}$;
- h) if in the BACKWARD COORDINATING state, set the variable $vr_1 = \text{back-end}$; and
- i) set the variable $vr_2 = \text{Msg Number}$.

3.2.6.1.11 AIDC-crd-ngtt Request Primitive

3.2.6.1.11.1 When Invoked

3.2.6.1.11.1.1 It shall be valid to invoke the AIDC-crd-ngtt request primitive when the AIDC-ASE protocol machine is in the NEGOTIATING state, or the RE-NEGOTIATING, or the BACKWARD-COORDINATING state.

3.2.6.1.11.2 Action Upon Invocation

3.2.6.1.11.2.1 Upon the receipt of an AIDC-crd-ngtt request primitive the AIDC-ASE shall:

- a) create an AIDC-crd-ngtt-apdu based on the *User Data* parameter, and the *Msg Number* parameter;
- b) invoke a AIDC-DATA request primitive with the AIDC-crd-ngtt-apdu as the *AIDC Data* parameter
- c) start the timer t_C ; and
- d) set the variable $vs_1 = \text{coord-negot}$, and $vs_2 = \text{Msg Number}$.

3.2.6.1.12 AIDC-DATA Indication Primitive with an AIDC-crd-ngtt-apdu

3.2.6.1.12.1 When Invoked

3.2.6.1.12.1.1 It shall be valid to invoke the AIDC-DATA indication primitive with an AIDC-crd-ngtt-apdu when the AIDC-ASE protocol machine is in the NEGOTIATING state, or the RE-NEGOTIATING state, or the BACKWARD-COORDINATING state.

3.2.6.1.12.2 Action Upon Invocation

3.2.6.1.12.2.1 Upon the receipt of an AIDC-DATA indication primitive with an AIDC-crd-ngtt-apdu the AIDC-ASE shall:

- a) extract the *User Data* parameter, and the *Msg Number* parameter from the AIDC-crd-ngtt-apdu; and
- b) invoke an AIDC-crd-ngtt indication primitive with the extracted parameters in a) above as the AIDC-crd-ngtt indication *User Data* parameter, and the AIDC-crd-ngtt indication primitive *Msg Number* parameter respectively;
- c) if running, stop the timer t_{1R} ;
- d) if running, stop the timer t_{2R} ;
- e) if running, stop the timer t_S ; and
- f) set the variable $vr_1 = \text{coord-negot}$ and $vr_2 = \text{Msg Number}$.

3.2.6.1.13 AIDC-crd-standby Request Primitive

3.2.6.1.13.1 When Invoked

3.2.6.1.13.1.1 It shall be valid to invoke the AIDC-crd-standby request primitive when the AIDC-ASE protocol machine is in the NEGOTIATING state, or the RE-NEGOTIATING state, or the BACKWARD-COORDINATING state.

3.2.6.1.13.2 Action Upon Invocation

3.2.6.1.13.2.1 Upon the receipt of an AIDC-crd-standby request primitive the AIDC-ASE shall:

- a) create an AIDC-crd-standby-apdu based on the *User Data* parameter, and the *Msg Number* parameter;
- b) invoke a AIDC-DATA request primitive with the AIDC-crd-standby-apdu as the *AIDC Data* parameter;
- c) start the timer t_C ; and
- d) set the variable $vs_1 = \text{coord-standby}$ and $vs_2 = \text{Msg Number}$.

3.2.6.1.14 AIDC-DATA Indication Primitive with an AIDC-crd-standby-apdu

3.2.6.1.14.1 When Invoked

3.2.6.1.14.1.1 It shall be valid to invoke the AIDC-DATA indication primitive with an AIDC-crd-standby-apdu when the AIDC-ASE protocol machine is in the NEGOTIATING state, or the RE-NEGOTIATING state, or the BACKWARD-COORDINATING state.

3.2.6.1.14.2 Action Upon Invocation

3.2.6.1.14.2.1 Upon the receipt of an AIDC-DATA indication primitive with an AIDC-crd-standby-apdu the AIDC-ASE shall:

- a) extract the *User Data* parameter, and the *Msg Number* parameter from the AIDC-crd-standby-apdu;
- b) invoke an AIDC-crd-standby indication primitive with the extracted parameters in a) above as the AIDC-crd-standby indication *User Data* parameter, and the AIDC-crd-standby indication primitive *Msg Number* parameter respectively;
- c) if running, stop the timer t_{1R} ;
- d) if running, stop the timer t_{2R} ; and
- e) set the variable $vr_1 = \text{coord-standby}$ and $vr_2 = \text{Msg Number}$.

3.2.6.1.15 AIDC-tfr-init Request Primitive

3.2.6.1.15.1 When Invoked

3.2.6.1.15.1.1 It shall be valid to invoke the AIDC-tfr-init request primitive when the AIDC-ASE protocol machine is in the COORDINATED state.

3.2.6.1.15.2 Action Upon Invocation

3.2.6.1.15.2.1 Upon the receipt of an AIDC-tfr-init request primitive the AIDC-ASE shall:

- a) create an AIDC-tfr-init-apdu based on the *User Data* parameter, and the *Msg Number* parameter;
- b) invoke a AIDC-DATA request primitive with the AIDC-tfr-init-apdu as the *AIDC Data* parameter;
- c) if running, stop the timer t_{1CT} ;
- d) if running, stop the timer t_{2CT} ;
- e) start the timer t_C ; and
- f) set the variable $vs_1 = \text{trns-init}$, and $vs_2 = \text{Msg Number}$.

3.2.6.1.16 AIDC-DATA Indication Primitive with an AIDC-tfr-init-apdu

3.2.6.1.16.1 When Invoked

3.2.6.1.16.1.1 It shall be valid to invoke the AIDC-DATA indication primitive with an AIDC-tfr-init-apdu when the AIDC-ASE protocol machine is in the COORDINATED state.

3.2.6.1.16.2 Action Upon Invocation

3.2.6.1.16.2.1 Upon the receipt of an AIDC-DATA indication primitive with an AIDC-tfr-init-apdu the AIDC-ASE shall:

- a) extract the *User Data* parameter, and the *Msg Number* parameter from the AIDC-tfr-init-apdu;
- b) invoke an AIDC-tfr-init indication primitive with the extracted parameters in a) above as the AIDC-tfr-init indication *User Data* parameter, and the AIDC-tfr-init indication primitive *Msg Number* parameter respectively;
- c) if running, stop the timer t_{1CT} ;
- d) if running, stop the timer t_{2CT} ; and
- e) set the variable $vr_1 = \text{trns-init}$ and $vr_2 = \text{Msg Number}$.

3.2.6.1.17 AIDC-tfr-rqst Request Primitive

3.2.6.1.17.1 When Invoked

3.2.6.1.17.1.1 It shall be valid to invoke the AIDC-tfr-rqst request primitive when the AIDC-ASE protocol machine is in the COORDINATED state.

3.2.6.1.17.2 Action Upon Invocation

3.2.6.1.17.2.1 Upon the receipt of an AIDC-tfr-rqst request primitive the AIDC-ASE shall:

- a) create an AIDC-tfr-rqst-apdu based on the *User Data* parameter, and the *Msg Number* parameter;
- b) invoke a AIDC-DATA request primitive with the AIDC-tfr-rqst-apdu as the *AIDC Data* parameter;
- c) start the timer t_c ; and
- d) set the variable $vs_2 = \text{Msg Number}$.

3.2.6.1.18 AIDC-DATA Indication Primitive with an AIDC-tfr-rqst-apdu

3.2.6.1.18.1 When Invoked

3.2.6.1.18.1.1 It shall be valid to invoke the AIDC-DATA indication primitive with an AIDC-tfr-rqst-apdu when the AIDC-ASE protocol machine is in the COORDINATED state.

3.2.6.1.18.2 Action Upon Invocation

3.2.6.1.18.2.1 Upon the receipt of an AIDC-DATA indication primitive with an AIDC-tfr-rqst-apdu the AIDC-ASE shall:

- a) extract the *User Data* parameter, and the *Msg Number* parameter from the AIDC-tfr-rqst-apdu;
- b) invoke an AIDC-tfr-rqst indication primitive with the extracted parameters in a) above as the AIDC-tfr-rqst indication *User Data* parameter, and the AIDC-tfr-rqst indication primitive *Msg Number* parameter respectively; and
- c) set the variable $vr_2 = \text{Msg Number}$.

3.2.6.1.19 AIDC-tfr-prpsl Request Primitive

3.2.6.1.19.1 When Invoked

3.2.6.1.19.1.1 It shall be valid to invoke the AIDC-tfr-prpsl request primitive when the AIDC-ASE protocol machine is in the PRE-TRANSFER state.

3.2.6.1.19.2 Action Upon Invocation

3.2.6.1.19.2.1 Upon the receipt of an AIDC-tfr-prpsl request primitive the AIDC-ASE shall:

- a) create an AIDC-tfr-prpsl-apdu based on the *User Data* parameter, and the *Msg Number* parameter;
- b) invoke a AIDC-DATA request primitive with the AIDC-tfr-prpsl-apdu as the *AIDC Data* parameter;
- c) start the timer t_C ;
- d) if P1, then set the variable $vs_1 = \text{trns-prpsl}$; and
- e) set the variable $vs_2 = \text{Msg Number}$.

3.2.6.1.20 AIDC-DATA Indication Primitive with an AIDC-tfr-prpsl-apdu

3.2.6.1.20.1 When Invoked

3.2.6.1.20.1.1 It shall be valid to invoke the AIDC-DATA indication primitive with an AIDC-tfr-prpsl-apdu when the AIDC-ASE protocol machine is in the PRE-TRANSFER state.

3.2.6.1.20.2 Action Upon Invocation

3.2.6.1.20.2.1 Upon the receipt of an AIDC-DATA indication primitive with an AIDC-tfr-prpsl-apdu the AIDC-ASE shall:

- a) extract the *User Data* parameter, and the *Msg Number* parameter from the AIDC-tfr-prpsl-apdu;
- b) invoke an AIDC-tfr-prpsl indication primitive with the extracted parameters in a) above as the AIDC-tfr-prpsl indication *User Data* parameter, and the AIDC-tfr-prpsl indication primitive *Msg Number* parameter respectively;
- c) if P1, then set the variable $vr_1 = \text{trns-prpsl}$; and
- d) set the variable $vr_2 = \text{Msg Number}$.

3.2.6.1.21 AIDC-tfr-accept Request Primitive

3.2.6.1.21.1 When Invoked

3.2.6.1.21.1.1 It shall be valid to invoke the AIDC-tfr-accept request primitive if the predicate P1 is true and when the AIDC-ASE protocol machine is in the PRE-TRANSFER state.

3.2.6.1.21.2 Action Upon Invocation

3.2.6.1.21.2.1 Upon the receipt of an AIDC-tfr-accept request primitive the AIDC-ASE shall:

- a) create an AIDC-tfr-accept-apdu based on the *User Data* parameter, and the *Msg Number* parameter;
- b) invoke a AIDC-DATA request primitive with the AIDC-tfr-accept-apdu as the *AIDC Data* parameter;
- c) start the timer t_C ; and

- d) set the variable $vs_2 = \text{Msg Number}$.

3.2.6.1.22 AIDC-DATA Indication Primitive with an AIDC-tfr-accept-apdu

3.2.6.1.22.1 When Invoked

3.2.6.1.22.1.1 It shall be valid to invoke the AIDC-DATA indication primitive with an AIDC-tfr-accept-apdu if the predicate P1 is true and when the AIDC-ASE protocol machine is in the PRE-TRANSFER state.

3.2.6.1.22.2 Action Upon Invocation

3.2.6.1.22.2.1 Upon the receipt of an AIDC-DATA indication primitive with an AIDC-tfr-accept-apdu the AIDC-ASE shall:

- a) if running stop timer t_{1R} ;
- b) extract the *User Data* parameter, and the *Msg Number* parameter from the AIDC-tfr-accept-apdu;
- c) invoke an AIDC-tfr-accept indication primitive with the extracted parameters in a) above as the AIDC-tfr-accept indication *User Data* parameter, and the AIDC-tfr-accept indication primitive *Msg Number* parameter respectively; and
- d) set the variable $vr_2 = \text{Msg Number}$.

3.2.6.1.23 AIDC-tfr-comm Request Primitive

3.2.6.1.23.1 When Invoked

3.2.6.1.23.1.1 It shall be valid to invoke the AIDC-tfr-comm request primitive when the AIDC-ASE protocol machine is in the PRE-TRANSFER state.

3.2.6.1.23.2 Action Upon Invocation

3.2.6.1.23.2.1 Upon the receipt of an AIDC-tfr-comm request primitive the AIDC-ASE shall:

- a) create an AIDC-tfr-comm-apdu based on the *User Data* parameter, and the *Msg Number* parameter;
- b) invoke a AIDC-DATA request primitive with the AIDC-tfr-comm-apdu as the *AIDC Data* parameter;
- c) start the timer t_C ; and
- d) set the variable $vs_1 = \text{trns-comm}$ and $vs_2 = \text{Msg Number}$.

3.2.6.1.24 AIDC-DATA Indication Primitive with an AIDC-tfr-comm-apdu

3.2.6.1.24.1 When Invoked

3.2.6.1.24.1.1 It shall be valid to invoke the AIDC-DATA indication primitive with an AIDC-tfr-comm-apdu when the AIDC-ASE protocol machine is in the PRE-TRANSFER state.

3.2.6.1.24.2 Action Upon Invocation

3.2.6.1.24.2.1 Upon the receipt of an AIDC-DATA indication primitive with an AIDC-tfr-comm-apdu the AIDC-ASE shall:

- a) extract the *User Data* parameter, and the *Msg Number* parameter from the AIDC-tfr-comm-apdu;
- b) invoke an AIDC-tfr-comm indication primitive with the extracted parameters in a) above as the AIDC-tfr-comm indication *User Data* parameter, and the AIDC-tfr-comm indication primitive *Msg Number* parameter respectively;
- c) if running, stop the timer t_{2R} ;
- d) if running, stop the timer t_{3R} ; and
- e) set the variable $vr_1 = \text{trns-comm}$ and $vr_2 = \text{Msg Number}$.

3.2.6.1.25 AIDC-tfr-comm-asm Request Primitive

3.2.6.1.25.1 When Invoked

3.2.6.1.25.1.1 It shall be valid to invoke the AIDC-tfr-comm-asm request primitive when the AIDC-ASE protocol machine is in the PRE-TRANSFER state, or the TRANSFERRING state.

3.2.6.1.25.2 Action Upon Invocation

3.2.6.1.25.2.1 Upon the receipt of an AIDC-tfr-comm-asm request primitive the AIDC-ASE shall:

- a) create an AIDC-tfr-comm-asm-apdu based on the *User Data* parameter, and the *Msg Number* parameter;
- b) invoke a AIDC-DATA request primitive with the AIDC-tfr-comm-asm-apdu as the *AIDC Data* parameter;
- c) start the timer t_C ;
- d) if running, stop the timer t_{3R} ; and
- e) set the variable $vs_1 = \text{trns-asm}$ and $vs_2 = \text{Msg Number}$.

3.2.6.1.26 AIDC-DATA Indication Primitive with an AIDC-tfr-comm-asm-apdu

3.2.6.1.26.1 When Invoked

3.2.6.1.26.1.1 It shall be valid to invoke the AIDC-DATA indication primitive with an AIDC-tfr-comm-asm-apdu when the AIDC-ASE protocol machine is in the PRE-TRANSFER state, or the TRANSFERRING state.

3.2.6.1.26.2 Action Upon Invocation

3.2.6.1.26.2.1 Upon the receipt of an AIDC-DATA indication primitive with an AIDC-tfr-comm-asm-apdu the AIDC-ASE shall:

- a) extract the *User Data* parameter, and the *Msg Number* parameter from the AIDC-tfr-comm-asm-apdu;

- b) invoke an AIDC-tfr-comm-asm indication primitive with the extracted parameters in a) above as the AIDC-tfr-comm-asm indication *User Data* parameter, and the AIDC-tfr-comm-asm indication primitive *Msg Number* parameter respectively;
- c) if running, stop the timer t_{1R} ;
- d) if running, stop the timer t_{2R} ; and
- e) set the variable $vr_1 = \text{trns-asm}$ and $vr_2 = \text{Msg Number}$.

3.2.6.1.27 AIDC-tfr-cntrl Request Primitive

3.2.6.1.27.1 When Invoked

3.2.6.1.27.1.1 It shall be valid to invoke the AIDC-tfr-cntrl request primitive when the AIDC-ASE protocol machine is in the COORDINATED state.

3.2.6.1.27.2 Action Upon Invocation

3.2.6.1.27.2.1 Upon the receipt of an AIDC-tfr-cntrl request primitive the AIDC-ASE shall:

- a) create an AIDC-tfr-cntrl-req-apdu based on the *User Data* parameter, and the *Msg Number* parameter;
- b) invoke a AIDC-DATA request primitive with the AIDC-tfr-cntrl-req-apdu as the *AIDC Data* parameter;
- c) if running, stop the timer t_{1CT} ;
- d) if running, stop the timer t_{2CT} ;
- e) start the timer t_C ; and
- f) set the variable $vs_1 = \text{trns-start}$ $vs_2 = \text{Msg Number}$.

3.2.6.1.28 AIDC-DATA Indication Primitive with an AIDC-tfr-cntrl-req-apdu

3.2.6.1.28.1 When Invoked

3.2.6.1.28.1.1 It shall be valid to invoke the AIDC-DATA indication primitive with an AIDC-tfr-cntrl-Req-apdu when the AIDC-ASE protocol machine is in the COORDINATED state.

3.2.6.1.28.2 Action Upon Invocation

3.2.6.1.28.2.1 Upon the receipt of an AIDC-DATA indication primitive with an AIDC-tfr-cntrl-req-apdu the AIDC-ASE shall:

- a) extract the *User Data* parameter, and the *Msg Number* parameter from the AIDC-tfr-cntrl-req-apdu;
- b) invoke an AIDC-tfr-cntrl indication primitive with the extracted parameters in a) above as the AIDC-tfr-cntrl indication *User Data* parameter, and the AIDC-tfr-cntrl indication primitive *Msg Number* parameter respectively;
- c) if running, stop the timer t_{1CT} ;
- d) if running, stop the timer t_{2CT} ;

- e) if running, stop the timer t_{2R} ; and
- f) set the variable $vr_1 = \text{trns-start}$ and $vr_2 = \text{Msg Number}$.

3.2.6.1.29 AIDC-tfr-cntrl Response Primitive

3.2.6.1.29.1 When Invoked

3.2.6.1.29.1.1 It shall be valid to invoke the AIDC-tfr-cntrl response primitive when the AIDC-ASE protocol machine is in the TRANSFERRING state.

3.2.6.1.29.2 Action Upon Invocation

3.2.6.1.29.2.1 Upon the receipt of an AIDC-tfr-cntrl response primitive the AIDC-ASE shall:

- a) create an AIDC-tfr-cntrl-rsp-apdu based on the *User Data* parameter, the *Result* parameter and the *Msg Number* parameter;
- b) invoke a AIDC-DATA request primitive with the AIDC-tfr-cntrl-rsp-apdu as the *AIDC Data* parameter;
- c) start the timer t_C ;
- d) if the *Result* parameter has the abstract value:
 - 1) “accepted”; set $vs_1 = \text{trns-accept}$;
 - 2) “rejected”; set $vs_1 = \text{trns-reject}$.
- e) set the variable $vs_2 = \text{Msg Number}$;

3.2.6.1.30 AIDC-DATA Indication with an AIDC-tfr-cntrl-rsp-apdu

3.2.6.1.30.1 When Invoked

3.2.6.1.30.1.1 It shall be valid to invoke the AIDC-DATA indication primitive with an AIDC-tfr-cntrl-rsp-apdu when the AIDC-ASE protocol machine is in the TRANSFERRING state.

3.2.6.1.30.2 Action Upon Invocation

3.2.6.1.30.2.1 Upon the receipt of an AIDC-DATA indication primitive with an AIDC-tfr-cntrl-rsp-apdu the AIDC-ASE shall:

- a) extract the *User Data* parameter, the *Result* parameter, and the *Msg Number* parameter from the AIDC-tfr-cntrl-rsp-apdu;
- b) invoke an AIDC-tfr-cntrl confirmation primitive with the extracted parameters in a) above as the AIDC-tfr-cntrl confirmation *User Data* parameter, and the AIDC-tfr-cntrl indication primitive *Msg Number* parameter respectively;
- c) if running, stop the timer t_{1R} ;
- d) if running, stop the timer t_{2R} ;
- e) if the *Result* parameter has the abstract value:
 - 1) “accepted”; set $vr_1 = \text{trns-accept}$;

2) “rejected”; set $vr_1 = \text{trns-reject}$; and

f) set the variable $vr_2 = \text{Msg Number}$.

3.2.6.1.31 AIDC-inf-tfr Request Primitive

3.2.6.1.31.1 When Invoked

3.2.6.1.31.1.1 It shall be valid to invoke the AIDC-inf-tfr request primitive when the AIDC-ASE protocol machine is in the IDLE state, or in the NOTIFY state, or the NEGOTIATING state, or the RE-NEGOTIATING state, or the COORDINATED state, or the PRE-TRANSFER state, or the TRANSFERRING state, or the TRANSFERRED state, or the BACKWARD-COORDINATING state.

3.2.6.1.31.2 Action Upon Invocation

3.2.6.1.31.2.1 Upon the receipt of an AIDC-inf-tfr request primitive the AIDC-ASE shall:

- a) create an AIDC-inf-tfr-apdu based on the *User Data* parameter, and the *Msg Number* parameter;
- b) invoke a AIDC-DATA request primitive with the AIDC-inf-tfr-apdu as the *AIDC Data* parameter;
- c) if running, stop the timer t_{2IN} ;
- d) start the timer t_C ; and
- e) set the variables $vs_1 = \text{info-trans}$ and $vs_2 = \text{Msg Number}$

3.2.6.1.32 AIDC-DATA Indication Primitive with an AIDC-inf-tfr-apdu

3.2.6.1.32.1 When Invoked

3.2.6.1.32.1.1 It shall be valid to invoke the AIDC-DATA indication primitive with an AIDC-inf-tfr-apdu when the AIDC-ASE protocol machine is in the IDLE state, or the NOTIFY state, or the NEGOTIATING state, or the RE-NEGOTIATING state, or the COORDINATED state, or the PRE-TRANSFER state, or the TRANSFERRING state, or the TRANSFERRED state or the BACKWARD-COORDINATING state.

3.2.6.1.32.2 Action Upon Invocation

3.2.6.1.32.2.1 Upon the receipt of an AIDC-DATA indication primitive with an AIDC-inf-tfr-apdu the AIDC-ASE shall:

- a) extract the *User Data* parameter, and the *Msg Number* parameter from the AIDC-inf-tfr-apdu;
- b) invoke an AIDC-inf-tfr indication primitive with the extracted parameters in a) above as the AIDC-inf-tfr indication *User Data* parameter, and the AIDC-inf-tfr indication primitive *Msg Number* parameter respectively;
- c) if running, stop the timer t_{1IN} ; and
- d) set the variables $vr_1 = \text{info-trans}$ and $vr_2 = \text{Msg Number}$.

3.2.6.1.33 AIDC-ucf Request Primitive

3.2.6.1.33.1 When Invoked

3.2.6.1.33.1.1 It shall be valid to invoke the AIDC-ucf request primitive when the AIDC-ASE protocol machine is in the IDLE state, or the NOTIFY state, or the NEGOTIATING state, or the RE-NEGOTIATING state, or the COORDINATED state, or the PRE-TRANSFER state, or the TRANSFERRING state, or the TRANSFERRED state, or the BACKWARD-COORDINATING state.

3.2.6.1.33.2 Action Upon Receipt

3.2.6.1.33.2.1 Upon the receipt of an AIDC-ucf request primitive, the AIDC-ASE shall:

- a) create an AIDC-ucf-apdu based on the *Result* parameter, the *Reason* parameter if present, and the *Reference ID* parameter;
- b) if $vr_2 = \text{Reference ID}$, then:
 - 1) invoke an AIDC-DATA request primitive with the AIDC-ucf-apdu as the *User Data* parameter;
 - 2) if the AIDC-ucf request primitive *Result* parameter has the abstract value “accepted” then:
 - i) if in the IDLE state:
 - A) if the variable $vr_1 = \text{notify}$, then:
 - I) start the timer t_{1NC} ; and
 - II) enter the NOTIFY state.
 - B) if the variable $vr_1 = \text{coord-start}$, then enter the NEGOTIATING state.
 - C) if the variable $vr_1 = \text{info-trans}$, then start the timer t_{1NC} ;
 - ii) if in the NOTIFY state:
 - A) if the variable $vr_1 = \text{notify}$, then:
 - I) start the timer t_{1NC} ; and
 - II) enter the NOTIFY state.
 - B) if the variable $vr_1 = \text{coord-start}$, then enter the NEGOTIATING state;
 - C) if the variable $vr_1 = \text{end}$, then enter the IDLE state
 - iii) if in the NEGOTIATING state:
 - A) if the variable $vr_1 = \text{coord-end}$, then:
 - I) if variable $vre = \text{accept}$, then:
 - (a) start the timer t_{1CT} ;

- (b) set the variable $vre = \text{NULL}$; and
 - (c) enter the COORDINATED state.
 - II) if variable $vre = \text{reject}$, then:
 - (a) start the timer t_{INC} ;
 - (b) set the variable $vre = \text{NULL}$; and
 - (c) enter the NOTIFY state.
 - B) if the variable $vr_1 = \text{coord-stndby}$, then start the timer t_s ;
 - C) if the variable $vr_1 = \text{end}$, then enter the IDLE state.
- iv) if in the COORDINATED state:
 - A) if the variable $vr_1 = \text{coord-start}$, then:
 - I) enter RE-NEGOTIATING state.
 - B) if the variable $vr_1 = \text{trns-init}$, then:
 - I) start the timer t_{3R} ; and
 - II) enter PRE-TRANSFER state.
 - C) if the variable $vr_1 = \text{trns-start}$, then enter the TRANSFERRING state.
 - D) if the variable $vr_1 = \text{end}$, then:
 - I) stop all timers; and
 - II) enter the IDLE state.
- v) if in the RE-NEGOTIATING state:
 - A) if the variable $vr_1 = \text{coord-end}$, then:
 - I) start the timer t_{ICT} ; and
 - II) enter the COORDINATED state.
 - B) if the variable $vr_1 = \text{coord-stnby}$, then start the timer t_s ;
 - C) if the variable $vr_1 = \text{end}$, then enter the IDLE state.
- vi) if in the PRE-TRANSFER state:
 - A) if the variable $vr_1 = \text{trns-comm}$, then enter the TRANSFERRING state.
 - B) if the variable $vr_1 = \text{trns-assm}$, then:
 - I) start the timer t_{TE} ; and
 - II) enter the TRANSFERRED state.

- vii) if in the TRANSFERRING state:
 - A) if the variable $vr_1 = \text{trns-assm}$, then:
 - I) start the timer t_{TE} ; and
 - II) enter the TRANSFERRED state.
 - B) if the variable $vr_1 = \text{trns-accept}$, then:
 - I) start the timer t_{TE} ; and
 - II) enter the TRANSFERRED state.
 - C) if the variable $vr_1 = \text{trns-reject}$, then:
 - I) start the timer t_{1CT} ; and
 - II) enter the COORDINATED state.
- viii) if in the TRANSFERRED state:
 - A) if the variable $vr_1 = \text{back}$, then enter the BACKWARD COORDINATING state.
 - B) if the variable $vr_1 = \text{end}$, then enter the IDLE state.
- ix) if in the BACKWARD-COORDINATING state:
 - A) if the variable $vr_1 = \text{back-end}$, then enter the TRANSFERRED state.
- 3) if the AIDC-ucf request primitive *Result* parameter has the abstract value “rejected”, then:
 - i) if in the IDLE state, then:
 - A) if $vr_1 = \text{notify}$ start t_{1NC} ;
 - B) if $vr_1 = \text{coord-start}$ timer t_{1NC} ;
 - C) if $vr_1 = \text{info-trans}$ start timer t_{1IN} ;
 - D) set the variable $vr_1 = \text{NULL}$, $vr_2 = \text{NULL}$ and $vre = \text{NULL}$; and
 - E) remain in the current state.
 - ii) if in the NOTIFY state, then:
 - A) start timer t_{1NC} ;
 - B) set the variable $vr_1 = \text{NULL}$, $vr_2 = \text{NULL}$, and $vre = \text{NULL}$; and
 - C) remain in the same state.

- iii) if in the COORDINATED state then:
 - A) start t_{1CT} ;
 - B) set the variable $vr_1 = \text{NULL}$, $vr_2 = \text{NULL}$, and $vre = \text{NULL}$; and
 - C) remain in the current state.
- iv) if in any other state, then:
 - A) if the variable $vr_1 \neq \text{NULL}$, then start the timer t_{2R} ;
 - B) set the variable $vr_1 = \text{NULL}$, $vr_2 = \text{NULL}$, and $vre = \text{NULL}$; and
 - C) remain in the current state.
- c) if the variable $vr_2 \neq \text{Reference ID}$ then:
 - 1) invoke an AIDC-pvd-abrt indication;
 - 2) invoke an AIDC-ABORT request;
 - 3) stop all timers; and
 - 4) enter the IDLE state.

3.2.6.1.34 AIDC-DATA Indication Primitive with an AIDC-ucf-apdu

3.2.6.1.34.1 When Invoked

3.2.6.1.34.1.1 It shall be valid to invoke the AIDC-DATA indication primitive with an AIDC-ucf-apdu when the AIDC-ASE protocol machine is in the IDLE state, or the NOTIFY state, or the NEGOTIATING state, or the RE-NEGOTIATING state, or the COORDINATED state, or the PRE-TRANSFER state, or the TRANSFERRING state, or the TRANSFERRED state, or the BACKWARD-COORDINATED state.

3.2.6.1.34.2 Action Upon Invocation

3.2.6.1.34.2.1 Upon the receipt of an AIDC-DATA indication primitive with an AIDC-ucf-apdu the AIDC-ASE shall:

- a) stop timer t_c ;
- b) extract the *Result* parameter, the *Reason* parameter if present, and the *Reference ID* parameter from the AIDC-ucf-apdu;

- c) if the variable $vs_2 = \text{Reference ID}$, invoke an AIDC-ucf indication primitive with the extracted parameters in a) above as the AIDC-ucf indication primitive *Result* parameter, the AIDC-ucf indication primitive *Reason* parameter, and the AIDC-ucf indication primitive *Reference ID* parameter;
- 1) if the *Response* parameter of the AIDC-ucf request primitive has the abstract value “accepted”:
 - i) if in the IDLE state, or the NOTIFY state:
 - A) if the variable $vs_1 = \text{notify}$, then:
 - I) start the timer t_{2NC} ; and
 - II) enter the NOTIFY state.
 - B) if the variable $vs_1 = \text{coord-start}$, then:
 - I) start the timer t_{1R} ; and
 - II) enter the NEGOTIATING state.
 - C) if the variable $vs_1 = \text{info-trans}$, then start the timer t_{2IN} .
 - D) if the variable $vs_1 = \text{end}$, then enter IDLE state;
 - ii) if in the NEGOTIATING state:
 - A) if the variable $vs_1 = \text{coord-negot}$, then start the timer t_{1R} .
 - B) if the variable $vs_1 = \text{coord-end}$, then:
 - I) if variable $vse = \text{accept}$, then:
 - (a) start the timer t_{2CT} ; and
 - (b) set the variable $vse = \text{NULL}$; and
 - (c) enter the COORDINATED state.
 - II) if variable $vse = \text{reject}$, then:
 - (a) start the timer t_{2NC} ; and
 - (b) set the variable $vse = \text{NULL}$; and
 - (c) enter the NOTIFY state.
 - C) if the variable $vs_1 = \text{end}$, then enter IDLE state.
 - iii) if in the COORDINATED state:
 - A) if the variable $vs_1 = \text{coord-start}$, then:
 - I) start the timer t_{1R} ; and
 - II) enter the RE-NEGOTIATING state.

- B) if the variable $vs_1 = \text{trns-init}$, then enter the PRE-TRANSFER state.
- C) if the variable $vs_1 = \text{trns-start}$, then enter the TRANSFERRING state.
- D) if the variable $vs_1 = \text{end}$, then:
 - I) stop all timers;
 - II) start timer t_{2NC} ; and
 - III) enter the IDLE state.
- iv) if in the RE-NEGOTIATING state:
 - A) if the variable $vs1 = \text{coord-negot}$, then start the timer t_{1R} ;
 - B) if the variable $vs1 = \text{coord-end}$, then:
 - I) start the timer t_{2CT} ;
 - II) enter the COORDINATED state.
 - C) if the variable $vs_1 = \text{end}$, then enter IDLE state.
- v) if in the PRE-TRANSFER state:
 - A) if the variable $vs1 = \text{trns-prpsl}$, then:
 - I) start timer t_{1R} ; and
 - II) remain in the current state.
 - B) if the variable $vs_1 = \text{trns-comm}$, then enter the TRANSFERRING state.
 - C) if the variable $vs_1 = \text{trns-assm}$, then enter the TRANSFERRED state.
- vi) if in the TRANSFERRING state:
 - A) if the variable $vs_1 = \text{trns-assm}$, then enter the TRANSFERRED state.
 - B) if the variable $vs_1 = \text{trns-accept}$, then enter the TRANSFERRED state.
 - C) if the variable $vs_1 = \text{trns-reject}$, then:
 - I) start the timer t_{2CT} ; and
 - II) enter the COORDINATED state.
- vii) if in the TRANSFERRED state:
 - A) if the variable $vs_1 = \text{back}$, then enter the state BACKWARD COORDINATING

- B) if the variable $vs_1 = \text{end}$, then enter IDLE state.
 - viii) if in the BACKWARD-COORDINATING state:
 - A) if the variable $vs_1 = \text{back-end}$, then enter the TRANSFERRED state.
- 2) if the *Response* parameter of the AIDC-ucf request primitive has the abstract value: rejected, then:
 - i) if in the IDLE state then:
 - A) if $vr_1 = \text{notify}$ start timer t_{2NC} ;
 - B) if $vr_1 = \text{coord-start}$ start timer t_{2NC} ;
 - C) if $vr_1 = \text{info-trans}$ start timer t_{2IN} ;
 - D) set the variable $vs_1 = \text{NULL}$, $vs_2 = \text{NULL}$ and $vse = \text{NULL}$; and
 - E) remain in the current state.
 - ii) if in the NOTIFY state then:
 - A) start timer t_{2NC} ;
 - B) set the variable $vs_1 = \text{NULL}$, $vs_2 = \text{NULL}$, and $vse = \text{NULL}$; and
 - C) remain in the current state.
 - iii) if in the COORDINATED state then:
 - A) start timer t_{2CT} ;
 - B) set the variable $vs_1 = \text{NULL}$, $vs_2 = \text{NULL}$, and $vse = \text{NULL}$; and
 - C) remain in the current state.
 - iv) if in any other state then:
 - A) stop all timers;
 - B) set the variable $vs_1 = \text{NULL}$, $vs_2 = \text{NULL}$, and $vse = \text{NULL}$; and
 - C) remain in the current state.
- d) if the variable $vs_2 \neq \text{Reference ID}$, then:
 - 1) invoke an AIDC-pvd-abrt indication;
 - 2) invoke an AIDC-ABORT request;
 - 3) stop all timers; and

- 4) enter the IDLE state.

3.2.6.1.35 AIDC-end Request Primitive

3.2.6.1.35.1 When Invoked

3.2.6.1.35.1.1 It shall be valid to invoke the AIDC-end request primitive when the AIDC-ASE protocol machine is in the NOTIFY state, or the COORDINATED state, or the TRANSFERRED state.

3.2.6.1.35.2 Action Upon Invocation

3.2.6.1.35.2.1 Upon the receipt of an AIDC-end request primitive the AIDC-ASE shall:

- a) create an AIDC-end-apdu based on the *User Data* parameter, if present, and the *Msg Number* parameter;
- b) invoke a AIDC-DATA request primitive with the AIDC-end-apdu as the *AIDC Data* parameter;
- c) if in the NOTIFY state, stop the timer t_{2NC} ;
- d) if in the COORDINATED state, then:
 - 1) if running, stop the timer t_{1CT} ;
 - 2) if running, stop the timer t_{2CT} ;
- e) start the timer t_C ; and
- f) set the variable $vs_1 = \text{end}$ and $vs_2 = \text{Msg Number}$.

3.2.6.1.36 AIDC-DATA Request with an AIDC-end-apdu

3.2.6.1.36.1 When Invoked

3.2.6.1.36.1.1 It shall be valid to invoke the AIDC-DATA indication primitive with an AIDC-end-apdu when the AIDC-ASE protocol machine is in the IDLE state, or the NOTIFY state, or the COORDINATED state, or the TRANSFERRED state.

3.2.6.1.36.2 Action Upon Invocation

3.2.6.1.36.2.1 Upon the receipt of an AIDC-DATA indication primitive with an AIDC-end-apdu the AIDC-ASE shall:

- a) extract the *User Data* parameter, if present, and the *Msg Number* parameter from the AIDC-end-apdu;
- b) invoke an AIDC-end indication primitive with the extracted parameters in a) above as the AIDC-end indication *User Data* parameter, and the AIDC-end indication primitive *Msg Number* parameter respectively;
- c) if in the IDLE state, stop the timer t_{2R} ;
- d) if in the NOTIFY state, stop the timer t_{1NC} ;

- e) if in the COORDINATED state:
 - 1) if running, stop the timer t_{1CT} ;
 - 2) if running, stop the timer t_{2CT} ;
- f) if in the TRANSFERRED state, stop the timer t_{TE} ; and
- g) set the variable $vr_1 = \text{end}$ and $vr_2 = \text{Msg Number}$.

3.2.6.1.37 AIDC-usr-abrt Request Primitive

3.2.6.1.37.1 When Invoked

3.2.6.1.37.1.1 It shall be valid to invoke the AIDC-usr-abrt request primitive when the AIDC-ASE protocol machine is in the NOTIFY state, or the NEGOTIATING state, or the RE-NEGOTIATING state, or the COORDINATED state, or the PRE-TRANSFER state, or the TRANSFERRING state, or the TRANSFERRED state, or the BACKWARD-COORDINATING state.

3.2.6.1.37.2 Action Upon Invocation

3.2.6.1.37.2.1 Upon the receipt of an AIDC-usr-abrt request primitive the AIDC-ASE shall:

- a) invoke an AIDC-ABORT request primitive;
- b) stop all timers; and
- c) enter the IDLE state.

3.2.6.1.38 AIDC-ABORT Indication Primitive

3.2.6.1.38.1 When Invoked

3.2.6.1.38.1.1 It shall be valid to invoke the AIDC-ABORT indication primitive when the AIDC-ASE protocol machine is in the NOTIFY state, or the NEGOTIATING state, or the RE-NEGOTIATING state, or the COORDINATED state, or the PRE-TRANSFER state, or the TRANSFERRING state, or the TRANSFERRED state, or the BACKWARD-COORDINATING state.

3.2.6.1.38.2 Action Upon Invocation

3.2.6.1.38.2.1 Upon the receipt of a AIDC-ABORT indication primitive the AIDC-ASE shall:

- a) invoke an AIDC-usr-abrt indication primitive;
- b) stop all timers; and
- c) enter the IDLE state.

3.2.6.1.39 AIDC-P-ABORT indication

3.2.6.1.39.1 When Invoked

3.2.6.1.39.1.1 It shall be valid to invoke the AIDC-P-ABORT indication primitive when the AIDC-ASE protocol machine is in the NOTIFY state, or the NEGOTIATING state, or the RE-NEGOTIATING state, or the COORDINATED state, or the PRE-TRANSFER state, or the TRANSFERRING state, or the TRANSFERRED state, or the BACKWARD-COORDINATING state.

3.2.6.1.39.2 Action Upon Invocation

3.2.6.1.39.2.1 Upon the receipt of a AIDC-P-ABORT indication primitive the AIDC-ASE shall:

- a) invoke an AIDC-pvd-abrt indication primitive with the *AbortReason* parameter set to the abstract value: communications service failure;
- b) stop all timers; and
- c) enter the IDLE state.

3.2.6.2 Exception Handling

3.2.6.2.1 Timer Expiration

3.2.6.2.1.1 If a timer expires the AIDC-ASE shall:

- a) interrupt any current activity;
- b) invoke AIDC-ABORT request primitive; and
- c) invoke an AIDC-pvd-abrt indication primitive with the AIDC-pvd-abrt *AbortReason* parameter set to the abstract value: timer expired.

3.2.6.2.2 Irrecoverable System Error

3.2.6.2.2.1 If an AIDC-ASE has an irrecoverable system error, the AIDC-ASE shall:

- a) interrupt any current activity;
- b) invoke AIDC-ABORT request primitive; and
- c) invoke an AIDC-pvd-abrt indication with the AIDC-pvd-abrt *AbortReason* parameter set to the abstract value: undefined-error.

3.2.6.2.3 Invalid APDU

3.2.6.2.3.1 If a APDU received is determined to be invalid, the AIDC-ASE shall:

- a) invoke an AIDC-ABORT request primitive; and
- b) invoke an AIDC-pvd-abrt indication primitive with the *AbortReason* parameter set to the abstract value: invalid APDU; and
- c) enter the IDLE state.

3.2.6.3 AIDC Application Timers

Note.— Table 3.2.6-1 lists the time constraints related to the AIDC application. Each time constraint requires a timer to be set in the AIDC protocol machine.

3.2.6.3.1 An AIDC-ASE shall measure the time between the initial event and the corresponding final event for each of the timers in Table 3.2.6-1.

3.2.6.3.1.1 An implementation of the AIDC-ASE shall provide a means for configuring each of the timer values in Table 3.2.6.1.

Note.— The exact means used to configure timer values is a local matter.

3.2.6.3.2 If the maximum time is exceeded before the final event has occurred, an AIDC-ASE shall take appropriate action as defined in 3.2.6.2.1.

3.2.6.3.2.1 **Recommendation.**—*The actions defined in 3.2.6.2.1 should be taken when the maximum time as indicated in Table 3.2.6-1 has expired.*

Table 3.2.6-1: AIDC-ASE Timers

Purpose	Timer	Timer Value	Timer Start Event	Timer Stop Event (Primitive invocation)
User Confirmation for all AIDC services except Abort services	t_c	2 min	AIDC-DATA Req with AIDC-ASE-apdu	AIDC-DATA Ind with AIDC-ucf-apdu
Monitors time for transition between an initial invocation of the Info-transfer service primitive in the IDLE state and a subsequent Regime	t_{1IN}	see 3.2.6.3.1.1	+ve AIDC-ucf Ind in response to AIDC-inf-tfr Req	•AIDC-nfy Ind •AIDC-crd-start Ind
Monitors timer for transition between an initial invocation of the Info-transfer service primitive in the IDLE state and a subsequent Regime	t_{2IN}	see 3.2.6.3.1.1	+ve AIDC-ucf-IND in response to AIDC-inf-tfr-Req	•AIDC-nfy Req •AIDC-crd-start Req
Monitors time for transition between Notifying Regime and Coordinating Regime.	t_{1NC}	see 3.2.6.3.1.1 TBD	+ve AIDC-ucf Req in response to AIDC-nfy Ind	•AIDC-nfy Ind •AIDC-crd-start Ind •AIDC-end Ind
Monitors time for transition between Notifying Regime and Coordinating Regime. Complementary timer for t_{1NC} .	t_{2NC}	see 3.2.6.3.1.1	+ve AIDC-ucf Ind in response to AIDC-nfy Req	•AIDC-nfy Req •AIDC-crd-start Req •AIDC-end Req
Response monitoring timer used with the following services: •AIDC-crd-start •AIDC-crd-ngtt •AIDC-tfr-prpsl (if P1 true) •AIDC-tfr-comm •AIDC-tfr-cntrl	t_{1R}	see 3.2.6.3.1.1	+ve AIDC-ucf Ind in response to: •AIDC-crd-start Req •AIDC-crd-ngtt Req •AIDC-tfr-prpsl Req (if P1 true) •AIDC-tfr-comm Req •AIDC-tfr-cntrl Req	•AIDC-crd-ngtt Ind •AIDC-crd-stndby Ind •AIDC-crd-end Ind •AIDC-tfr-accept Ind •AIDC-tfr-comm-assm Ind •AIDC-tfr-cntrl Cnf

Purpose	Timer	Timer Value	Timer Start Event	Timer Stop Event (Primitive invocation)
Response monitoring timer after -ve User Confirmation for all services except: •AIDC-tfr-rqst •AIDC-tfr-prpsl •AIDC-tfr-accept •AIDC-info-transfer •AIDC-usr-abrt •AIDC-pvd-abrt	t_{2R}	see 3.2.6.3.1.1	-ve AIDC-ucf Req	AIDC-ASE service Ind primitive except for the following services: •AIDC-tfr-rqst •AIDC-tfr-prpsl •AIDC-tfr-accept •AIDC-inf-tfr •AIDC-usr-abrt •AIDC-pvd-abrt
Standby timer associated with AIDC-crd-stndby service.	t_S	see 3.2.6.3.1.1	+ve AIDC-ucf Req in response to AIDC-crd-stndby Ind	•AIDC-crd-ngtt Ind •AIDC-crd-end Ind
Monitors time for transition between Coordinating Regime and Transferring Regime.	t_{1CT}	see 3.2.6.3.1.1	+ve AIDC-ucf Req in response to AIDC-crd-end Ind	•AIDC-tfr-init Ind or Req •AIDC-tfr-cntrl Ind or Req •AIDC-end Ind or Req •AIDC-crd-start Ind or Req
Monitors time for transition between Coordinating Regime and Transferring Regime. Complementary timer for t_{1CT} .	t_{2CT}	see 3.2.6.3.1.1	+ve AIDC-ucf Ind in response to AIDC-crd-end Req	•AIDC-tfr-init Req or Ind •AIDC-tfr-cntrl Req or Ind •AIDC-end Req or Ind •AIDC-crd-start Ind or Req
Response monitoring timer used with the AIDC-tfr-init service	t_{3R}	see 3.2.6.3.1.1	+ve AIDC-ucf Req in response to AIDC-tfr-init Ind	•AIDC-tfr-comm Ind •AIDC-tfr-comm-asm-Req
End timer to monitor time after completion of Transferring Regime to the invocation of the AIDC-end service or commencement of backward coordination.	t_{TE}	see 3.2.6.3.1.1	+ve AIDC-ucf Req in response to AIDC-tfr-comm-asm Ind or AIDC-tfr-cntrl Cnf	•AIDC-crd-start Ind or Req •AIDC-end Ind

3.2.6.4 State Table

3.2.6.4.1 The AIDC-ASE shall behave in accordance with the following state table, which show diagrammatically the state transitions and actions performed by the AIDC-ASE in response to incoming events. Incoming events are shown in the first column of the state table. Each cell in the state table shows:

- a) optionally, one or more variables, denoted “vrN”, or “vsN”, where N is either an integer or the character “e”. The variables are defined as required and take on a value within the state table.
- b) the new state that the AIDC-ASE shall enter after the action has been performed
- c) the action, if any, which the AIDC-ASE shall perform.

3.2.6.4.2 Blank cells indicate error conditions. The error handling shall result in the association being aborted, if one exists, and a notification being given to the AIDC-User.

3.2.6.4.3 In the event of a conflict between the actions implied by the state table and the text in the above, the text shall take precedence.

Note 1.— Variables

vr1-variable holding the last received event type for saving

vr2-variable holding the last received Msg Number for saving

vre-variable holding the last receive Result parameter of the AIDC-crd-end service for saving

vs1-variable holding the last sent event type for saving

vs2-variable holding the last sent Msg Number for saving

vse-variable holding the last send Result parameter of the AIDC-crd-end service for saving

Note 2.— Predicates

c1 - Logical Confirmation result = Accept

c2 - vr2 = Reference ID

c3 - vs2 = Reference ID

c4 - vr1 = notify

c5 - vs1 = notify

c6 - vr1 = coord-start

c7 - vs1 = coord-start

c8 - vr1 = coord-negot

c9 - vs1 = coord-negot

c10 - vr1 = coord-standby

c11 - vs1 = coord-standby

c12 - vr1 = coord-end

c13 - vs1 = coord-end

c14 - vr1 = trns-init

c15 - vs1 = trns-init

c16 - vr1 = trns-start

c17 - vs1 = trns-start

c18 - vr1 = trns-accept

c19 - vs1 = trns-accept

c20 - vr1 = trns-reject

c21 - vs1 = trns-reject

c22 - vr1 = trns-comm

c23 - vs1 = trns-comm

c24 - vr1 = trns-assm

c25 - vs1 = trns-assm

c26 - vr1 = back

c27 - vs1 = back

c28 - vr1 = back-end

c29 - vs1 = back-end

c30 - vr1 = end

c31 - vs1 = end

c32 - vr1 = info-trans

c33 - vs1 = info-trans

c34 - AIDC-crd-end Req/Ind Result = accept

c35 - AIDC-crd-end Req/Ind Result = reject

c36 - AIDC-tfr-cntrl Rsp/Cnf Result = accepted

c37 - AIDC-tfr-cntrl Rsp/Cnf Result = rejected

c38 - vr1 = NULL

c39 - vre = accept

c40 - vre = reject

c41 - vse = accept

c42 - vse = reject

c43 - vs1 = trns-prpsl

Note 3.— Where a predicate is shown in the state table with a preceding exclamation mark (!), this indicates that the predicate is not true.

Table 3.2.6-2: State Table

State→ Event↓	IDLE	NOTIFY	NEGOTIATING	RE-NEGOTIATING	COORDINATED
AIDC-nfy REQ	• vs1= notify • vs2=Msg Number • AIDC-DATA REQ with AIDC-nfy-apdu • start t_C • stop t_{2IN} ⇒ IDLE	• vs1= notify • vs2=Msg Number • AIDC-DATA REQ with AIDC-nfy-apdu • stop t_{2NC} • stop t_{2IN} • start t_C ⇒ NOTIFY			
rcv AIDC-nfy-apdu	• vr1= notify • vr2=Msg Number • AIDC-nfy IND • stop t_{2R} • stop t_{1IN} ⇒ IDLE	• vr1= notify • vr2=Msg Number • AIDC-nfy IND • stop t_{1NC} • stop t_{2R} • stop t_{1IN} ⇒ NOTIFY			
AIDC-crd-start REQ	• vs1= back • vs2=Msg Number • start t_C • stop t_{2IN} • AIDC-DATA REQ with AIDC-crd-start-apdu ⇒ IDLE	• vs1= back • vs2=Msg Number • stop t_{2IN} • stop t_{1CT} • start t_C • AIDC-DATA REQ with AIDC-crd-start-apdu ⇒ NOTIFY			• vs1= back • vs2=Msg Number • stop t_{2IN} • stop t_{1CT} • stop t_{2CT} • start t_C • AIDC-DATA REQ with AIDC-crd-start-apdu ⇒ COORDINATED
rcv AIDC-crd-start-apdu	• vr1= back • vr2=Msg Number • AIDC-crd-start IND • stop t_{1IN} • stop t_{2R} ⇒ IDLE	• vr1= back • vr2=Msg Number • AIDC-crd-start IND • stop t_{1NC} • stop t_{1IN} • stop t_{2R} ⇒ NOTIFY			• vr1= back • vr2=Msg Number • AIDC-crd-start IND • stop t_{1CT} • stop t_{2CT} • stop t_{1IN} • stop t_{2R} ⇒ COORDINATED
AIDC-crd-end REQ			• vs1= coord-end • vs2=Msg Number • start t_C • AIDC-DATA REQ with AIDC-crd-end-apdu • if c34 vse=accept • if c35 vse=reject ⇒ NEGOTIATING	• vs1= coord-end • vs2=Msg Number • start t_C • AIDC-DATA REQ with AIDC-crd-end-apdu ⇒ RE-NEGOTIATING	

State→ Event↓	IDLE	NOTIFY	NEGOTIATING	RE-NEGOTIATING	COORDINATED
rcv AIDC-crd-end-apdu			• vr1 = coord-end • vr2 = Msg Number • AIDC-crd-end IND • stop t_{1R} • stop t_{2R} • stop t_s • if c34 vre=accept ⇒ NEGOTIATING	• vr1 = coord-end • vr2 = Msg Number • AIDC-crd-end IND • stop t_{1R} • stop t_{2R} • stop t_s ⇒ RE-NEGOTIATING	
AIDC-crd-ngtt REQ			• vs1 = coord-negot • vs2 = Msg Number • AIDC-DATA with AIDC-crd-ngtt-apdu • start t_C ⇒ NEGOTIATING	• vs1 = coord-negot • vs2 = Msg Number • AIDC-DATA with AIDC-crd-ngtt-apdu • start t_C ⇒ RE-NEGOTIATING	
rcv AIDC-crd-ngtt-apdu			• vr1 = coord-negot • vr2 = coord-negot • AIDC-crd-ngtt IND • stop t_{1R} • stop t_{2R} • stop t_s ⇒ NEGOTIATING	• vr1 = coord-negot • vr2 = coord-negot • AIDC-crd-ngtt IND • stop t_{1R} • stop t_{2R} • stop t_s ⇒ RE-NEGOTIATING	
AIDC-crd-stndby REQ			• vs1 = coord-stndby • vs2 = Msg Number • AIDC-DATA REQ with AIDC-crd-stndby-apdu • start t_C ⇒ NEGOTIATING	• vs1 = coord-stndby • vs2 = Msg Number • AIDC-DATA REQ with AIDC-crd-stndby-apdu • start t_C ⇒ RE-NEGOTIATING	
rcv AIDC-crd-stndby-apdu			• vr1 = coord-stndby • vr2 = Msg Number • AIDC-crd-stndby IND • stop t_{1R} • stop t_{2R} ⇒ NEGOTIATING	• vr1 = coord-stndby • vr2 = Msg Number • AIDC-crd-stndby IND • stop t_{1R} • stop t_{2R} ⇒ RE-NEGOTIATING	
AIDC-tfr-init REQ					• vs1 = trns-init • vs2 = Msg Number • AIDC-DATA REQ with AIDC-tfr-init-apdu • stop t_{1CT} • stop t_{2CT} • start t_C ⇒ COORDINATED

State→ Event↓	IDLE	NOTIFY	NEGOTIATING	RE-NEGOTIATING	COORDINATED
rcv AIDC-tfr-init -apdu					• vr1 = trns-init • vr2 = Msg Number • AIDC-tfr-init IND • stop t_{1CT} • stop t_{2CT} ⇒ COORDINATED
AIDC-tfr-rqst REQ					• vs2 = Msg Number • AIDC-DATA REQ with AIDC-tfr-rqst-apdu • start t_C ⇒ COORDINATED
rcv AIDC-tfr-rqst-apdu					• vr2 = Msg Number • AIDC-tfr-rqst IND ⇒ COORDINATED
AIDC-tfr-cntrl REQ					• vs1 = trns-start • vs2 = Msg Number • AIDC-DATA REQ with AIDC-tfr-cntrl-Req-apdu • stop t_{1CT} • stop t_{2CT} • start t_C ⇒ COORDINATED
rcv AIDC-tfr-cntrl-req-apdu					• vr1 = trns-start • vr2 = Msg Number • AIDC-tfr-cntrl IND • stop t_{1CT} • stop t_{2CT} • stop t_{2R} ⇒ COORDINATED
AIDC-inf-tfr REQ	• vs1 = info-trans • vs2 = Msg Number • AIDC-DATA REQ with AIDC-inf-tfr-apdu • start t_C • stop t_{2IN} ⇒ IDLE	• vs1 = info-trans • vs2 = Msg Number • AIDC-DATA REQ with AIDC-inf-tfr-apdu • start t_C • stop t_{2IN} ⇒ NOTIFY	• vs1 = info-trans • vs2 = Msg Number • AIDC-DATA REQ with AIDC-inf-tfr-apdu • start t_C • stop t_{2IN} ⇒ NEGOTIATING		• vs1 = info-trans • vs2 = Msg Number • AIDC-DATA REQ with AIDC-inf-tfr-apdu • start t_C • stop t_{2IN} ⇒ COORDINATED
rcv AIDC-inf-tfr-apdu	• vr1 = info-trans • vr2 = Msg Number • AIDC-inf-tfr IND • stop t_{1IN} ⇒ IDLE	• vr1 = info-trans • vr2 = Msg Number • AIDC-inf-tfr IND • stop t_{1IN} ⇒ NOTIFY	• vr1 = info-trans • vr2 = Msg Number • AIDC-inf-tfr IND • stop t_{1IN} ⇒ NEGOTIATING		• vr1 = info-trans • vr2 = Msg Number • AIDC-inf-tfr IND • stop t_{1IN} ⇒ COORDINATED

State→ Event↓	IDLE	NOTIFY	NEGOTIATING	RE-NEGOTIATING	COORDINATED
AIDC-end REQ		• vs1 = end • vs2 = Msg Number • AIDC-DATA REQ with AIDC-end-apdu • stop t_{2NC} • start t_C ⇒ NOTIFY			• vs1 = end • vs2 = Msg Number • AIDC-DATA REQ with AIDC-end-apdu • stop t_{1CT} • stop t_{2CT} • start t_C ⇒ COORDINATED
rcv AIDC-end -apdu	• vr1 = end • vr2 = Msg Number • AIDC-end IND • stop t_{2R} ⇒ IDLE	• vr1 = end • vr2 = Msg Number • AIDC-end IND • stop t_{1NC} ⇒ NOTIFY			• vr1 = end • vr2 = Msg Number • AIDC-end IND • stop t_{1CT} • stop t_{2CT} ⇒ COORDINATED
AIDC-usr-abrt REQ	• AIDC-ABORT REQ • stop all timers ⇒ IDLE	• AIDC-ABORT REQ • stop all timers ⇒ IDLE	• AIDC-ABORT REQ • stop all timers ⇒ IDLE	• AIDC-ABORT REQ • stop all timers ⇒ IDLE	• AIDC-ABORT REQ • stop all timers ⇒ IDLE
AIDC-ABORT IND		• AIDC-usr-abrt IND • stop all timers ⇒ IDLE	• AIDC-usr-abrt IND • stop all timers ⇒ IDLE	• AIDC-usr-abrt IND • stop all timers ⇒ IDLE	• AIDC-usr-abrt IND • stop all timers ⇒ IDLE
AIDC-P-ABORT IND		• AIDC-pvd-abrt IND • stop all timers ⇒ IDLE	• AIDC-pvd-abrt IND • stop all timers ⇒ IDLE	• AIDC-pvd-abrt IND • stop all timers ⇒ IDLE	• AIDC-pvd-abrt IND • stop all timers ⇒ IDLE

State→ Event↓	IDLE	NOTIFY	NEGOTIATING	RE-NEGOTIATING	COORDINATED
AIDC-ucf REQ	- if c2 { • AIDC-DATA REQ with AIDC-ucf-apdu • if c1 & c4 { • start t_{INC} ⇒ NOTIFY} • if c1 & c6 ⇒ NEGOTIATING • if c1 & c32 { • start t_{INC} ⇒ IDLE } • if !c1 { • if c4 start t_{INC} • if c6 start t_{INC} • if c32 start t_{INC} • vr1 = NULL • vr2 = NULL • vre = NULL ⇒ IDLE } } - if !c2 { • AIDC-pvd-abrt IND • AIDC-ABORT REQ • stop all timers ⇒ IDLE }	- if c2 { • AIDC-DATA REQ with AIDC-ucf-apdu • if c1 & c4 { • start t_{INC} ⇒ NOTIFY} • if c1 & c6 ⇒ NEGOTIATING • if c1 & c30 ⇒ IDLE • if !c1 { • start t_{INC} • vr1 = NULL • vr2 = NULL • vre = NULL ⇒ NOTIFY } } - if !c2 { • AIDC-pvd-abrt IND • AIDC-ABORT REQ • stop all timers ⇒ IDLE }	- if c2 { • AIDC-DATA REQ with AIDC-ucf-apdu • if c1 & c12 & c39 { • start t_{ICT} • vre = NULL ⇒ COORDINATED} • if c1 & c12 & c40 { • start t_{INC} • vre = NULL ⇒ NOTIFY } • if c1 & c10 { • start t_s ⇒ NEGOTIATING } • if c1 & c30 ⇒ IDLE • if !c1 { • if !c38 start t_{2R} • vr1 = NULL • vr2 = NULL • vre = NULL ⇒ NEGOTIATING } } - if !c2 { • AIDC-pvd-abrt IND • AIDC-ABORT REQ • stop all timers ⇒ IDLE }	- if c2 { • AIDC-DATA REQ with AIDC-ucf-apdu • if c1 & c12 { • start t_{ICT} ⇒ COORDINATED} • if c1 & c10 { • start t_s ⇒ RE-NEGOTIATING } • if c1 & c30 ⇒ IDLE • if !c1 { • if !c38 start t_{2R} • vr1 = NULL • vr2 = NULL • vre = NULL ⇒ RE-NEGOTIATING } } - if !c2 { • AIDC-pvd-abrt IND • AIDC-ABORT REQ • stop all timers ⇒ IDLE }	- if c2 { • AIDC-DATA REQ with AIDC-ucf-apdu • if c1 & c6 ⇒ RENEGOTIATING • if c1 & c14 { • start t_{3R} ⇒ PRE-TRANSFER } • if c1 & c16 ⇒ TRANSFERRING • if c1 & c30 { • stop all timers ⇒ IDLE } • if !c1 { • start t_{ICT} • vr1 = NULL • vr2 = NULL • vre = NULL ⇒ COORDINATED } } - if !c2 { • AIDC-pvd-abrt IND • AIDC-ABORT REQ • stop all timers ⇒ IDLE }

State→ Event↓	IDLE	NOTIFY	NEGOTIATING	RE-NEGOTIATING	COORDINATED
rcv AIDC-ucf REQ pdu	- if c3 { • AIDC-ucf IND • if c1 & c5 { • start t_{2NC} ⇒ NOTIFY } • if c1 & c7 { • start t_{1R} ⇒ NEGOTIATING } • if c1 & c33 { • start t_{2IN} ⇒ IDLE } • if c1 & c31 ⇒ IDLE • if !c1 { • if c4 start t_{2NC} • if c6 start t_{2NC} • if c32 start t_{2IN} • vs1=NULL • vs2=NULL • vse=NULL ⇒ IDLE } } - if !c3 { • AIDC-pvd-abrt IND • AIDC-ABORT REQ • stop all timers ⇒ IDLE }	- if c3 { • AIDC-ucf IND • if c1 & c5 { • start t_{2NC} ⇒ NOTIFY } • if c1 & c7 { • start t_{1R} ⇒ NEGOTIATING } • if c1 & c33 { • start t_{2IN} ⇒ NOTIFY } • if c1 & c31 ⇒ IDLE • if !c1 { • start t_{2NC} • vs1=NULL • vs2=NULL • vse=NULL ⇒ NOTIFY } } - if !c3 { • AIDC-pvd-abrt IND • AIDC-ABORT REQ • stop all timers ⇒ IDLE }	- if c3 { • AIDC-ucf IND • if c1 & c9 { • start t_{1R} ⇒ NEGOTIATING } • if c1 & c13 & c41 { • start t_{2CT} • vse = NULL ⇒ COORDINATED } • if c1 & c13 & c42 { • start t_{2NC} • vse = NULL ⇒ NOTIFY } • if c1 & c31 ⇒ IDLE • if !c1 { • stop all timers • vs1=NULL • vs2=NULL • vse=NULL ⇒ NEGOTIATING } } - if !c3 { • AIDC-pvd-abrt IND • AIDC-ABORT REQ • stop all timers ⇒ IDLE }	- if c3 { • AIDC-ucf IND • if c1 & c9 { • start t_{1R} ⇒ RE-NEGOTIATING } • if c1 & c13 { • start t_{2CT} ⇒ COORDINATED } • if c1 & c31 ⇒ IDLE • if !c1 { • stop all timers • vs1=NULL • vs2=NULL • vse=NULL ⇒ RE-NEGOTIATING } } - if !c3 { • AIDC-pvd-abrt IND • AIDC-ABORT REQ • stop all timers ⇒ IDLE }	- if c3 { • AIDC-ucf IND • if c1 & c7 { • start t_{1R} ⇒ RE-NEGOTIATING } • if c1 & c15 ⇒ PRE-TRANSFER • if c1 & c17 ⇒ TRANSFERRING • if c1 & c31 { • stop all timers • start t_{2NC} ⇒ IDLE } • if !c1 { • start t_{2CT} • vs1=NULL • vs2=NULL • vse=NULL ⇒ COORDINATED } } - if !c3 { • AIDC-pvd-abrt IND • AIDC-ABORT REQ • stop all timers ⇒ IDLE }

State→ Event↓	PRE-TRANSFER	TRANSFERRING	TRANSFERRED	BACKWARD COORDINATING
AIDC-crd-start REQ			• vs1 = back • vs2 = Msg Number • AIDC-DATA REQ with AIDC-crd-start-apdu • start t_C • stop t_{2IN} • stop t_{TE} ⇒ TRANSFERRED	
rcv AIDC-crd-start-apdu			• vr1 = back • vr2 = Msg Number • AIDC-crd-start IND • stop t_{1IN} • stop t_{2R} • stop t_{TE} ⇒ TRANSFERRED	
AIDC-crd-end REQ				• vs1 = back-end • vs2 = Msg Number • AIDC-DATA REQ with AIDC-crd-end-apdu • start t_C ⇒ BACKWARD COORDINATING
rcv AIDC-crd-end-apdu				• vr1 = back-end • vr2 = Msg Number • AIDC-crd-end IND • stop t_{1R} • stop t_{2R} • stop t_S ⇒ BACKWARD COORDINATING
AIDC-crd-ngtt REQ				• vs1 = coord-negot • vs2 = Msg Number • AIDC-DATA REQ with AIDC-crd-ngtt-apdu • start t_C ⇒ BACKWARD-COORDINATING
rcv AIDC-crd-ngtt-apdu				• vr1 = coord-negot • vr2 = Msg Number • AIDC-crd-ngtt IND • stop t_{1R} • stop t_{2R} • stop t_S ⇒ BACKWARD-COORDINATING

State→ Event↓	PRE-TRANSFER	TRANSFERRING	TRANSFERRED	BACKWARD COORDINATING
AIDC-crd-stdby REQ				• vs1 = coord-stdby • vs2 = Msg Number • AIDC-DATA REQ with AIDC-crd-stdby-apdu • start t_C ⇒ BACKWARD COORDINATING
rcv AIDC-crd-stdby -apdu				• vr1 = coord-stdby • vr2 = Msg Number • stop t_{1R} • stop t_{2R} ⇒ BACKWARD-COORDINATING
AIDC-tfr-prpsl REQ	• if P1, vs1 = trns-prpsl • vs2 = Msg Number • AIDC-DATA REQ with AIDC-tfr-prpsl-apdu • start t_C ⇒ PRE-TRANSFER			
rcv AIDC-tfr-prpsl-apdu	• If P1, vr1 = trns-prpsl • vr2 = Msg Number • AIDC_tfr-prpsl IND ⇒ PRE-TRANSFER			
AIDC-tfr-accept REQ	if P1 { • vs2 = Msg Number • AIDC-DATA REQ with AIDC-tfr-accept-apdu • start t_C ⇒ PRE-TRANSFER}			
rcv AIDC-tfr-accept-apdu	if P1 { • stop t_{1R} • AIDC-tfr-accept IND • vr2 = Msg Number ⇒ PRE-TRANSFER}			
AIDC-tfr-comm REQ	• vs1 = trns-comm • vs2 = Msg Number • AIDC-DATA REQ with AIDC-tfr-comm-apdu • start t_C ⇒ PRE-TRANSFER			

State→ Event↓	PRE-TRANSFER	TRANSFERRING	TRANSFERRED	BACKWARD COORDINATING
rcv AIDC-tfr-comm apdu	• vr1 = trns-comm • vr2 = Msg Number • AIDC-tfr-comm IND • stop t_{2R} • stop t_{3R} ⇒ PRE-TRANSFER			
AIDC-tfr-comm-assm REQ	• vs1 = trns-assm • vs2 = Msg Number • AIDC-DATA REQ with AIDC-tfr-comm-assm-apdu • start t_C • stop t_{3R} ⇒ PRE-TRANSFER	• vs1 = trns-assm • vs2 = Msg Number • AIDC-DATA REQ with AIDC-tfr-comm-assm-apdu • start t_C • stop t_{3R} ⇒ TRANSFERRING		
rcv AIDC-tfr-comm-assm apdu	• vr1 = tnsf-assm • vr2 = Msg Number • AIDC-tfr-comm-assm IND • stop t_{1R} • stop t_{2R} ⇒ PRE-TRANSFER	• vr1 = tnsf-assm • vr2 = Msg Number • AIDC-tfr-comm-assm IND • stop t_{1R} • stop t_{2R} ⇒ TRANSFERRING		
AIDC-tfr-cntrl RSP		• if c36 vs1 = trns-accept • if c37 vs1 = trns-reject • vs2 = Msg Number • AIDC-DATA REQ with AIDC-tfr-cntrl-Rsp-apdu • start t_C TRANSFERRING		
rcv AIDC-tfr-cntrl-Rsp-apdu		• if c36 vr1 = trns-accept • if c37 vr1 = trns-reject • vr2 = Msg Number • AIDC-tfr-cntrl CNF • stop t_{1R} • stop t_{2R} ⇒ TRANSFERRING		
AIDC-inf-tfr REQ	• vs1 = info-trans • vs2 = Msg Number • AIDC-DATA REQ with AIDC-inf-tfr-apdu • start t_C • stop t_{2IN} ⇒ PRE-TRANSFER	• vs1 = info-trans • vs2 = Msg Number • AIDC-DATA REQ with AIDC-inf-tfr-apdu • start t_C • stop t_{2IN} ⇒ TRANSFERRING	• vs1 = info-trans • vs2 = Msg Number • AIDC-DATA REQ with AIDC-inf-tfr-apdu • start t_C • stop t_{2IN} ⇒ TRANSFERRED	

State→ Event↓	PRE-TRANSFER	TRANSFERRING	TRANSFERRED	BACKWARD COORDINATING
rcv AIDC-inf-tfr-apdu	• vr1 = info-trans • vr2 = Msg Number • AIDC-inf-tfr IND • stop t_{1IN} ⇒ PRE-TRANSFER	• vr1 = info-trans • vr2 = Msg Number • AIDC-inf-tfr IND • stop t_{1IN} ⇒ TRANSFERRING	• vr1 = info-trans • vr2 = Msg Number • AIDC-inf-tfr IND • stop t_{1IN} ⇒ TRANSFERRED	
AIDC-end REQ			• vs1 = end • vs2 = Msg Number • AIDC-DATA REQ with AIDC-end-apdu • start t_C ⇒ TRANSFERRED	
rcv AIDC-end REQ pdu			• vr1 = end • vr2 = Msg Number • AIDC-end IND • stop t_{TE} ⇒ TRANSFERRED	
AIDC-usr-abrt REQ	• AIDC-ABORT REQ • stop all timers ⇒ IDLE	• AIDC-ABORT REQ • stop all timers ⇒ IDLE	• AIDC-ABORT REQ • stop all timers ⇒ IDLE	• AIDC-ABORT REQ • stop all timers ⇒ IDLE
AIDC-ABORT IND	• AIDC-usr-abrt IND • stop all timers ⇒ IDLE	• AIDC-usr-abrt IND • stop all timers ⇒ IDLE	• AIDC-usr-abrt IND • stop all timers ⇒ IDLE	• AIDC-usr-abrt IND • stop all timers ⇒ IDLE
AIDC-P-ABORT IND	• AIDC-pvd-abrt IND • stop all timers ⇒ IDLE	• AIDC-pvd-abrt IND • stop all timers ⇒ IDLE	• AIDC-pvd-abrt IND • stop all timers ⇒ IDLE	• AIDC-pvd-abrt IND • stop all timers ⇒ IDLE

State→ Event↓	PRE-TRANSFER	TRANSFERRING	TRANSFERRED	BACKWARD COORDINATING
AIDC-ucf REQ	if c2 { • AIDC-DATA REQ with AIDC-ucf-apdu • if c1 & c22 ⇒ TRANSFERRING • if c1 & c24 { • start t_{TE} ⇒ TRANSFERRED } • if !c1 { • if !c38 start t_{2R} • vr1 = NULL • vr2 = NULL • vre = NULL ⇒ PRE-TRANSFER } } if !c2 { • AIDC-pvd-abrt IND • AIDC-ABORT REQ • stop all timers • IDLE }	if c2 { • AIDC-DATA REQ with AIDC-ucf-apdu • if c1 & c24 { • start t_{TE} ⇒ TRANSFERRED } • if c1 & c18 { • start t_{TE} ⇒ TRANSFERRED } • if c1 & c20 { • start t_{ICT} ⇒ COORDINATED } • if !c1 { • if !c38 start t_{2R} • vr1 = NULL • vr2 = NULL • vre = NULL ⇒ PRE-TRANSFER } } if !c2 { • AIDC-pvd-abrt IND • AIDC-ABORT REQ • stop all timers • IDLE }	if c2 { • AIDC-DATA REQ with AIDC-ucf-apdu • if c1 & c26 ⇒ BACKWARD COORDINATING • if c1 & c30 ⇒ IDLE • if !c1 { • if !c38 start t_{2R} • vr1 = NULL • vr2 = NULL • vre = NULL ⇒ PRE-TRANSFER } } if !c2 { • AIDC-pvd-abrt IND • AIDC-ABORT REQ • stop all timers • IDLE }	if c2 { • AIDC-DATA REQ with AIDC-ucf-apdu • if c1 & c28 ⇒ TRANSFERRED • if !c1 { • if !c38 start t_{2R} • vr1 = NULL • vr2 = NULL • vre = NULL ⇒ PRE-TRANSFER } } if !c2 { • AIDC-pvd-abrt IND • AIDC-ABORT REQ • stop all timers • IDLE }
rcv AIDC-ucf-REQ pdu	if c3 { • AIDC-ucf IND • if c1 & c43 { • start t_{1R} ⇒ PRE-TRANSFER } • if c1 & c23 ⇒ TRANSFERRING • if c1 & c25 ⇒ TRANSFERRED • if !c1 { • stop all timers • vs1=NULL • vs2=NULL • vse=NULL ⇒ PRE-TRANSFER } } if !c3 { • AIDC-p • AIDC-ABORT REQ • stop all timers ⇒ IDLE }	if c3 { • AIDC-ucf IND • if c1 & c25 ⇒ TRANSFERRED • if c1 & c19 ⇒ TRANSFERRED • if c1 & c21 { • start t_{2CT} ⇒ COORDINATED } • if !c1 { • stop all timers • vs1=NULL • vs2=NULL • vse=NULL ⇒ TRANSFERRING } } if !c3 { • AIDC-pvd-abrt IND • AIDC-ABORT REQ • stop all timers ⇒ IDLE }	if c3 { • AIDC-ucf IND • if c1 & c27 ⇒ BACKWARD COORDINATING • if c1 & c31 ⇒ IDLE • if !c1 { • stop all timers • vs1=NULL • vs2=NULL • vse=NULL ⇒ TRANSFERRED } } if !c3 { • AIDC-pvd-abrt IND • AIDC-ABORT REQ • stop all timers ⇒ IDLE }	if c3 { • AIDC-ucf IND • if c1 & c29 ⇒ TRANSFERRED • if !c1 { • stop all timers • vs1=NULL • vs2=NULL • vse=NULL ⇒ BACKWARD COORDINATING } } if !c3 { • AIDC-pvd-abrt IND • AIDC-ABORT REQ • stop all timers ⇒ IDLE }

3.2.7 AIDC FORMAL DEFINITIONS

Note.—The following defines the ASN.1 [ISO/IEC 8824-1] abstract syntax for the AIDC-AE.

3.2.7.1 AIDC ASN.1 Abstract Syntax

3.2.7.1.1 Each AIDC-APDU shall conform to the abstract syntax as defined below:

AIDCMessageSetVersion1 DEFINITIONS ::=

BEGIN

 -- AIDC-APDU

AIDC-APDU ::= CHOICE

{		
aidc-ucf-apdu	[0]	AIDC-ucf-apdu,
aidc-nfy-apdu	[1]	AIDC-nfy-apdu,
aidc-crd-start-apdu	[2]	AIDC-crd-start-apdu,
aidc-crd-end-apdu	[3]	AIDC-crd-end-apdu,
aidc-crd-ngtt-apdu	[4]	AIDC-crd-ngtt-apdu,
aidc-crd-stndby-apdu	[5]	AIDC-crd-stndby-apdu,
aidc-tfr-init-apdu	[6]	AIDC-tfr-init-apdu,
aidc-tfr-rqst-apdu	[7]	AIDC-tfr-rqst-apdu,
aidc-tfr-prpsl-apdu	[8]	AIDC-tfr-prpsl-apdu,
aidc-tfr-accept-apdu	[9]	AIDC-tfr-accept-apdu,
aidc-tfr-cntrl-req-apdu	[10]	AIDC-tfr-cntrl-Req-apdu,
aidc-tfr-cntrl-rsp-apdu	[11]	AIDC-tfr-cntrl-Rsp-apdu,
aidc-tfr-comm-apdu	[12]	AIDC-tfr-comm-apdu,
aidc-tfr-comm-assm-apdu	[13]	AIDC-tfr-comm-assm-apdu,
aidc-inf-tfr-apdu	[14]	AIDC-inf-tfr-apdu,
aidc-end-apdu	[15]	AIDC-end-apdu,
...		
}		

 -- AIDC-apdu

AIDC-ucf-apdu ::= SEQUENCE

{			
result	[0]	Result,	
reason	[1]	ApplicationErrorData	OPTIONAL,
referenceid	[2]	MessageNumber	
}			

AIDC-nfy-apdu ::= SEQUENCE

{			
calledICAOFacilityDesignation	[0]	ICAOFacilityDesignation,	
callingICAOFacilityDesignation	[1]	ICAOFacilityDesignation	OPTIONAL,
notify	[2]	Notify,	
msgnumber	[3]	MessageNumber	
}			

AIDC-crd-start-apdu ::= SEQUENCE

{			
calledICAOFacilityDesignation	[0]	ICAOFacilityDesignation,	
callingICAOFacilityDesignation	[1]	ICAOFacilityDesignation	OPTIONAL,
startdata	[2]	Startdata,	
msgnumber	[3]	MessageNumber	
}			

Startdata ::= CHOICE

{			
coordinateinitial	[0]	CoordinateInitial,	
coordinateupdate	[1]	CoordinateUpdate	
}			

AIDC-crd-end-apdu ::= SEQUENCE

{			
enddata	[0]	Enddata,	
result	[1]	Result,	
msgnumber	[2]	MessageNumber	
}			

Enddata ::= CHOICE

{			
coordinateaccept	[0]	CoordinateAccept,	
coordinatereject	[1]	CoordinateReject	
}			

AIDC-crd-ngtt-apdu ::= SEQUENCE

{			
coordinatenegotiate	[0]	CoordinateNegotiate,	
msgnumber	[1]	MessageNumber	
}			

AIDC-crd-stndby-apdu ::= SEQUENCE

{			
coordinatestandby	[0]	CoordinateStandby,	
msgnumber	[1]	MessageNumber	
}			

AIDC-tfr-init-apdu ::= SEQUENCE		
{		
transferinitiate	[0]	TransferInitiate,
msgnumber	[1]	MessageNumber
}		
AIDC-tfr-rqst-apdu ::= SEQUENCE		
{		
transferrequest	[0]	TransferRequest,
msgnumber	[1]	MessageNumber
}		
AIDC-tfr-prpsl-apdu ::= SEQUENCE		
{		
transferconditionsproposal	[0]	TransferConditionsProposal,
msgnumber	[1]	MessageNumber
}		
AIDC-tfr-accept-apdu ::= SEQUENCE		
{		
transferconditionsaccept	[0]	TransferConditionsAccept,
msgnumber	[1]	MessageNumber
}		
AIDC-tfr-cntrl-Req-apdu ::= SEQUENCE		
{		
transfercontrol	[0]	TransferControl,
msgnumber	[1]	MessageNumber
}		
AIDC-tfr-cntrl-Rsp-apdu ::= SEQUENCE		
{		
transfercontroldata	[0]	TransferControlData,
result	[1]	Result,
msgnumber	[2]	MessageNumber
}		
TransferControlData ::= CHOICE		
{		
transfercontrolassume	[0]	TransferControlAssume,
transfercontrolreject	[1]	TransferControlReject
}		
AIDC-tfr-comm-apdu ::= SEQUENCE		
{		
transfercomm	[0]	TransferComm,
msgnumber	[1]	MessageNumber
}		

AIDC-tfr-comm-assm-apdu ::= SEQUENCE

{		
transfercommassume	[0]	TransferCommAssume,
msgnumber	[1]	MessageNumber
}		

AIDC-inf-tfr-apdu ::= SEQUENCE

{		
infodata	[0]	InfoData,
msgnumber	[1]	MessageNumber
}		

InfoData ::= CHOICE

{		
generalexecutivedata	[0]	GeneralExecutiveData,
generalpoint	[1]	GeneralPoint,
surveillancegeneral	[2]	SurveillanceGeneral,
generalfreetext	[3]	GeneralFreeText,
emergencyfreetext	[4]	EmergencyFreeText,
...		
}		

AIDC-end-apdu ::= SEQUENCE

{			
cancel	[0]	Cancel	OPTIONAL,
msgnumber	[1]	MessageNumber	
}			

MessageType ::= ENUMERATED

{	
aidc-ucf-apdu	(0),
aidc-nfy-apdu	(1),
aidc-crd-start-apdu	(2),
aidc-crd-end-apdu	(3),
aidc-crd-ngtt-apdu	(4),
aidc-crd-stndby-apdu	(5),
aidc-tfr-init-apdu	(6),
aidc-tfr-rqst-apdu	(7),
aidc-tfr-prpsl-apdu	(8),
aidc-tfr-accept-apdu	(9),
aidc-tfr-cntrl-req-apdu	(10),
aidc-tfr-cntrl-rsp-apdu	(11),
aidc-tfr-comm-apdu	(12),
aidc-tfr-comm-assm-apdu	(13),
aidc-inf-tfr-apdu	(14),
aidc-end-apdu	(15),
...	
}	

-- AIDC MESSAGE DEFINITIONS

Notify ::= SEQUENCE

{			
flightID	[0]	FlightID,	
departure	[1]	DepartureAirportTime,	
destination	[2]	DestinationAirport	OPTIONAL,
flightRuleFlightType	[3]	FlightRuleFlightType	OPTIONAL,
beaconCode	[4]	BeaconCode	OPTIONAL,
aircraftNumberType	[5]	AircraftNumberType,	
cnsEquipment	[6]	CNSEquipment	OPTIONAL,
boundaryEstimate	[7]	BoundaryEstimate,	
route	[8]	Route	OPTIONAL,
otherInfo	[9]	OtherInformation	OPTIONAL,
timestamp	[10]	YMDHMS,	
...			
}			

CoordinateInitial ::= SEQUENCE

{			
flightID	[0]	FlightID,	
departure	[1]	DepartureAirportTime,	
destination	[2]	DestinationAirport	OPTIONAL,
flightRuleFlightType	[3]	FlightRuleFlightType	OPTIONAL,
beaconCode	[4]	BeaconCode	OPTIONAL,
aircraftNumberType	[5]	AircraftNumberType,	
cnsEquipment	[6]	CNSEquipment	OPTIONAL,
boundaryEstimate	[7]	BoundaryEstimate,	
route	[8]	Route	OPTIONAL,
otherInfo	[9]	OtherInformation	OPTIONAL,
timestamp	[10]	YMDHMS,	
...			
}			

CoordinateUpdate ::= SEQUENCE

{			
flightID	[0]	FlightID,	
departure	[1]	DepartureAirportTime,	
destination	[2]	DestinationAirport	OPTIONAL,
beaconCode	[3]	BeaconCode	OPTIONAL,
boundaryEstimate	[4]	BoundaryEstimate,	
route	[5]	Route	OPTIONAL,
timestamp	[6]	YMDHMS,	
...			
}			

CoordinateNegotiate ::= SEQUENCE

{			
flightID	[0]	FlightID,	
departure	[1]	DepartureAirportTime,	
destination	[2]	DestinationAirport	OPTIONAL,
boundaryEstimate	[3]	BoundaryEstimate,	
route	[4]	Route	OPTIONAL,
timestamp	[5]	YMDHMS,	
...			
}			

CoordinateStandby ::= SEQUENCE

{			
flightID	[0]	FlightID,	
departure	[1]	DepartureAirportTime,	
destination	[2]	DestinationAirport	OPTIONAL,
timestamp	[3]	YMDHMS,	
...			
}			

CoordinateAccept ::= SEQUENCE

{			
flightID	[0]	FlightID,	
departure	[1]	DepartureAirportTime,	
destination	[2]	DestinationAirport	OPTIONAL,
frequency	[3]	Frequency	OPTIONAL,
timestamp	[4]	YMDHMS,	
...			
}			

CoordinateReject ::= SEQUENCE

{			
flightID	[0]	FlightID,	
departure	[1]	DepartureAirportTime,	
destination	[2]	DestinationAirport	OPTIONAL,
timestamp	[3]	YMDHMS,	
...			
}			

TransferInitiate ::= SEQUENCE

{			
flightID	[0]	FlightID,	
departure	[1]	DepartureAirportTime,	
destination	[2]	DestinationAirport	OPTIONAL,
executiveData	[3]	ExecutiveData	OPTIONAL,
trackData	[4]	TrackData	OPTIONAL,
timestamp	[5]	YMDHMS,	
...			
}			

TransferConditionsProposal ::= SEQUENCE

{			
flightID	[0]	FlightID,	
departure	[1]	DepartureAirportTime,	
destination	[2]	DestinationAirport	OPTIONAL,
executiveData	[3]	ExecutiveData	OPTIONAL,
timestamp	[4]	YMDHMS,	
...			
}			

TransferConditionsAccept ::= SEQUENCE

{			
flightID	[0]	FlightID,	
departure	[1]	DepartureAirportTime,	
destination	[2]	DestinationAirport	OPTIONAL,
frequency	[3]	Frequency	OPTIONAL,
timestamp	[4]	YMDHMS,	
...			
}			

TransferRequest ::= SEQUENCE

{			
flightID	[0]	FlightID,	
departure	[1]	DepartureAirportTime,	
destination	[2]	DestinationAirport	OPTIONAL,
frequency	[3]	Frequency	OPTIONAL,
timestamp	[4]	YMDHMS,	
...			
}			

TransferControl ::= SEQUENCE

{			
flightID	[0]	FlightID,	
departure	[1]	DepartureAirportTime,	
destination	[2]	DestinationAirport	OPTIONAL,
executiveData	[3]	ExecutiveData	OPTIONAL,
timestamp	[4]	YMDHMS,	
...			
}			

TransferControlAssume ::= SEQUENCE

{			
flightID	[0]	FlightID,	
departure	[1]	DepartureAirportTime,	
destination	[2]	DestinationAirport	OPTIONAL,
timestamp	[3]	YMDHMS,	
...			
}			

TransferControlReject ::= SEQUENCE

{			
flightID	[0]	FlightID,	
departure	[1]	DepartureAirportTime,	
destination	[2]	DestinationAirport	OPTIONAL,
timestamp	[3]	YMDHMS,	
...			
}			

TransferComm ::= SEQUENCE

{			
flightID	[0]	FlightID,	
departure	[1]	DepartureAirportTime,	
destination	[2]	DestinationAirport	OPTIONAL,
executiveData	[3]	ExecutiveData	OPTIONAL,
releaseIndicator	[4]	ReleaseIndicator	OPTIONAL,
timestamp	[5]	YMDHMS,	
...			
}			

TransferCommAssume ::= SEQUENCE

{			
flightID	[0]	FlightID,	
departure	[1]	DepartureAirportTime,	
destination	[2]	DestinationAirport	OPTIONAL,
timestamp	[3]	YMDHMS,	
...			
}			

SurveillanceGeneral ::= SEQUENCE

{			
flightID	[0]	FlightID,	
departure	[1]	DepartureAirportTime	OPTIONAL,
destination	[2]	DestinationAirport	OPTIONAL,
trackData	[3]	TrackData,	
timestamp	[4]	YMDHMS,	
...			
}			

GeneralPoint ::= SEQUENCE

{			
functionalAddress	[0]	FunctionalAddress	OPTIONAL,
flightID	[1]	FlightID,	
departure	[2]	DepartureAirportTime	OPTIONAL,
destination	[3]	DestinationAirport	OPTIONAL,
flightRuleFlightType	[4]	FlightRuleFlightType,	
beaconCode	[5]	BeaconCode	OPTIONAL,
aircraftNumberType	[6]	AircraftNumberType,	
cnsEquipment	[7]	CNSEquipment,	
boundaryEstimate	[8]	BoundaryEstimate	OPTIONAL,
route	[9]	Route	OPTIONAL,
otherInfo	[10]	OtherInformation	OPTIONAL,
timestamp	[11]	YMDHMS,	
...			
}			

GeneralExecutiveData ::= SEQUENCE

{			
flightID	[0]	FlightID,	
frequency	[1]	Frequency,	
executivedata	[2]	ExecutiveData,	
timestamp	[4]	YMDHMS,	
...			
}			

EmergencyFreeText ::= SEQUENCE

{			
functionalAddress	[0]	FunctionalAddress	OPTIONAL,
flightID	[1]	FlightID,	
freeText	[2]	FreeText,	
timestamp	[3]	YMDHMS,	
...			
}			

GeneralFreeText ::= SEQUENCE

{			
functionalAddress	[0]	FunctionalAddress	OPTIONAL,
flightID	[1]	FlightID,	
freeText	[2]	FreeText,	
timestamp	[3]	YMDHMS,	
...			
}			

Cancel ::= SEQUENCE

{			
flightID	[0]	FlightID,	
departure	[1]	DepartureAirportTime,	
destination	[2]	DestinationAirport	OPTIONAL,
boundaryEstimate	[3]	BoundaryEstimate	OPTIONAL,
otherinfo	[4]	OtherInformation	OPTIONAL,
timestamp	[5]	YMDHMS,	
...			
}			

-- AIDC MESSAGE ELEMENTS

ProviderAbortReason ::= ENUMERATED

{	
protocolerror	(0),
timerexpired	(1),
undefinederror	(2),
providererror	(3),
rejectedpermanent	(4),
rejectedtransient	(5),
...	
}	

AircraftNumberType ::= SEQUENCE

{			
numberOfAircraft	[0]	NumberOfAircraft	OPTIONAL,
aircraftType	[1]	AircraftType,	
wakeTurbulenceCategory	[2]	WakeTurbulenceCategory	OPTIONAL
}			

AircraftIdentification ::= IA5String (SIZE(2..7))

AircraftType ::= IA5String (SIZE(2..4))

AircraftAddress ::= BIT STRING (SIZE(24))

Airport ::= IA5String (SIZE(4))

DepartureAirportTime ::= SEQUENCE

{			
airport	[0]	Airport,	
time	[1]	Time	OPTIONAL
}			

ATSRouteDesignator ::= IA5String (SIZE(2..7))

Level ::= CHOICE

{		
levelFeet	[0]	LevelFeet,
levelMetre	[1]	LevelMetre,
levelFlightLevel	[2]	LevelFlightLevel,
levelFlightLevelMetric	[3]	LevelFlightLevelMetric
}		

LevelFeet ::= INTEGER (-60..7000)

-- unit = Feet, Range (-600..70000), resolution = 10

LevelMetre ::= INTEGER (-30..25000)

-- unit = metre, Range (-30..25000), resolution = 1

LevelFlightLevel ::= INTEGER (30..700)

-- unit = Level (100 feet), Range (30..700), resolution = 1

LevelFlightLevelMetric ::= INTEGER (100..2500)

-- unit = Level (10 metres), Range (100..2500), resolution = 1

ApplicationErrorData ::= SEQUENCE

{			
messageType	[0]	MessageType,	
componentType	[1]	ComponentType,	
errorCode	[2]	ErrorCode,	
errorData	[3]	ErrorData	OPTIONAL
}			

ATWLevel ::= SEQUENCE

{		
atw	[0]	ATWLevelTolerance,
level	[1]	Level
}		

ATWLevelTolerance ::= ENUMERATED

{	
at	(0),
atorabove	(1),
atorbelow	(2)
}	

BeaconCode ::= SEQUENCE SIZE (4) OF BeaconCodeOctalDigit

BeaconCodeOctalDigit ::= INTEGER (0..7)

BoundaryEstimate ::= SEQUENCE

{			
boundaryFix	[0]	Position,	
crossingTime	[1]	Time,	
crossingLevel	[2]	Level,	
atwLevel	[3]	ATWLevel	OPTIONAL
}			

CNSEquipment ::= SEQUENCE

{			
comNavEquipmentStatus	[0]	SEQUENCE SIZE (0..24) OF ComNavEquipmentStatus	OPTIONAL,
ssrEquipmentAvailable	[1]	SSREquipmentAvailable,	
adsAvailable	[2]	BOOLEAN,	
acasAvailable	[3]	BOOLEAN,	
dataLink	[4]	SEQUENCE SIZE (0..4) OF DataLink	
}			

ComNavEquipmentStatus ::= ENUMERATED

{		
aloranA	(0),	
cloranC	(1),	
ddme	(2),	
edecca	(3),	
fadf	(4),	
ggnss	(5),	
hhfRtf	(6),	
iinertialNavigation	(7),	
lils	(8),	
momega	(9),	
ovor	(10),	
pdoppler	(11),	
rrnavRouteEquipment	(12),	
ttacan	(13),	
uuhfRTF	(14),	
vvhfRTF	(15),	
...		
}		

DataLink ::= ENUMERATED

{		
hf	(0),	
modeS	(1),	
satcom	(2),	
vhf	(3)	
}		

Date ::= SEQUENCE

{		
year	[0]	Year,
month	[1]	Month,
day	[2]	Day
}		

Day ::= INTEGER (1..31)

-- unit = Day, Range (1..31), resolution = 1

Degrees ::= CHOICE

{		
degreesMagnetic	[0]	DegreesMagnetic,
degreesTrue	[1]	DegreesTrue
}		

DegreesMagnetic ::= INTEGER (1..360)

-- unit = degree, Range (1..360), resolution = 1

DegreeMinutes ::= INTEGER (0..5999)

-- unit = Minute, Range (0..59.99), resolution = 0.01

DegreeSeconds ::= INTEGER (0..59)

-- unit = Second, Range (0..59), resolution = 1

DegreesTrue ::= INTEGER (1..360)

-- unit = degree, Range (1..360), resolution = 1

DestinationAirport ::= Airport

DirectRouting ::= SEQUENCE

{			
fix2	[0]	Position	OPTIONAL,
fix1	[1]	Position	
}			

Distance ::= CHOICE

{		
distanceNM	[0]	DistanceNM,
distancekm	[1]	Distancekm
}		

Distancekm ::= INTEGER (0..2000)

-- unit = kilometre, Range (0..2000), resolution = 1

DistanceNM ::= INTEGER (0..1000)

-- unit = Nautical Mile, Range (0..1000), resolution = 1

ExecutiveData ::= SEQUENCE

{			
speed	[0]	Speed	OPTIONAL,
level	[1]	Level	OPTIONAL,
heading	[2]	DegreesMagnetic	OPTIONAL,
vertRate	[3]	VerticalChange	OPTIONAL,
directRouting	[4]	DirectRouting	OPTIONAL
}			

FixName ::= IA5String (SIZE(1..5))**FlightID** ::= SEQUENCE

{			
aircraftIdentification	[0]	AircraftIdentification,	
selcal	[1]	Selcal	OPTIONAL,
registration	[2]	Registration	OPTIONAL,
airframeID	[3]	AircraftAddress	OPTIONAL
}			

FlightRule ::= ENUMERATED

{	
ifr	(0),
vfr	(1),
ifrfirst	(2),
vfrfirst	(3)
}	

FlightRuleFlightType ::= SEQUENCE

{		
flightRule	[0]	FlightRule,
flightType	[1]	FlightType
}		

FlightType ::= ENUMERATED

{	
scheduledAirTransport	(0),
nonScheduledAirTransport	(1),
generalAviation	(2),
military	(3),
otherFlights	(4)
}	

FreeText ::= IA5String (SIZE(1..256))

Frequency ::= CHOICE

{		
frequencyHF	[0]	FrequencyHF,
frequencyVHF	[1]	FrequencyVHF,
frequencyUHF	[2]	FrequencyUHF,
frequencySatChannel	[3]	FrequencySatChannel
}		

FrequencyHF ::= INTEGER (2850..28000)

-- unit = Kilohertz, Range (2850..28000), resolution = 1

FrequencyVHF ::= INTEGER (23600..27398)

-- unit = Megahertz, Range (118.000..136.990), resolution = 0.005

FrequencyUHF ::= INTEGER (9000..15999)

-- unit = Megahertz, Range (225.000..399.975), resolution = 0.025

FrequencySatChannel ::= NumericString (SIZE(12))

-- FrequencySatChannel corresponds to a 12-digit telephone number

FunctionalAddress ::= IA5String (SIZE(1..18))

ICAOFacilityDesignation ::= IA5String (SIZE(4..8))

Latitude ::= SEQUENCE

{			
latitudeDegrees	[0]	LatitudeDegrees,	
latitudeMinutes	[1]	DegreeMinutes	OPTIONAL,
latitudeSeconds	[2]	DegreeSeconds	OPTIONAL,
latitudeDirection	[3]	LatitudeDirection	
}			

LatitudeDegrees ::= INTEGER (0..90000)

-- unit = Degree, Range (0..90), resolution = 0.001

LatitudeDirection ::= ENUMERATED

{	
north	(0),
south	(1)
}	

LatitudeLongitude ::= SEQUENCE

{		
latitude	[0]	Latitude,
longitude	[1]	Longitude
}		

Longitude ::= SEQUENCE

{			
longitudeDegrees	[0]	LongitudeDegrees,	
longitudeMinutes	[1]	DegreeMinutes	OPTIONAL,
longitudeSeconds	[2]	DegreeSeconds	OPTIONAL,
longitudeDirection	[3]	LongitudeDirection	
}			

LongitudeDegrees ::= INTEGER (0..180000)

-- unit = Degree, Range (0..180), resolution = 0.001

LongitudeDirection ::= ENUMERATED

{	
east	(0),
west	(1)
}	

MessageNumber ::= INTEGER (0..999999)

Month ::= INTEGER (1..12)

-- unit = Month, Range (1..12), resolution = 1

Navaid ::= IA5String (SIZE(1..4))

NumberOfAircraft ::= INTEGER (1..2)

OtherInformation ::= FreeText

PlaceBearing ::= SEQUENCE

{			
fixName	[0]	FixName,	
latitudeLongitude	[1]	LatitudeLongitude	OPTIONAL,
degrees	[2]	Degrees	
}			

PlaceBearingDistance ::= SEQUENCE

{		
placeBearing	[0]	PlaceBearing,
distance	[1]	Distance
}		

PlaceBearingPlaceBearing ::= SEQUENCE SIZE (2) OF PlaceBearing

Position ::= CHOICE

{		
fixName	[0]	FixName,
navaid	[1]	Navaid,
airport	[2]	Airport,
latitudeLongitude	[3]	LatitudeLongitude,
placeBearingDistance	[4]	PlaceBearingDistance
}		

PublishedIdentifier ::= SEQUENCE

{		
fixName	[0]	FixName,
latitudeLongitude	[1]	LatitudeLongitude
}		

Registration ::= IA5String (SIZE(7))

ReleaseIndicator ::= ENUMERATED

{		
climb	(0),	
descent	(1),	
turns	(2),	
allActions	(3)	
}		

Result ::= ENUMERATED

{		
accepted	(0),	
rejected	(1),	
...		
}		

Route ::= SEQUENCE

{		
SEQUENCE SIZE (1..128) OF RouteInformation,		
position	[0]	Position,
time	[1]	Time,
level	[2]	Level,
speedGround	[3]	SpeedGround,
trueTrackAngle	[4]	TrueTrackAngle
}		

RouteInformation ::= CHOICE

{		
publishedIdentifier	[0]	PublishedIdentifier,
latitudeLongitude	[1]	LatitudeLongitude,
placeBearingPlaceBearing	[2]	PlaceBearingPlaceBearing,
placeBearingDistance	[3]	PlaceBearingDistance,
aTSRouteDesignator	[4]	ATSRouteDesignator,
trackDetail	[5]	TrackDetail
}		

Selcal ::= IA5String (SIZE(4))

Speed ::= CHOICE

{		
speedGround	[0]	SpeedGround,
speedGroundMetric	[1]	SpeedGroundMetric,
speedMach	[2]	SpeedMach,
speedIndicated	[3]	SpeedIndicated,
speedIndicatedMetric	[4]	SpeedIndicatedMetric,
speedTrue	[5]	SpeedTrue,
speedTrueMetric	[6]	SpeedTrueMetric
}		

SpeedGround ::= INTEGER (-50..2000)

-- unit = Knots, Range (-50..2000), resolution = 1

SpeedGroundMetric ::= INTEGER (-100..4000)

-- unit = kilometre/hour, Range (-100..4000), resolution = 1

SpeedIndicated ::= INTEGER (0..400)

-- unit = Knots, Range (0..400), resolution = 1

SpeedIndicatedMetric ::= INTEGER (0..800)

-- unit =kilometre/hour, Range (0..800), resolution = 1

SpeedMach ::= INTEGER (500..4000)

-- unit = Mach, Range (0.5..4.0), resolution = 0.001

SpeedTrue ::= INTEGER (0..2000)

-- unit = Knots, Range (0..2000), resolution = 1

SpeedTrueMetric ::= INTEGER (0..4000)

-- unit =kilometre/hour, Range (0..4000), resolution = 1

SSREquipmentAvailable ::= ENUMERATED

{	
nnil	(0),
atransponderModeA	(1),
ctransponderModeAandC	(2),

```

xatransponderModeS      (3),
ptransponderModeSPA     (4),
itransponderModeSID     (5),
satransponderModeSPAID  (6),
...
}
-- PA: Pressure Level; ID: Aircraft Identification

```

Time ::= SEQUENCE

```

{
  hours           [0]    TimeHours,
  minutes         [1]    TimeMinutes
}

```

Timehhmmss ::= SEQUENCE

```

{
  hoursminute      Time,
  seconds          TimeSeconds
}

```

TimeHours ::= INTEGER (0..23)

-- unit = Hour, Range (0..23), resolution = 1

TimeMinutes ::= INTEGER (0..59)

-- unit = Minute, Range (0..59), resolution = 1

TimeSeconds ::= INTEGER(0..59)

-- unit= Second, Range (0..59), resolution = 1

TrackData ::= SEQUENCE

```

{
  position          [0]    Position,
  time              [1]    Time,
  level             [2]    Level,
  speedGround       [3]    SpeedGround,
  trueTrackAngle    [4]    TrueTrackAngle
}

```

TrackDetail ::= SEQUENCE

```

{
  trackName         [0]    TrackName,
  latitudeLongitude [1]    LatitudeLongitude
}

```

TrackName ::= IA5String (SIZE(1..6))

TrueTrackAngle ::= Degrees

VerticalChange ::= SEQUENCE

{		
direction	[0]	VerticalDirection,
rate	[1]	VerticalRate
}		

VerticalDirection ::= ENUMERATED

{	
up	(0),
down	(1)
}	

VerticalRate ::= CHOICE

{		
verticalRateEnglish	[0]	VerticalRateEnglish,
verticalRateMetric	[1]	VerticalRateMetric
}		

VerticalRateEnglish ::= INTEGER (0..3000)

-- unit = Feet/Minute, Range (0..30000), resolution = 10

VerticalRateMetric ::= INTEGER (0..1000)

-- unit = metre/Minute, Range (0..1000), resolution = 1

WakeTurbulenceCategory ::= ENUMERATED

{	
high	(0),
medium	(1),
low	(2)
}	

Year ::= INTEGER (1996..2095)

-- unit = Year, Range (1996..2095), resolution = 1

YMDHMS ::= SEQUENCE

{		
date	[0]	Date,
timehhmmss	[1]	Timehhmmss
}		

 -- AIDC ERROR-RELATED TYPES

ComponentType ::= ENUMERATED

{	
ctUnknown	(0),
ctNotApplicable	(1),

ctAircraftNumberType	(2),
ctBeaconCode	(3),
ctBoundaryEstimate	(4),
ctCNSEquipment	(5),
ctDepartureAirportTime	(6),
ctDestinationAirport	(7),
ctExecutiveData	(8),
ctFlightID	(9),
ctFlightRuleFlightType	(10),
ctFreeText	(11),
ctFrequency	(12),
ctFunctionalAddress	(13),
ctReleaseIndicator	(14),
ctRoute	(15),
ctTrackData	(16),
ctUnrecognised	(255),
...	
}	

ErrorCode ::= ENUMERATED

{	
invalidNumberOfAircraft	(0),
invalidAircraftType	(1),
invalidWakeTurbulenceCategory	(2),
invalidBeaconCodeOctalDigit	(3),
invalidFixName	(4),
invalidNavaid	(5),
invalidAirport	(6),
invalidLatitude	(7),
invalidLongitude	(8),
invalidTime	(9),
invalidLevelFeet	(10),
invalidLevelMetre	(11),
invalidLevelFlightLevel	(12),
invalidLevelFlightLevelMetric	(13),
invalidATWLevelTolerance	(14),
invalidComNavEquipmentStatus	(15),
invalidSSREquipmentAvailable	(16),
invalidDataLink	(17),
invalidSpeedGround	(18),
invalidSpeedGroundMetric	(19),
invalidSpeedMach	(20),
invalidSpeedIndicated	(21),
invalidSpeedIndicatedMetric	(22),
invalidSpeedTrue	(23),
invalidSpeedTrueMetric	(24),
invalidVerticalDirection	(25),
invalidVerticalRateEnglish	(26),
invalidVerticalRateMetric	(27),

invalidAircraftIdentification	(28),
invalidSelcal	(29),
invalidRegistration	(30),
invalidAircraftAddress	(31),
invalidFlightRule	(32),
invalidFlightType	(33),
invalidFrequencyHF	(34),
invalidFrequencyVHFChannel	(35),
invalidFrequencyUHF	(36),
invalidFrequencySatChannel	(37),
invalidFunctionalAddress	(38),
invalidReleaseIndicator	(39),
invalidDistancekm	(40),
invalidDistanceNM	(41),
invalidATSRouteDesignator	(42),
invalidTrackName	(43),
invalidmsgnumber	(250),
invalidreferenceid	(251),
invalidcallingICAOFacilityDesignation	(252),
invalidcalledICAOFacilityDesignation	(253),
invalidtimestamp	(254),
unknown	(255),
...	
}	

ErrorData ::= BIT STRING (SIZE(1..256))

END

3.2.8 COMMUNICATION REQUIREMENTS

3.2.8.1 Encoding Rules

3.2.8.1.1 The AIDC application shall use PER as defined in ISO/IEC 8825-2, using the Basic Unaligned variant to encode/decode the ASN.1 message structure and content specified in 3.2.7, or a functionally equivalent means which provides the same result.

3.2.8.2 Quality-of-Service Requirements

3.2.8.2.1 Routing Policy

3.2.8.2.1.1 Routing Class shall be conveyed by local means, using the values for Security Tag Value specified in Table 5.6-1.

3.2.8.2.1.2 For the AIDC application no routing class is specified, thus the value corresponding to “ATSC: No Traffic Type Policy Preference” shall be conveyed.

Note.— It is stated in 5.2.7.3.1, “The mechanism by which the connection initiator determines the appropriate ATN Security Label is a local matter. For example, it may be identified by an extension to the transport service interface, be implicit in the choice of a given TSAP, or be identified using a Systems Management function.”

3.2.8.2.2 Priority

3.2.8.2.2.1 The AIDC application service priority shall have the value corresponding to “normal priority flight safety messages”.

3.2.8.2.2.2 Priority shall map to the session connection priority component of the A-ASSOCIATE request primitive Quality of Service parameter (see Table 3.2.5-1), using the values for Transport Layer Priority specified in Table 1.2 (see .3.8).

Note.— Although transport priority and network priority are semantically independent of each other, 5.5.1.2 states that the TS-user specifies the Application Service Priority, which in turn is mapped into the resulting CLNP PDUs according to Table 1.2, which defines the fixed relationship between transport priority and the network priority.

3.2.8.2.3 Residual Error Rate

3.2.8.2.3.1 The required AIDC application service RER shall have the value corresponding to “low”.

3.2.8.2.3.2 Thus the residual error rate component of the A-ASSOCIATE request primitive Quality of Service parameter (see Table 3.2.5-1) shall be set to zero.

Note.— 5.5.1.2 states that the transport service user specifies the required residual error rate to determine whether or not the transport checksum is required.

3.2.9 AIDC-USER REQUIREMENTS

Note.— The following identifies the requirements imposed upon the AIDC-User by the AIDC-AE.

3.2.9.1 Inter-Operability

3.2.9.1.1 To achieve inter-operability for the implementation of AIDC between ATS units, or ATS regions on a global basis, the following requirements shall be satisfied:

- a) a common AIDC message set and associated services is agreed upon by the ATS units involved,
- b) agreement, by the ATS units involved, as to what flight related conditions dictate the invocation of the AIDC services,
- c) agreement as to the timing associated with the use of the AIDC services,
- d) agreement as to the predicate P1 as defined in 3.2.6.1.4.

3.2.9.2 Message Handling

3.2.9.2.1 Message Priorities

3.2.9.2.1.1 In cases where message queueing is implemented by the AIDC-User, an application specific priority scheme shall be implemented.

3.2.9.2.1.1.1 Under this scheme the AIDC messages received shall be assigned one of the following priorities:

- a) Normal;
- b) Urgent; or
- c) Distress.

3.2.9.2.1.1.2 Assignment of priorities:

- a) General freetext messages shall be assigned the priority of “Normal”.
- b) Emergency freetext messages shall be assigned the priority of “Distress”.
- c) Surveillance data transfer messages shall be assigned the priority of “Urgent”.
- d) All other AIDC messages shall be assigned the priority “Normal”.

3.2.9.2.1.1.3 The AIDC-User shall process messages with a priority of Distress first, followed by messages with a priority of Urgent and then messages with a priority of Normal.

3.2.9.3 Operational Timers

3.2.9.3.1 The AIDC-User shall manage a number of timers associated with the sending of AIDC messages. These timers are described below:

- a) a Confirmation Timer, used to detect the failure of the AIDC-User to receive the user confirmation for a previously sent message;
- b) a Response Timer, used to detect the failure of the AIDC-User to receive the appropriate message in response to a previously sent coordination or transfer message;
- c) a Monitor Timer, used to detect the failure of the AIDC-User to receive another expected AIDC message; and
- d) a Standby Timer, used to extend the time before a response to a coordination message is expected.

3.2.9.4 AIDC-AE Specific Requirements

3.2.9.4.1 Management of AIDC-AE Instantiations

3.2.9.4.1.1 As each instantiation of the AIDC-AE is on a flight by flight basis, the AIDC-User shall manage these instantiations by some local means.

3.2.9.4.2 The User-confirmation Service

3.2.9.4.2.1 Upon the receipt of an AIDC message, the AIDC-User shall validate the semantics of the message and use the User-confirmation service to indicate to the peer AIDC-User whether the message has been accepted or rejected.

3.2.9.4.2.2 In order for the AIDC-AE to correlate user service primitives with the User-confirmation primitives, the AIDC-User shall generate and manage a set of unique identifiers.

3.2.9.4.2.3 The identifiers received by an AIDC-User in indication service primitives, shall be used as the *Referenced Number* parameter of the User-confirmation request primitive.

3.2.9.4.3 Coordination Messages

Note.— The Coordinate-start service is used for the initial flight coordination and the updating of the coordination conditions for a flight. The passing of either the CoordinateInitial or CoordinateUpdate message when using the service is under user control and procedures.

3.2.9.4.3.1 If for some reason the D-ATSU AIDC-User cannot accept the type of message received, it shall signal that condition with a negative User-confirmation

3.2.9.5 Error Processing Requirements

3.2.9.5.1 In the event of information input by the user being incompatible with that able to be processed by the system, the user shall be notified.

3.2.10 SEQUENCE DIAGRAMS

3.2.10.1 Sequence Diagrams

3.2.10.1.1 On the invocation of an AIDC-User request primitives or on the receipt of AIDC-ASE primitives from the underlying communication service provider, the AIDC-AE shall ensure that the appropriate sequencing of primitives shown in the following figures, is enforced.

Note 1.— The sequence diagrams, shown below, do not mandate the user sequencing of primitives. Nor do they cover all possible sequences.

Note 2.— The following figures, show the sequence of AIDC-User primitives and the CF mapping of those primitives to/from the primitives of the AIDC-ASE.

Note 3.— As the User-confirmation service is common to all services, except the abort services, the invocation of the primitives are shown as a dotted line.

Note 4.— The “+” symbol preceding the User-confirmation primitive indicates that the user has validated and accepted the contents or semantics of the received message.

Note 5.— The “-” symbol preceding the User-confirmation primitive indicates that the user has rejected the contents of the received message.

Note 6.— The “|” symbol represents an OR statement.

Note 7.— Primitives shown with dashed lines identify the User-Confirmation service primitives.

Note 8.— The dotted arrows under the column headed “ATN Service Provider” simply identifies transition of data between AIDC-AE peers across the ATN internet.

Note 9.— Dotted extensions to the columns identifies that the sequence shown is either the continuation of a sequence of service invocations or other service invocations are to follow.

Note 10.— The timers shown in the diagrams are technical timers and are defined in 3.2.6 of this document.

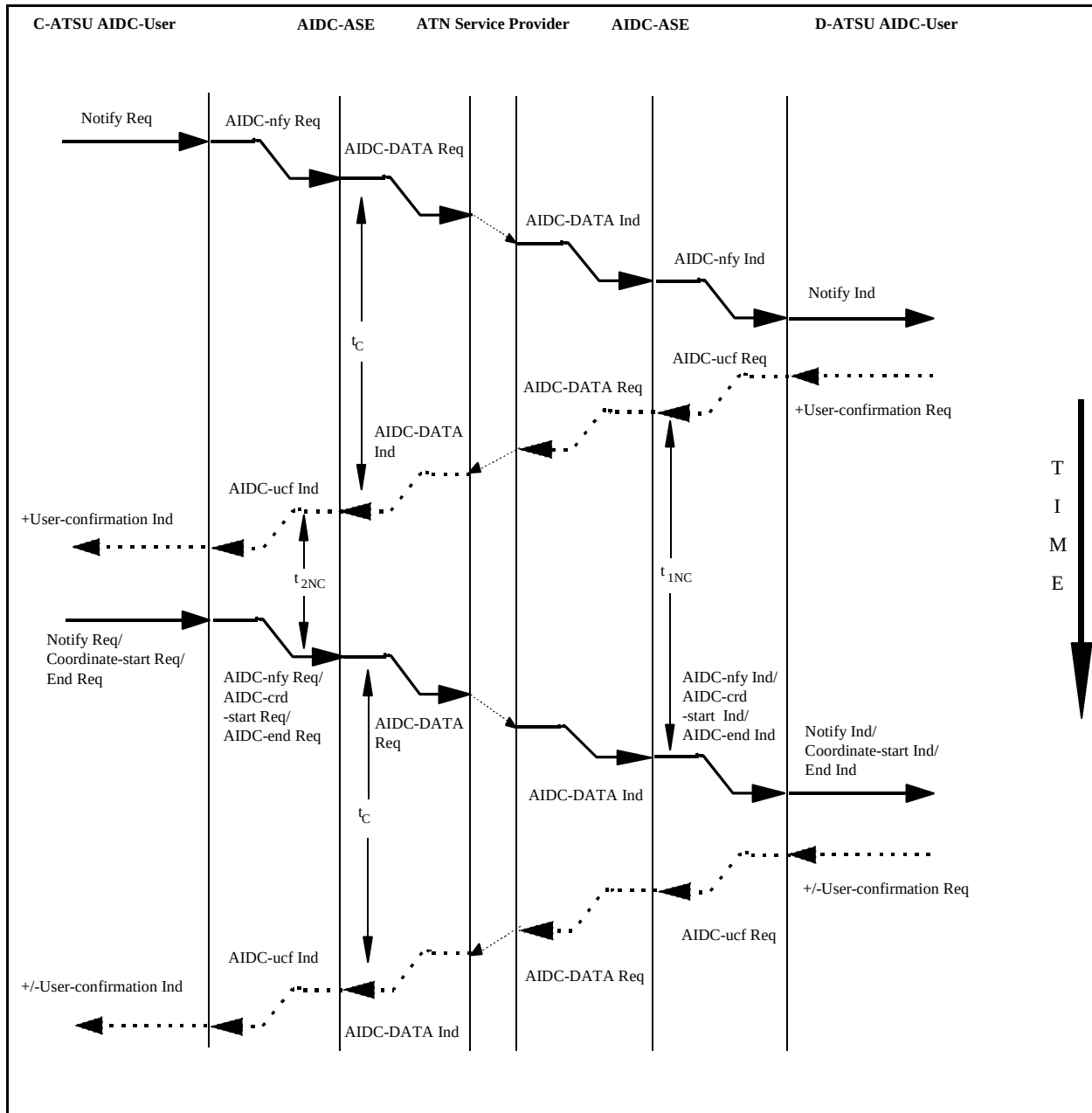


Figure 3.2.10-1: Sequence Diagram showing the entry into the Notifying Regime through the invocation of the Notify service



Figure 3.2.10-2: Sequence Diagram showing the entry and exit of the Coordination Regime through the invocation of the Coordinate-start Service and the Coordinate-end Service respectively

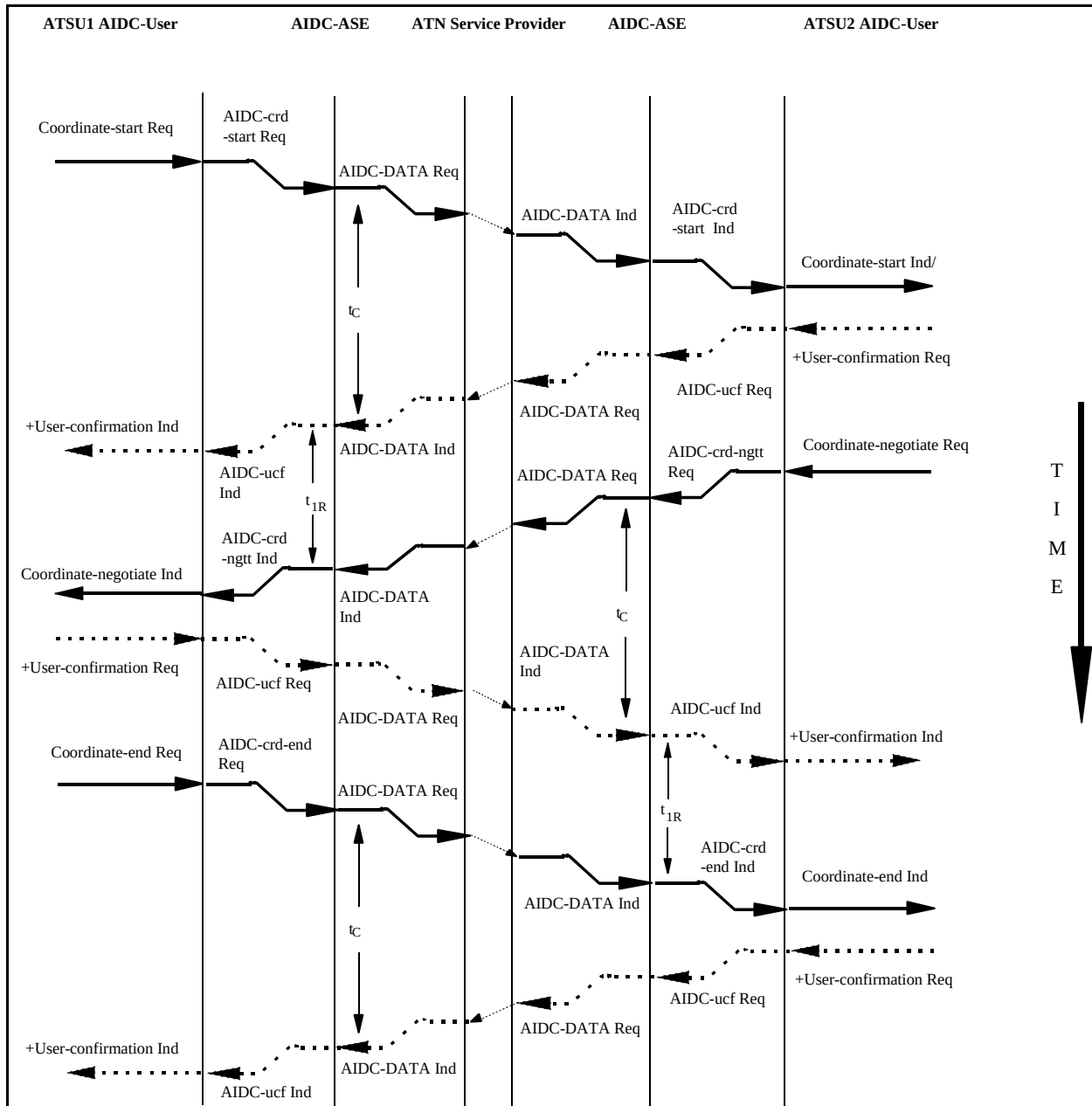


Figure 3.2.10-3: Sequence Diagram showing the invocation of the Coordinate-negotiate Service



Figure 3.2.10-4: Sequence Diagram showing multiple invocations of the Coordinate-negotiate service

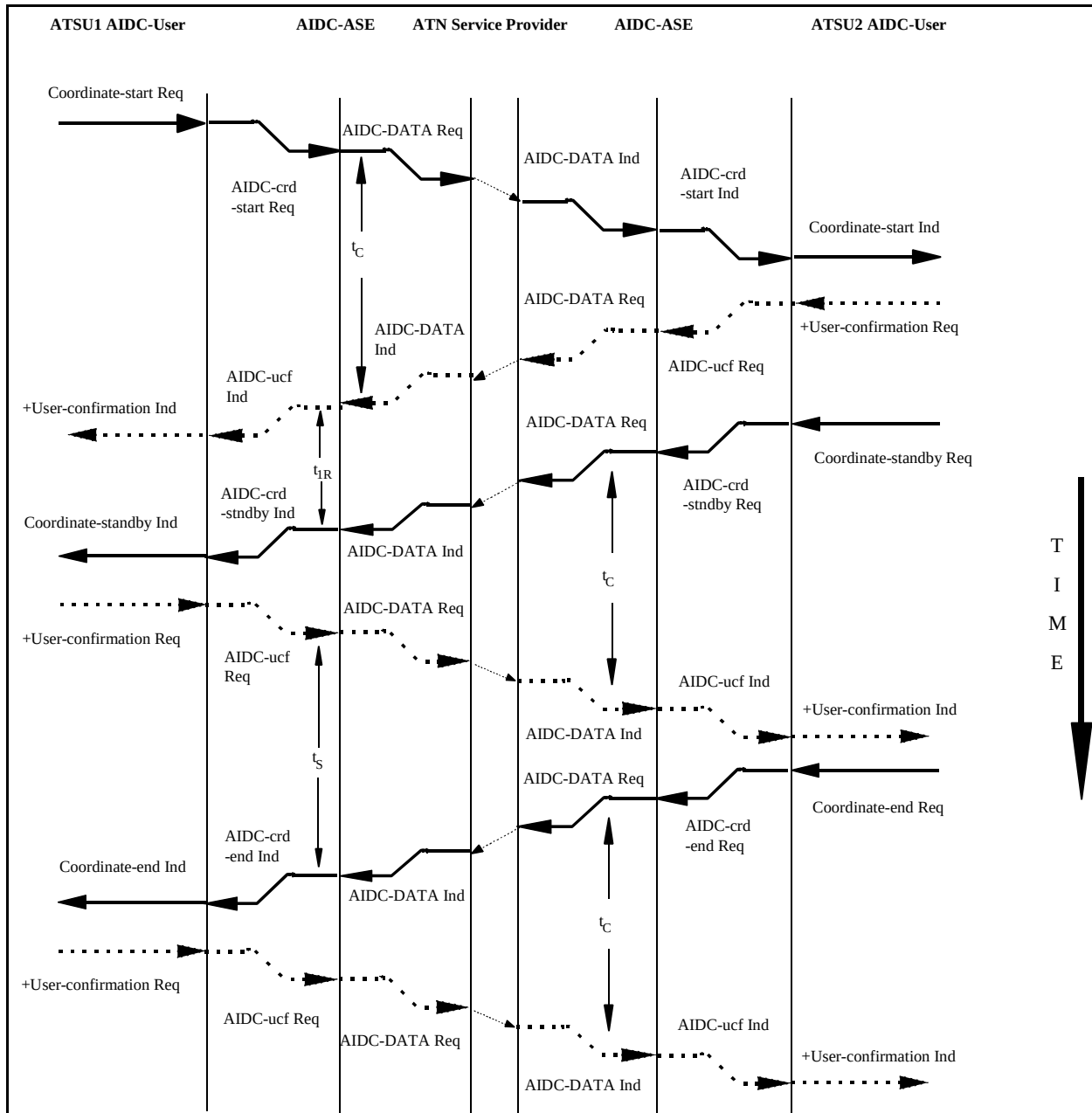


Figure 3.2.10-5: Sequence Diagram showing the invocation of the Coordinate-standby service

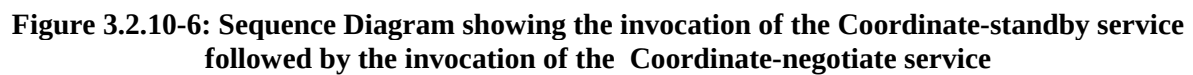


Figure 3.2.10-6: Sequence Diagram showing the invocation of the Coordinate-standby service followed by the invocation of the Coordinate-negotiate service

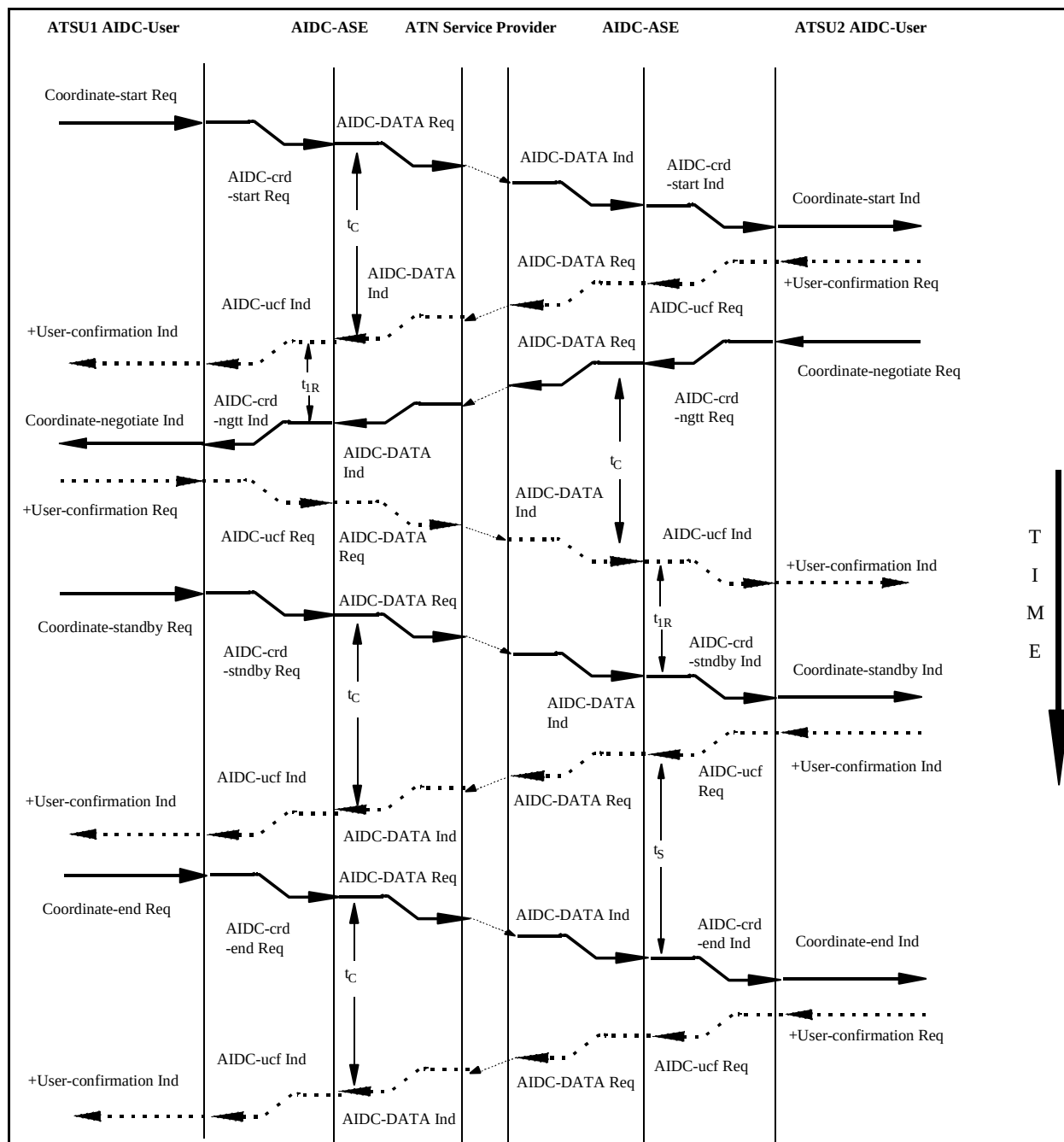


Figure 3.2.10-7: Sequence Diagram showing the invocation of the Coordinate-standby service followed by the Coordinate-negotiate service

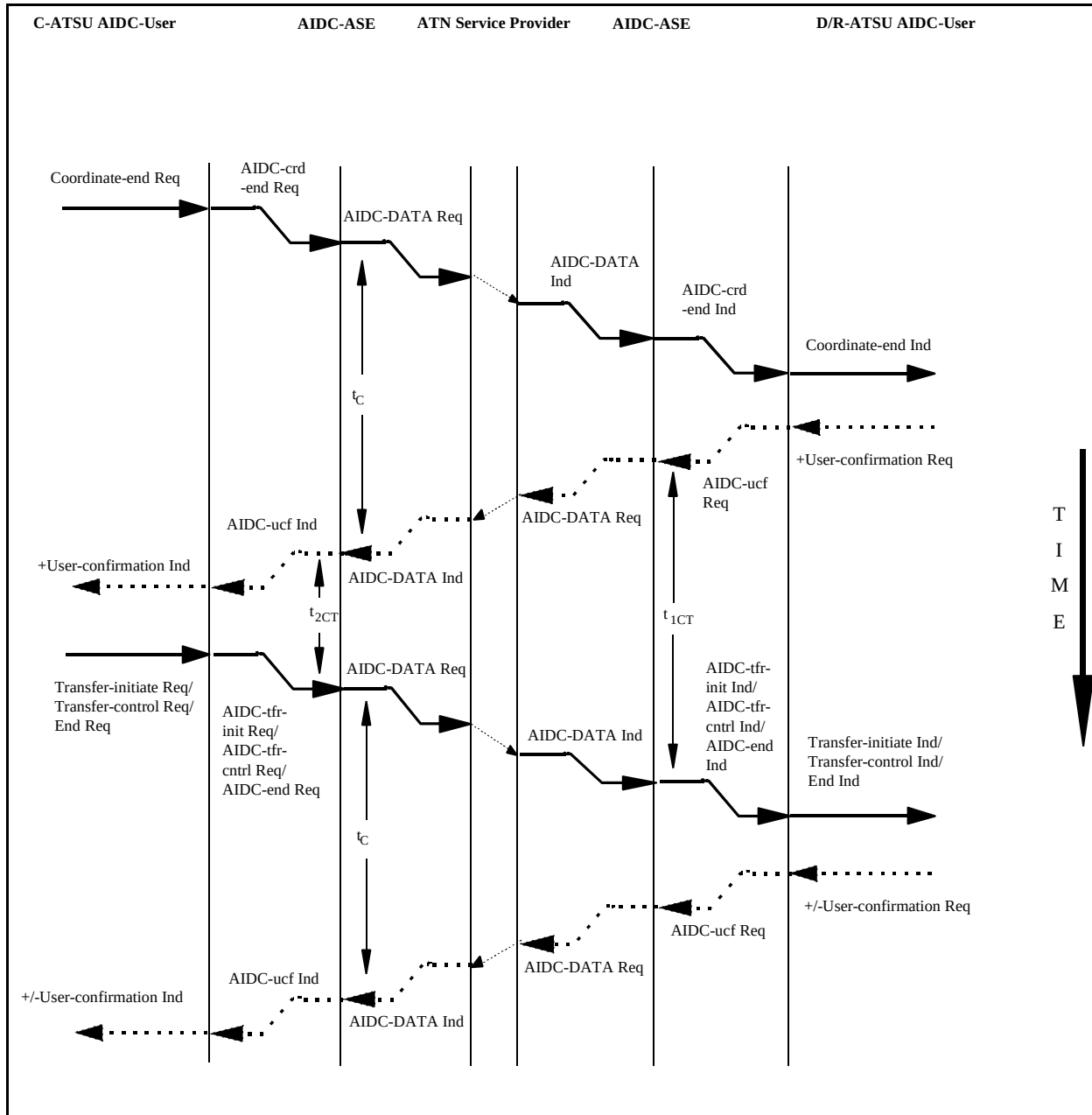


Figure 3.2.10-8: Sequence Diagram showing the start of the Transferring Regime through the invocation of the Transfer-initiate service

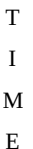


Figure 3.2.10-9: Sequence Diagram showing the start of the Transferring Regime through the invocation of the Transfer-initiate service

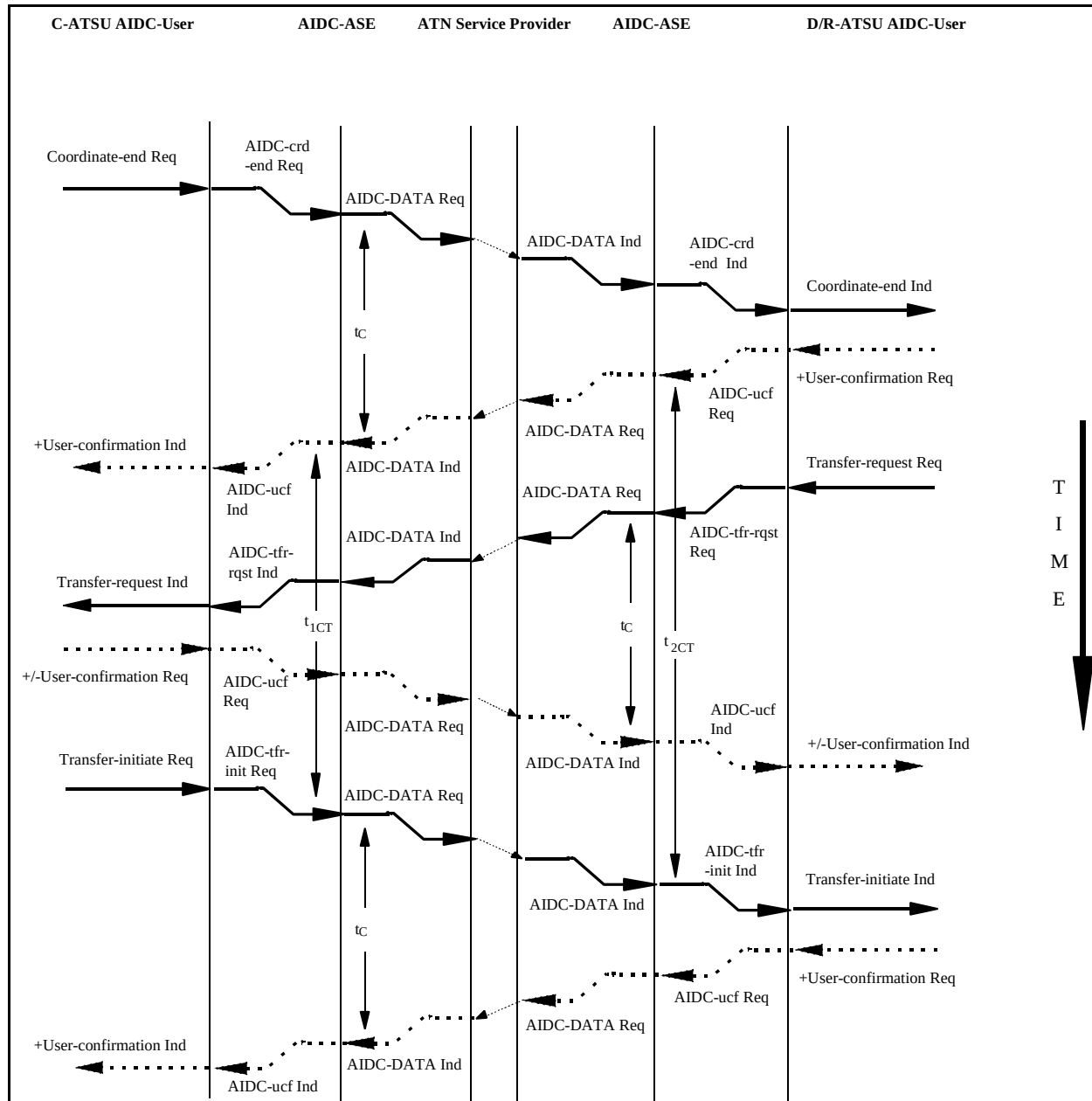


Figure 3.2.10-10: Sequence Diagram showing the invocation of the Transfer-request service

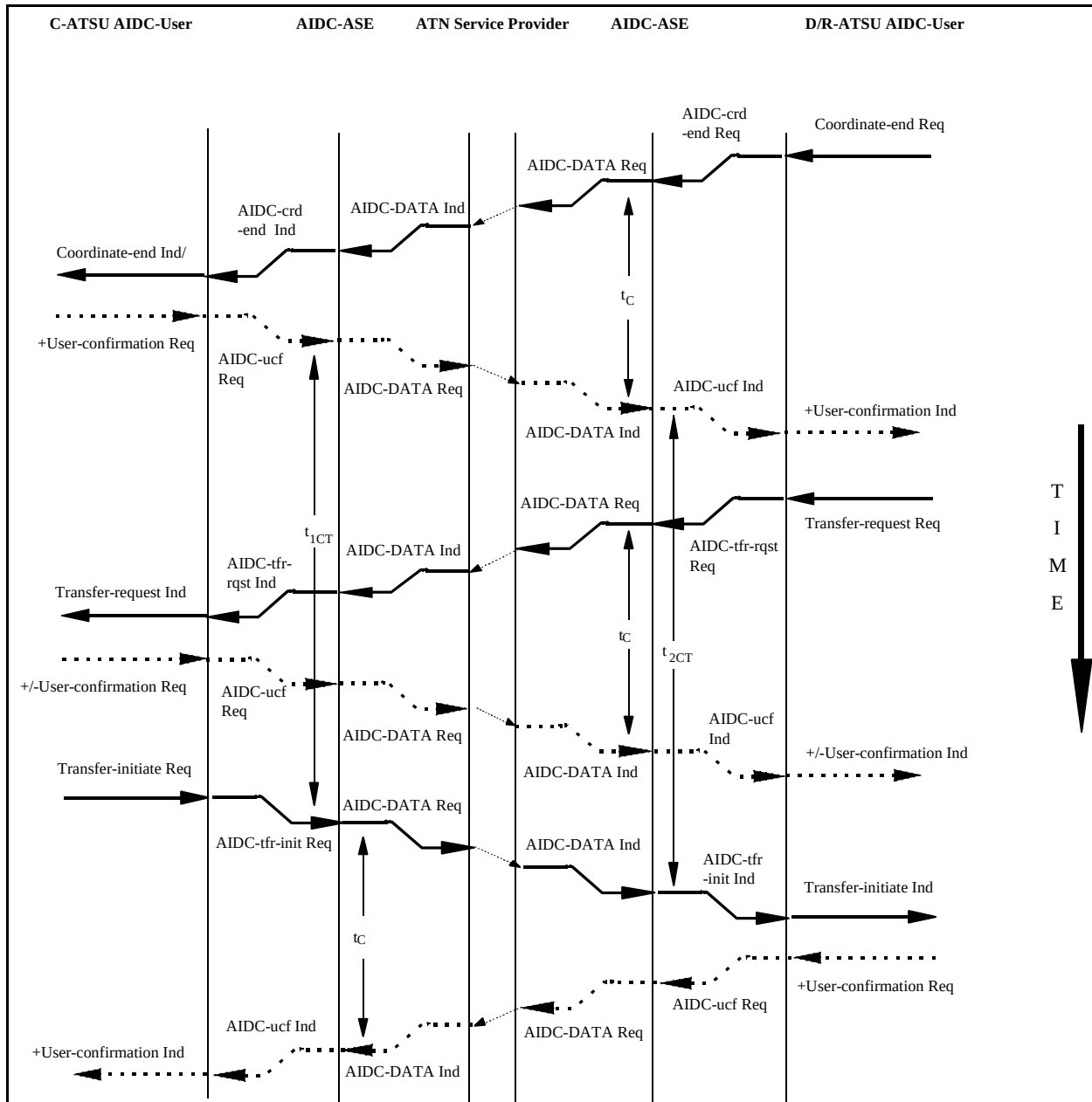


Figure 3.2.10-11: Sequence Diagram showing the invocation of the Transfer-request service

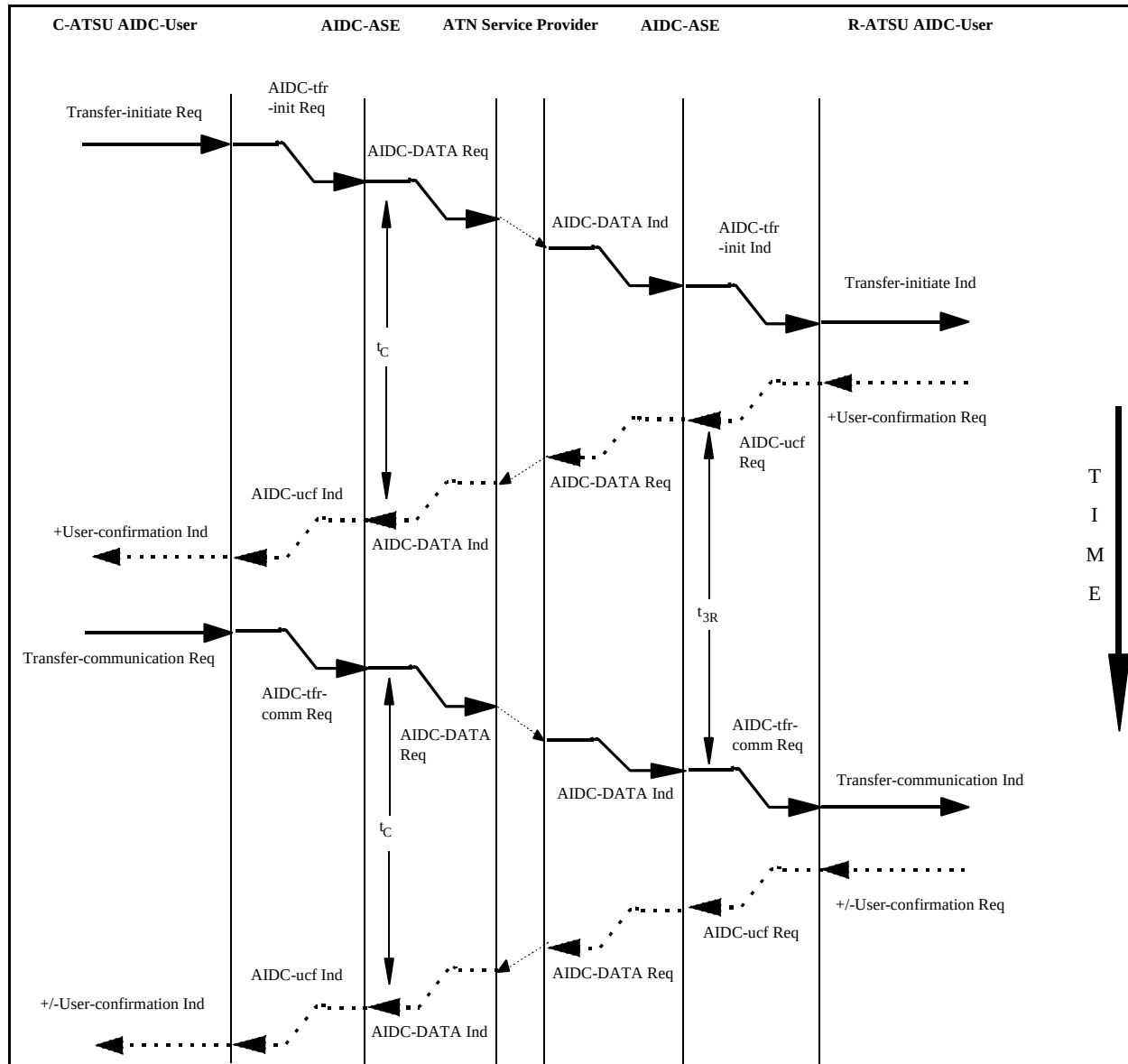


Figure 3.2.10-12: Sequence Diagram showing the invocation of the Transfer-communication service after the Transfer-initiate service

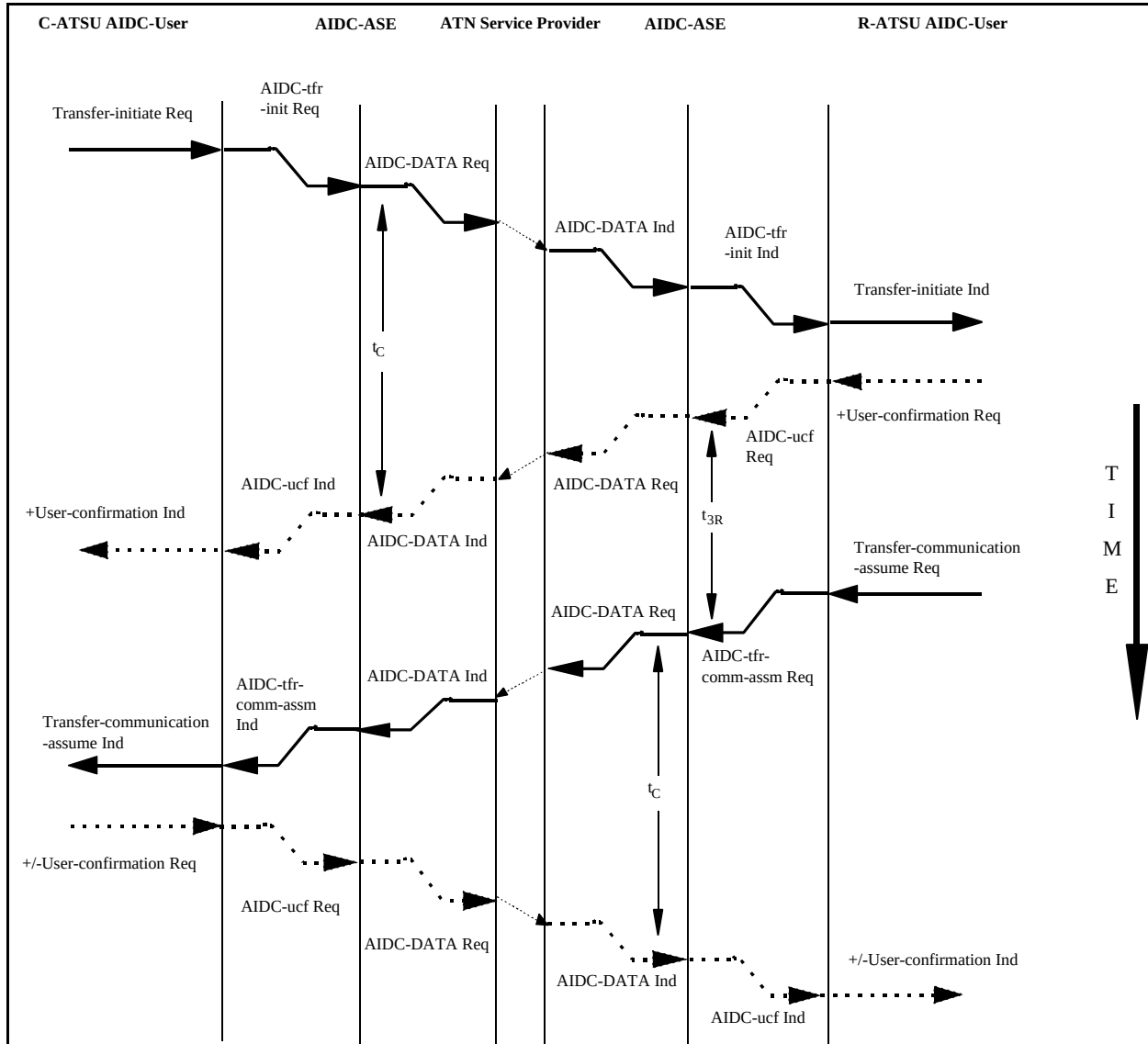


Figure 3.2.10-13: Sequence Diagram showing the invocation of the Transfer-communication-assume service after the Transfer-initiate service

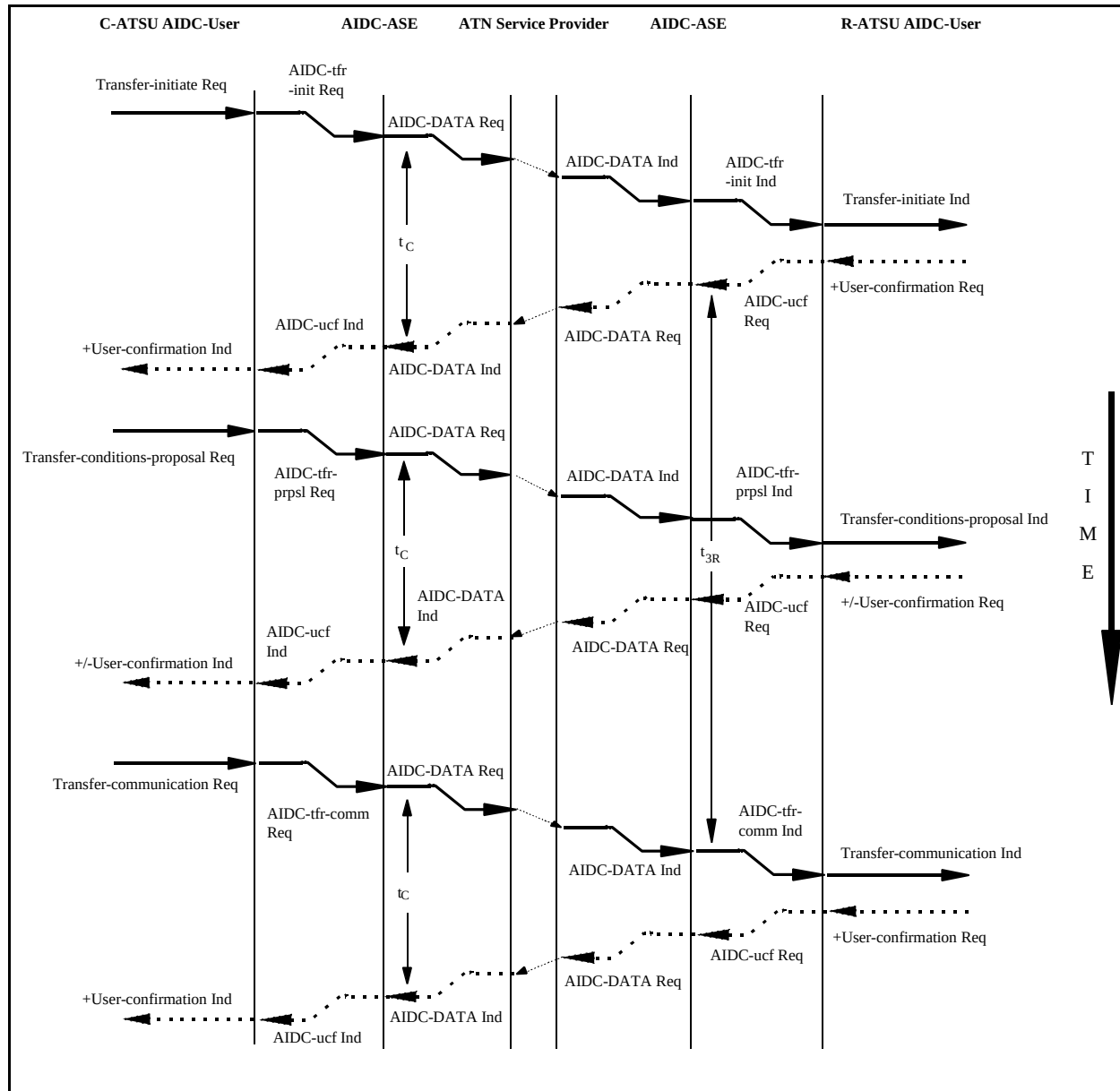


Figure 3.2.10-14: Sequence Diagram showing the invocation of the Transfer-conditions-proposal service, with predicate P1 false, followed by Transfer-communication service

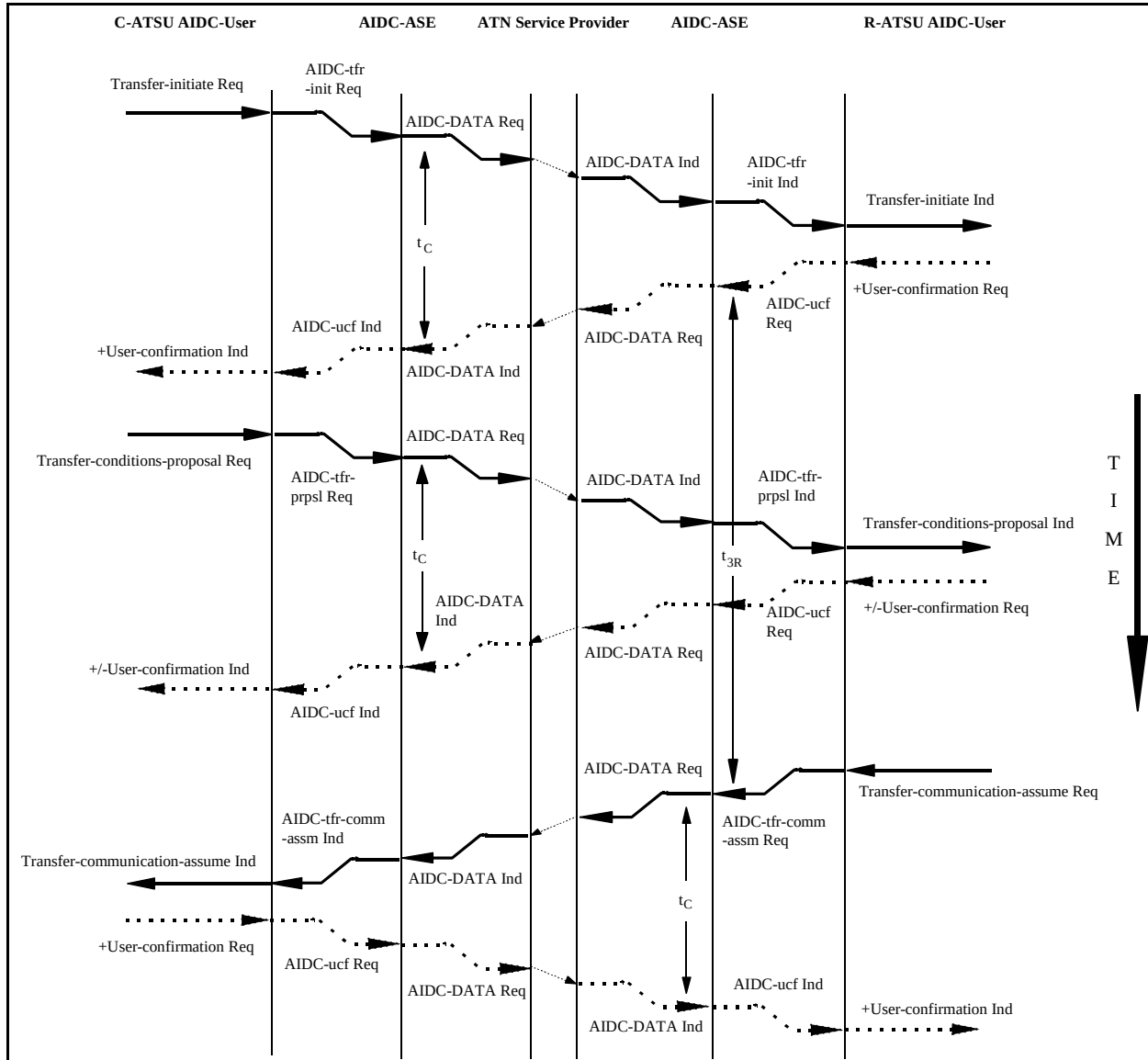


Figure 3.2.10-15: Sequence Diagram showing the invocation of the Transfer-conditions-proposal service, with the predicate P1 false, followed by the invocation of the Transfer-communication-assume service

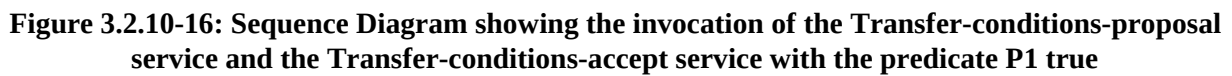


Figure 3.2.10-16: Sequence Diagram showing the invocation of the Transfer-conditions-proposal service and the Transfer-conditions-accept service with the predicate P1 true

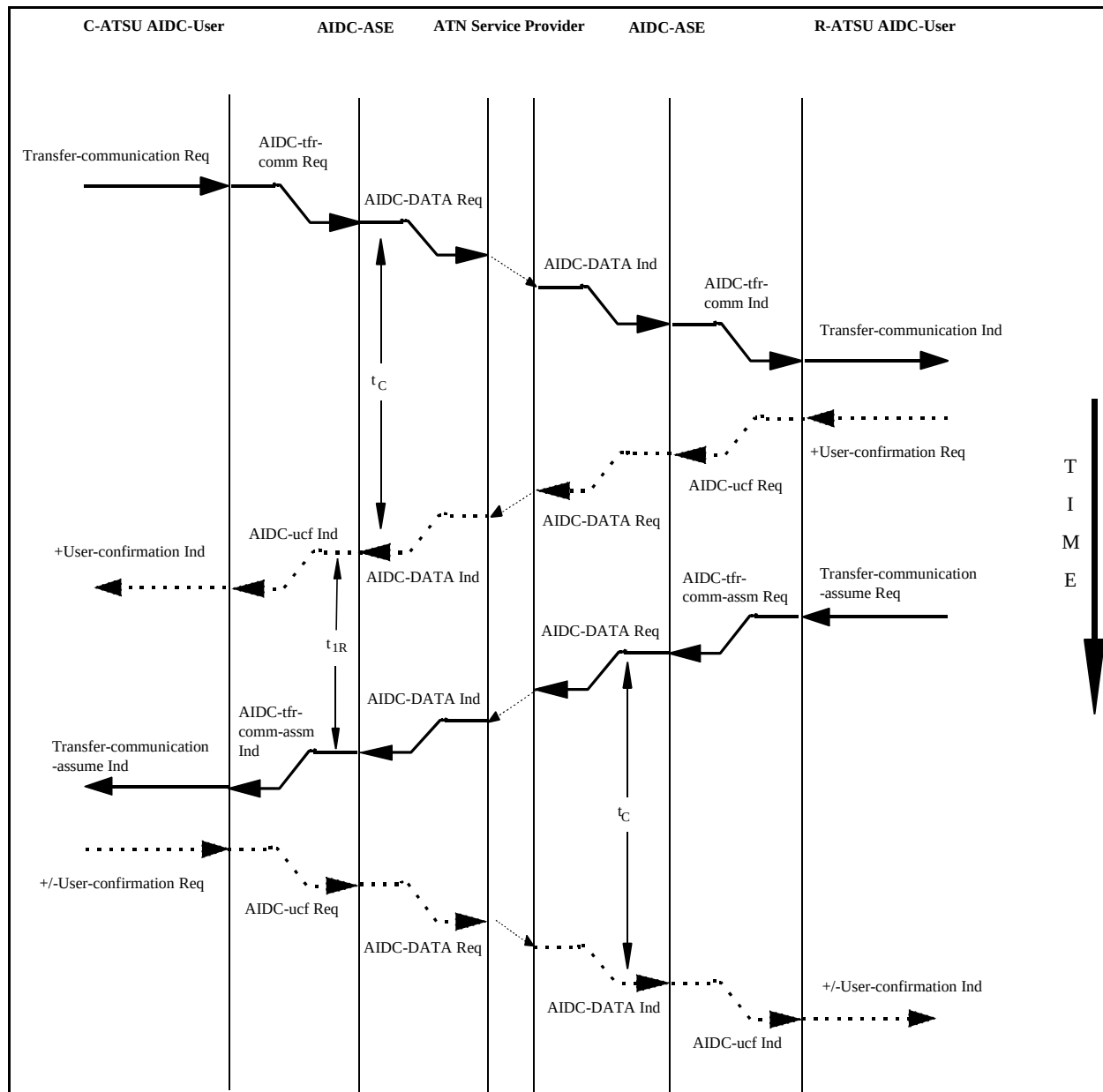


Figure 3.2.10-17: Sequence Diagram showing the invocation of the Transfer-communications service with the invocation of the Transfer-communications-assume service

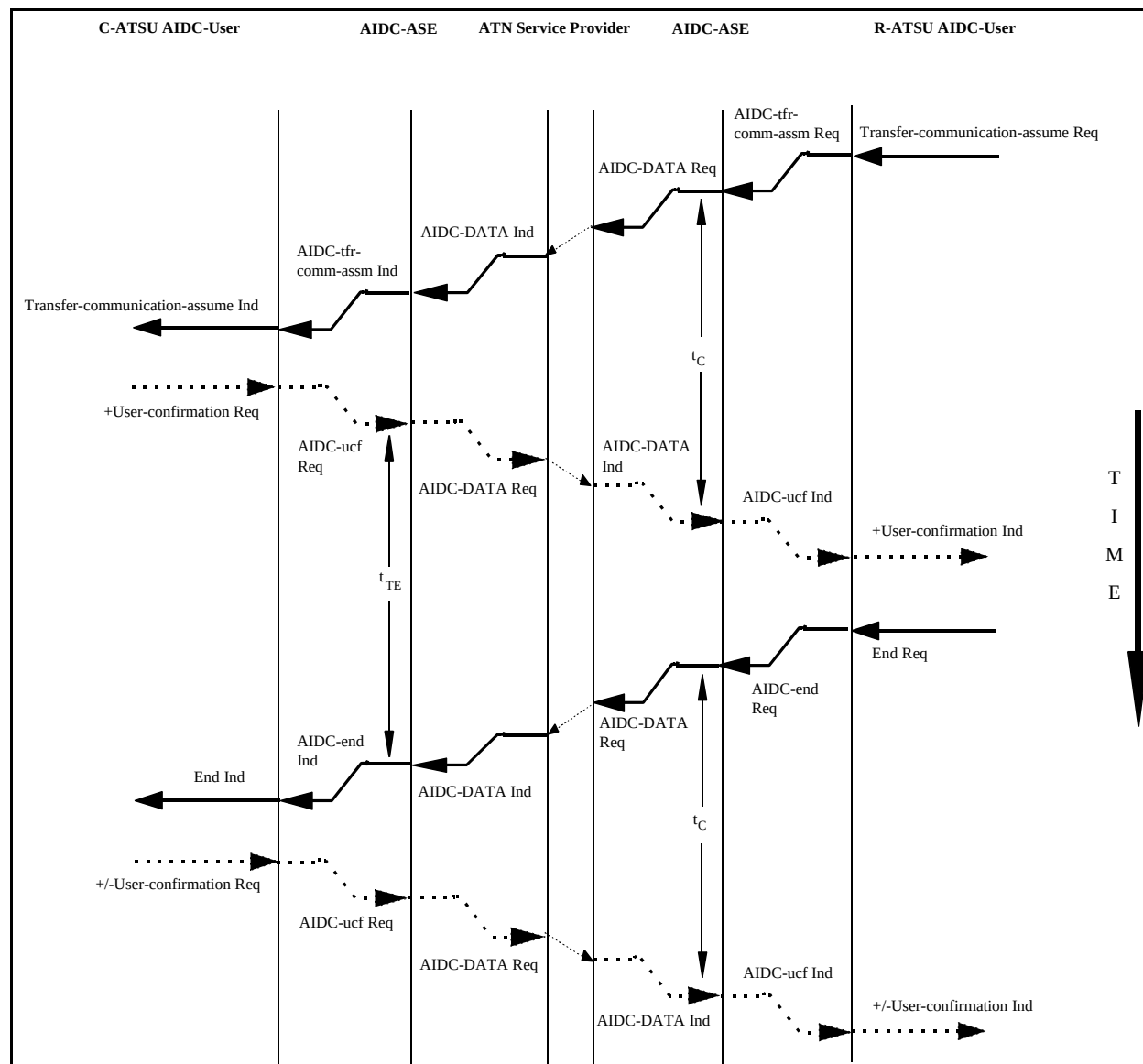


Figure 3.2.10-18: Sequence Diagram showing the end of the AIDC service through the invocation of the End service after the Transfer-communication-assume service

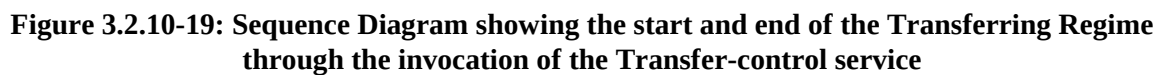


Figure 3.2.10-19: Sequence Diagram showing the start and end of the Transferring Regime through the invocation of the Transfer-control service

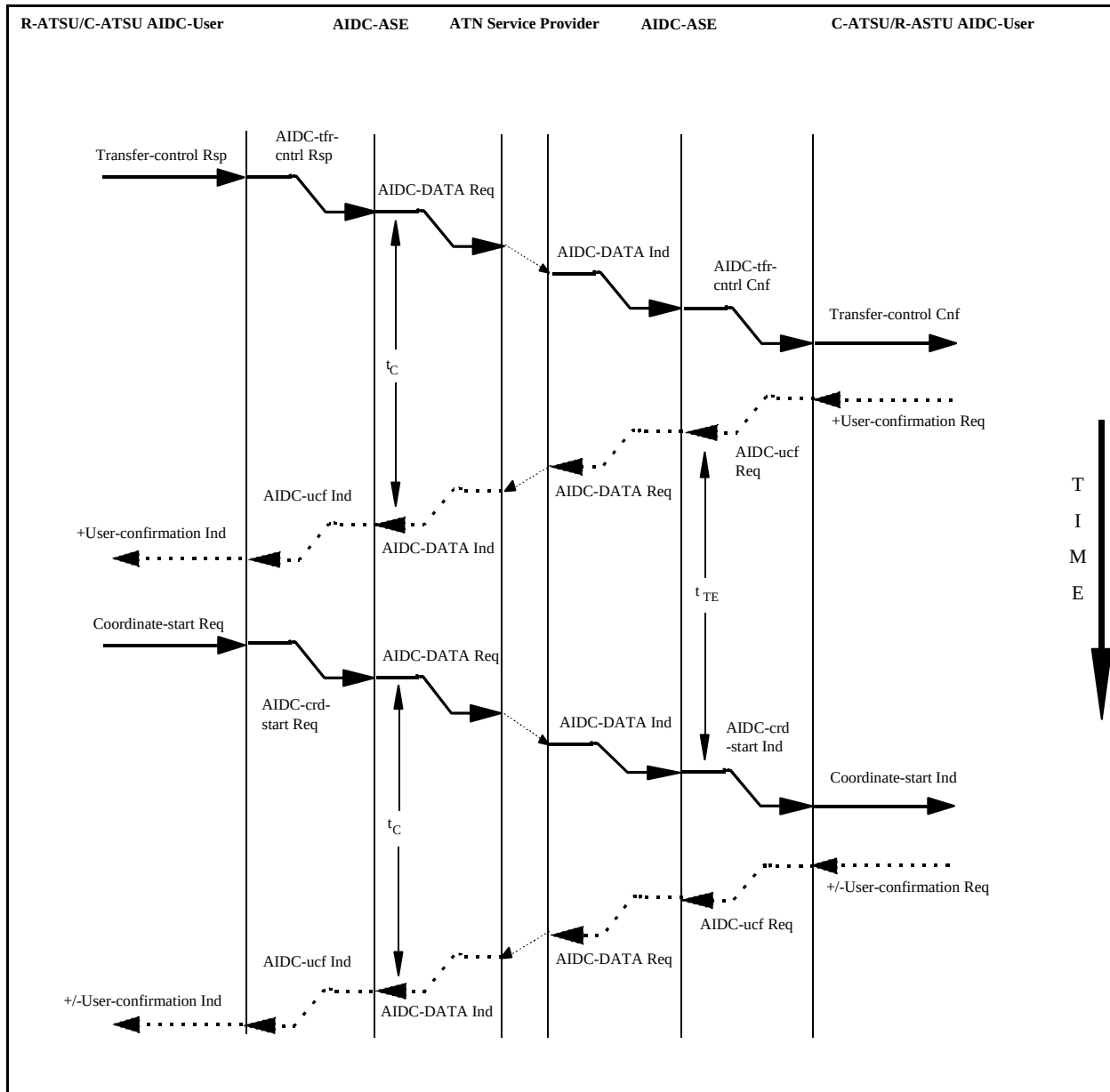


Figure 3.2.10-20: Sequence Diagram showing the re-entry into the Coordinating Regime, after the end of the Transferring Regime, through the invocation of the Coordinate-start service

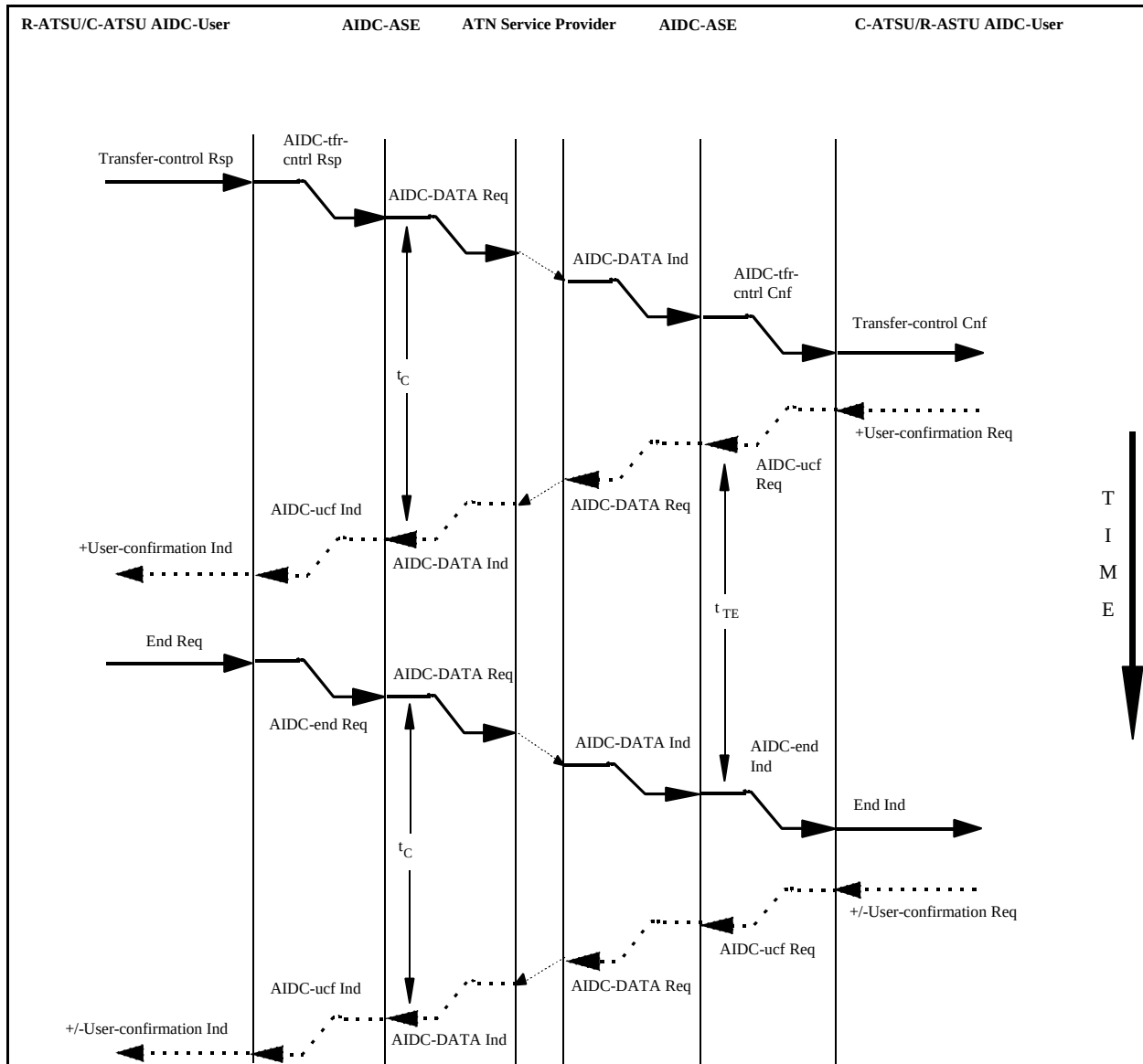


Figure 3.2.10-21: Sequence Diagram showing the end of the AIDC service after the Transfer-control service

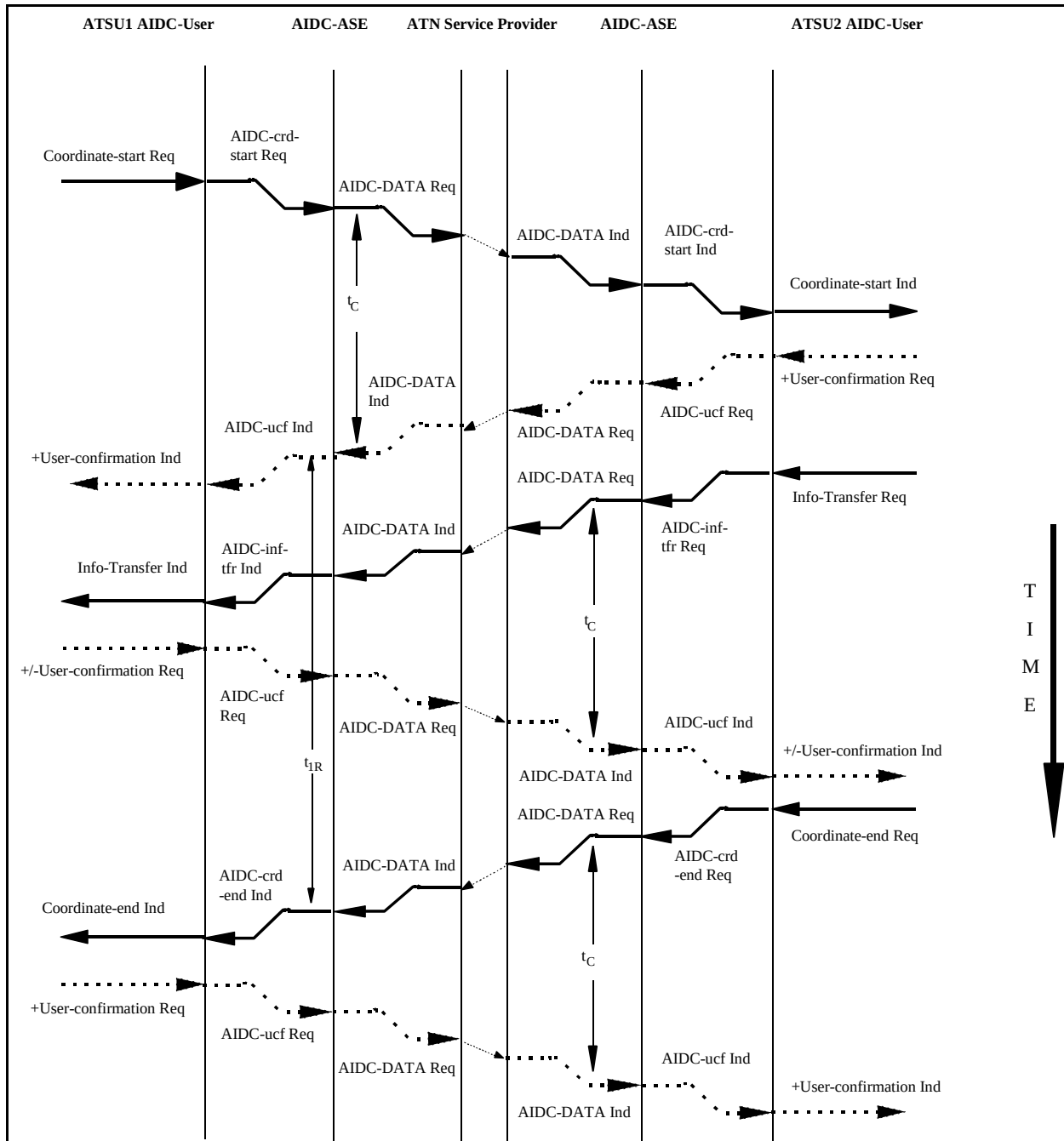


Figure 3.2.10-22: Sequence Diagram showing the invocation of the Info-transfer service in the Coordinating Regime

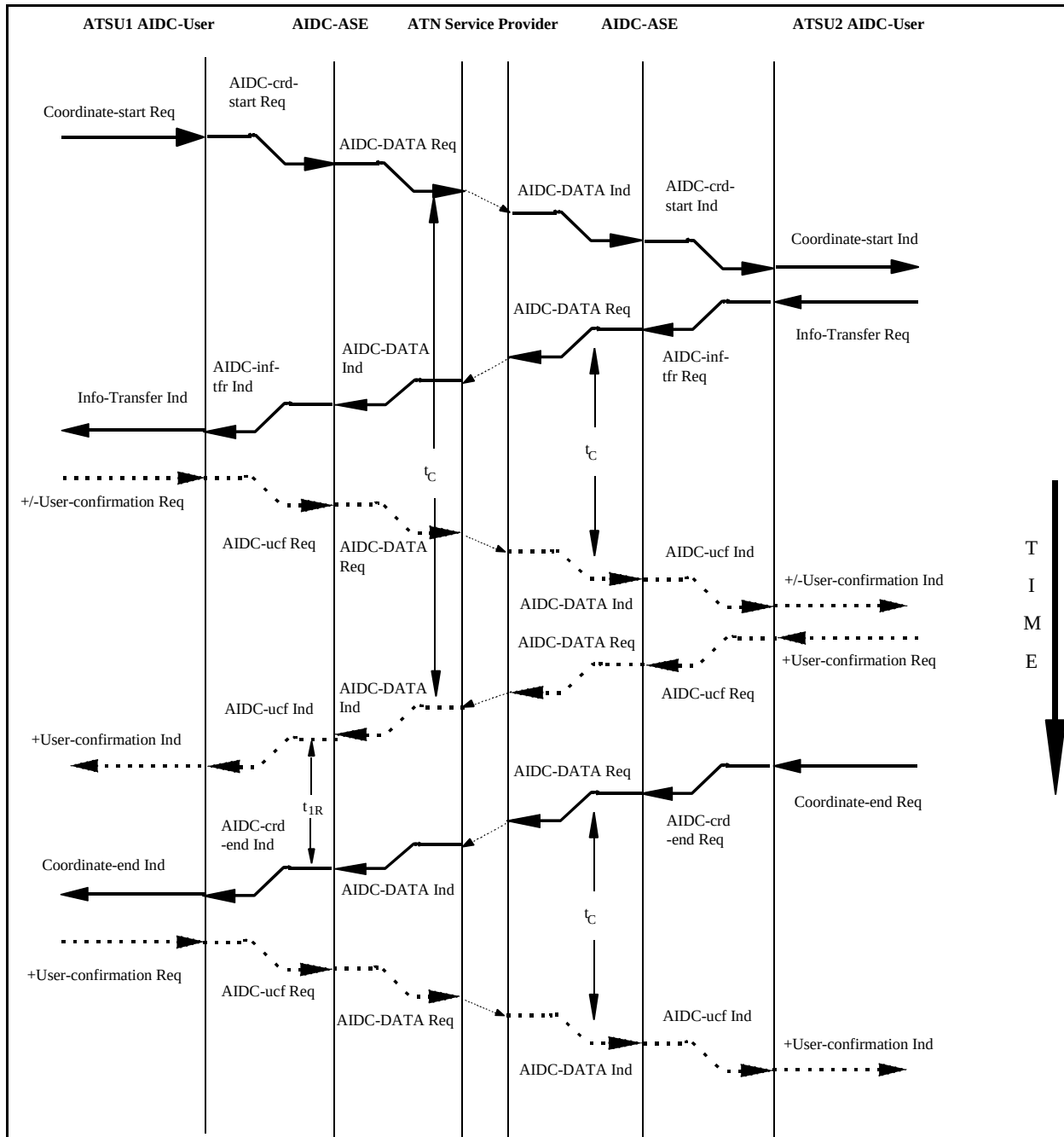


Figure 3.2.10-24: Sequence Diagram showing the invocation of the Info-transfer service before the User Confirmation of the Coordinate-start service

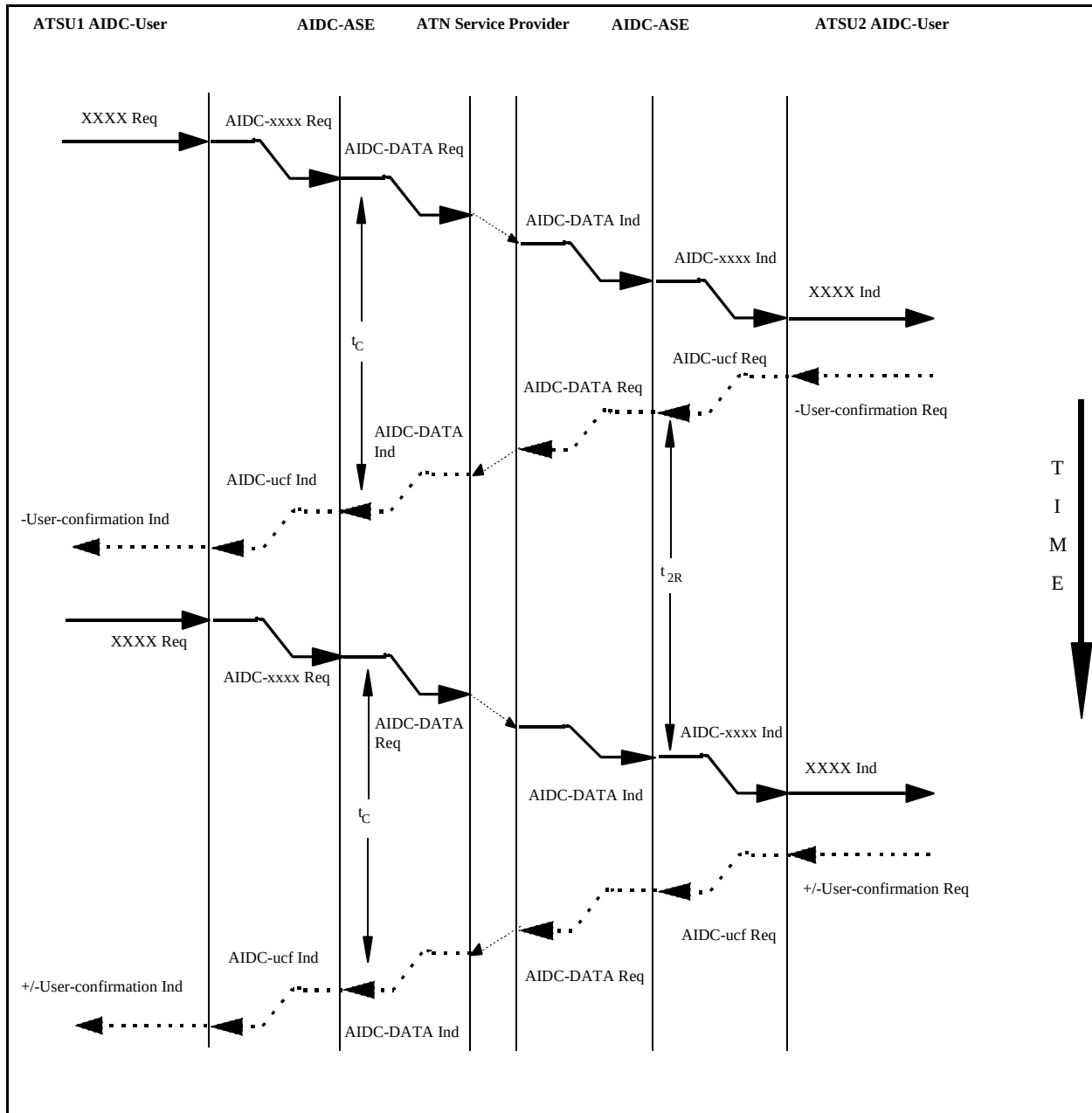


Figure 3.2.10-25: Sequence Diagram showing the sequence for a negative User Confirmation for all services except the Transfer-request, Transfer-conditions-proposal, Transfer-conditions-accept, and Info-Transfer services

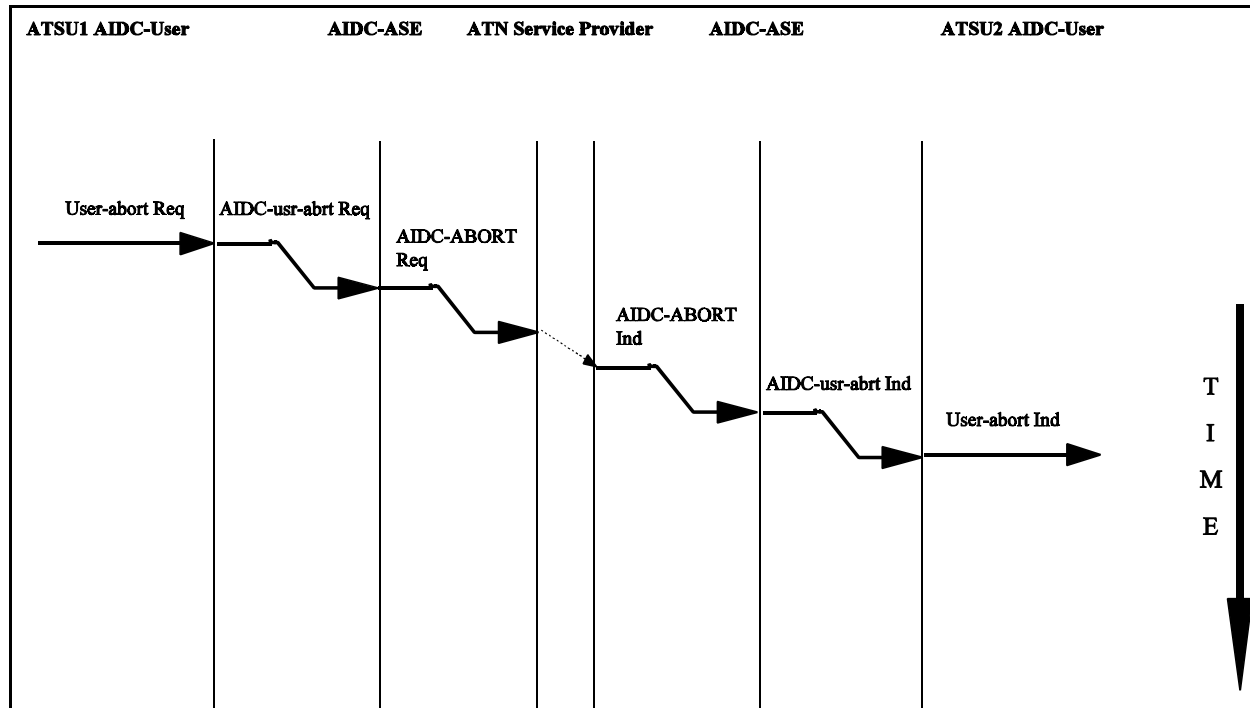


Figure 3.2.10-26: Sequence Diagram for the User-abort service

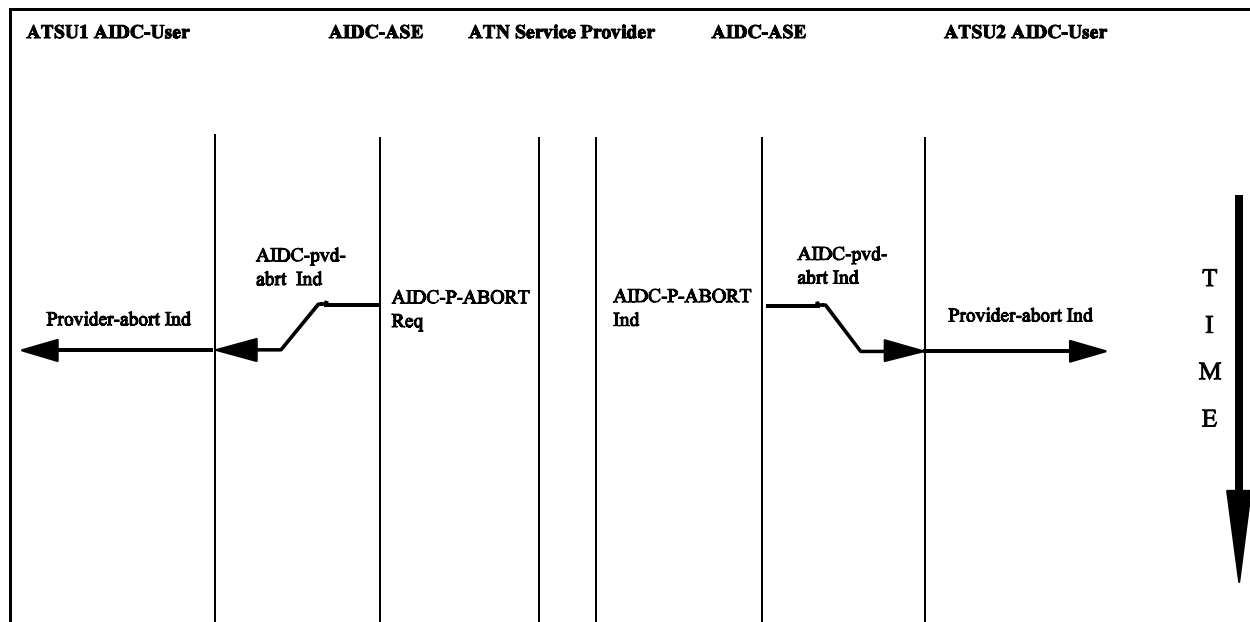


Figure 3.2.10-27: Sequence Diagram showing the invocation of the Provider-abort service invoked by the ATN service provider

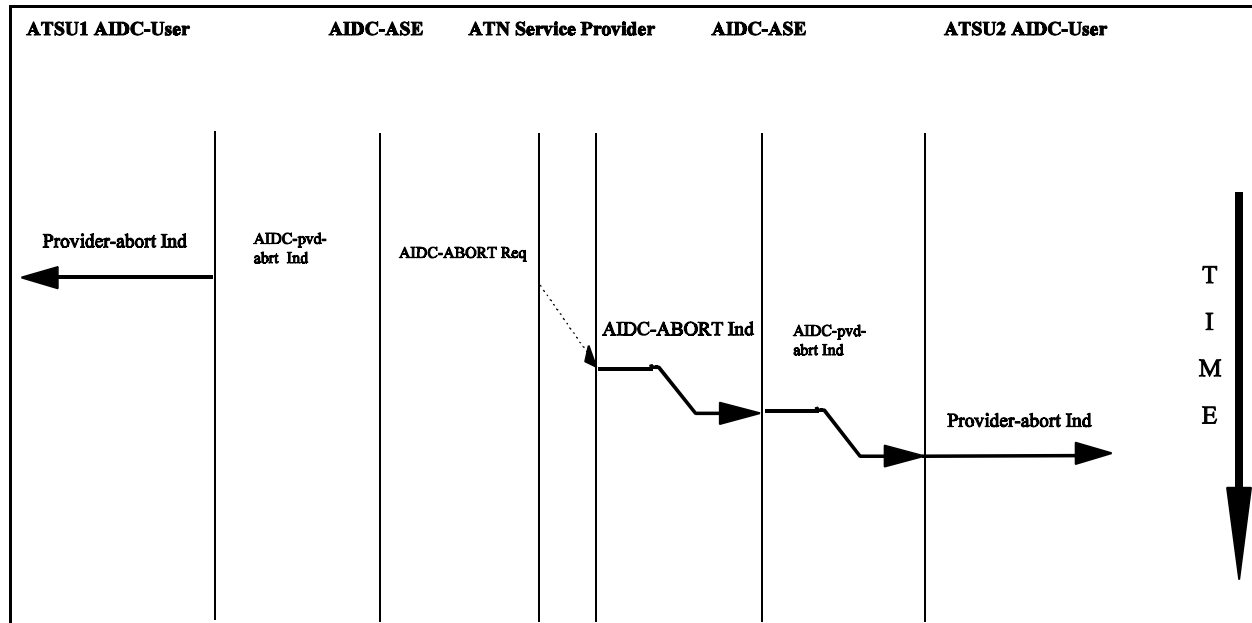


Figure 3.2.10-28: Sequence Diagram showing the service invocation sequence when the AIDC-ASE aborts

Sub-Volume IV

Upper Layer Communications Service

NOTE ON THE SECOND EDITION

The list below shows the parts of this sub-volume that are different from similar parts of the first edition.

Reference	Nature of change
4.3.2.1	Modification
Table 4.3-2	Modification
Table 4.3-4	Modification
4.3.3.4.5.1.1	Modification
4.3.3.6.3.1.1	Modification
4.3.3.6.5.2.2.2 b)	Modification
Table 4.6-9	Modification

4.1 INTRODUCTION

4.1.1 Scope and Objectives

4.1.1.1 The initial version of the ATN Upper Layer (UL) communications service is specified here.

4.1.1.2 The UL specification supports all current ATN applications except the ATS Message Application. This specification is designed to optimise the use of communications bandwidth, and consequently restricts the functionality available from the OSI Session and Presentation layers.

4.1.1.3 The ATN requirements are addressed for Session Layer (Layer 5), Presentation Layer (Layer 6), and a part of the Application Layer (Layer 7) of the OSI reference model. Figure 4.1-1 shows a conceptual view of the scope of the UL communications service. The remaining part of the Application Layer is the province of the individual ATN applications (i.e. the ADS, CM, CPDLC and FIS (ATIS) specifications for air-ground applications, and the ICC (AIDC) specifications for ground-ground applications).

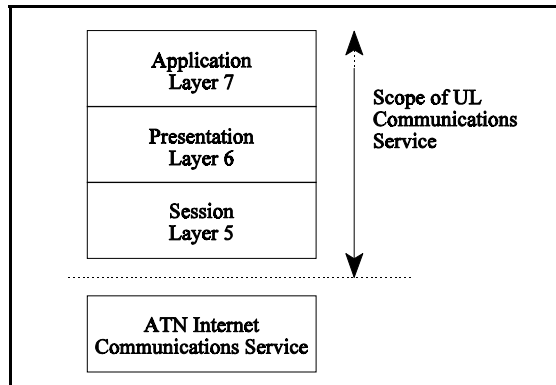


Figure 4.1-1. Conceptual view of the scope of the UL Communications Service

4.1.2 Background

4.1.2.1 The communication aspects of the ATN applications are modelled as Application Entities (AEs) (see 4.1.4.2). Figure 4.1-2 illustrates an example of the application layer structure for the ATN applications.

4.1.2.2 The specification of the UL communications service includes a profile for the protocols in the upper layers, an AE structure and a number of common application services.

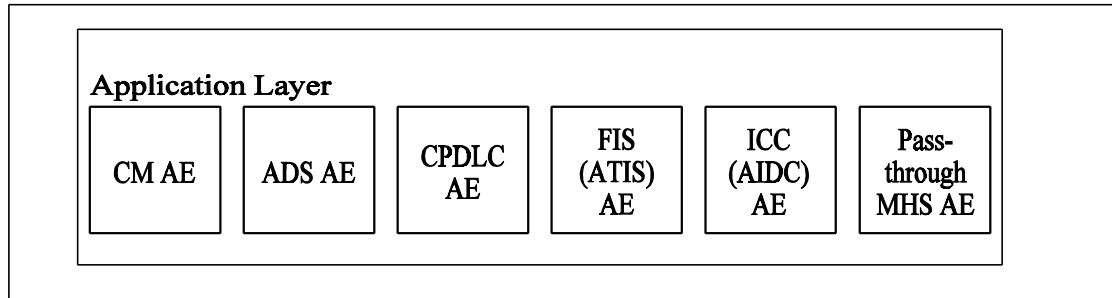


Figure 4.1-2. Conceptual view of Application Layer

4.1.3 Structure of UL Communications Service Specification

4.1.3.1 This specification is structured as follows:

- a) Introduction (4.1) contains the purpose and structure of the UL Communications Service Specification, and a background to the functionality defined herein.
- b) Dialogue Service Description (4.2) describes the abstract service which is defined for application specifications to refer to in order to provide a common communications service.
- c) Application Entity (AE) Description (4.3) describes the Application Entity and specifies the Control Function (CF) which co-ordinates the operation of the various Application Service Elements (ASEs). It also describes the names which are assigned to various upper layer entities.
- d) Session Layer Requirements (4.4) describes the requirements for the OSI Session Layer, in the form of a Profile Requirements List (PRL).
- e) Presentation Layer Requirements (4.5) describes the requirements for the OSI Presentation Layer, in the form of a PRL.
- f) ACSE Specification (4.6) describes the requirements for the Association Control Service Element (ACSE).

4.1.4 Upper Layer Functionality

4.1.4.1 Upper Layer Profile Overview

4.1.4.1.1 A profile is specified for the connection-oriented protocols of Session layer, Presentation layer and the Association Control Service Element (ACSE).

4.1.4.1.2 The Session portion of the specified profile is based on the efficiency enhancements to the Session protocol which are standardised in ISO/IEC 8327-1: 1996 / Amd. 1: 1997

4.1.4.1.3 The Presentation portion of the specified profile is based on the efficiency enhancements to the Presentation protocol which are standardised in ISO/IEC 8823-1: 1994 / Amd. 1: 1997

4.1.4.1.4 As a consequence of using the Session and Presentation protocol efficiency enhancements, the protocol control information transferred by these protocols amounts to two octets in each direction during the connection establishment phase, and zero octets at all other times.

4.1.4.1.5 The ACSE portion of the specified profile is based on ISO/IEC 8650-1, including the extensibility notation as specified as Amendment 1 to that standard.

4.1.4.2 Application Entity (AE) Structure

4.1.4.2.1 The specified AE structure is based on the application layer structure defined in ISO/IEC 9545, where the concepts of Application Service Element (ASE), Application Service Object (ASO) and Control Function (CF) are defined.

4.1.4.2.2 Figure 4.1-3 shows the generic structure of an AE with arrows representing the abstract service boundaries of the various elements. The “upper” service boundary is the abstract service provided by an ASE to its user(s). The “lower” service boundary is the abstract service which is provided to the ASE by the CF.

4.1.4.2.3 The ASE is an element engineered to perform a required task. ISO/IEC 9545 describes how two or more ASEs may be combined, together with a CF to co-ordinate their operation to form an ASO. In turn, an ASO may be combined with other ASOs or ASEs with a CF to form larger ASOs. The AE is the outermost ASO.

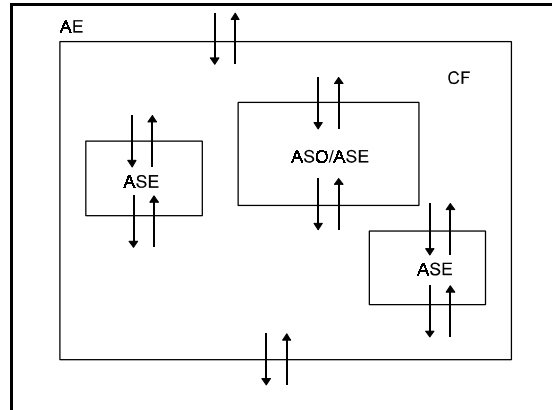


Figure 4.1-3. Generic Application Entity structure

4.1.4.3 Application Services

4.1.4.3.1 For each of the current ATN applications a specific ASE exists, and is defined in the relevant ATN Application specification. The generic name “ATN-App ASE” is used for these specific ASEs.

4.1.4.3.2 Various abstract services are specified. The services are provided at abstract service boundaries. The abstract service provided by the AE to the Application-user (i.e. the service provided at the upper boundary of the AE) is specified in 4.3. In the AE structure specified here, this service is a pass-through to the ATN-App ASE.

4.1.4.3.3 Figure 4.1-4 shows the AE structure which is used to model the ATN applications. This is described in detail in 4.3.

4.1.4.3.4 The Dialogue Service (DS) as defined in 4.2 is the abstract service which the ATN-App ASEs use to interact with the UL communications service. That is, the DS is the combination of specific internal primitives made available by the CF at the lower boundary of the ATN ASE/ASO - it is the application’s “world view”. In order to provide this service, the CF uses the services of ACSE.

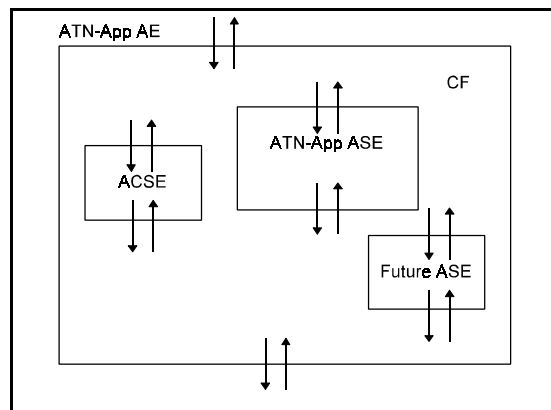


Figure 4.1-4. ATN-specific AE structure

4.2 DIALOGUE SERVICE DESCRIPTION

4.2.1 Scope of Dialogue Service

4.2.1.1 Implementations of the ATN-App ASE, together with the UL elements which provide the Dialogue Service (DS), shall exhibit the behaviour defined in this abstract service definition.

Note 1.— The Dialogue Service is the abstract service which is used by an ATN-App ASE at its lower service boundary. There is no requirement to implement the DS in any product. ATN end systems will in general be designed in such a way that it is impossible to detect (from external access) whether or not an interface corresponding to the DS has been built.

Note 2.— The DS is described from the viewpoint of the ATN-App ASE, using abstract service definition conventions. The abstract service definition is a descriptive technique used to specify the behaviour exhibited by part of the ATN application layer. Specifications of application service elements (ASEs), such as the specifications of ADS, CPDLC, CM and ATIS, may include common functionality by reference to the DS. The DS allows ATN-App ASEs to be specified without the need to consider some of the complexities of some aspects of the underlying communications.

Note 3.— The DS supports a communication relationship between two peers for a duration which exists until the peers agree to terminate the relationship or the relationship is aborted.

*Note 4.— The DS defines a service which may be used to support an ATN-App ASE at its “lower” service boundary. Such an ASE is denoted a **DS-User**. The DS-User can be specified to use the DS in a variety of ways that can be defined in terms of reliability characteristics. A number of user-visible service levels can thus be offered, including for example the following:*

- a) An unconfirmed service, which allows individual messages to be transmitted after a dialogue has been set up.*
- b) A confirmed service, which provides end-to-end confirmation that a message sent by one DS-User was received and acknowledged by the peer DS-User.*

Note 5.— An implementation of the DS provider will typically be responsible for detection of errors such as:

- a) Invalid primitive (primitive unknown or error in parameter(s))*
- b) Invalid sequence (primitive issued at inappropriate time)*
- c) Insufficient resources on submission*
- d) Invalid or unreachable recipient on submission*
- e) Data field too large on receive (local implementation constraint exceeded)*
- f) Invalid or unreachable recipient on receive.*

Note 6.— An implementation of an ATN application which makes use of the DS has to be designed with error handling procedures for local error conditions.

4.2.2 Service Primitives

4.2.2.1 Implementations which claim to support the DS functionality shall exhibit the behaviour defined by the service primitives in Table 4.2-1.

Table 4.2-1. Summary of Dialogue Service primitives

Service	Description
D-START	This is a confirmed service used to establish the binding between the communicating DS-Users.
D-DATA	This unconfirmed service is used by a DS-User to send a message from that DS-User to the peer DS-User.
D-END	This is a confirmed service used to provide the orderly unbinding between the communicating DS-Users, such that any data in transit between the partners is delivered before the unbinding takes effect.
D-ABORT	This unconfirmed service can be invoked to abort the relationship between the communicating DS-Users. Any data in transit between them may be lost.
D-P-ABORT	This unconfirmed service is used to indicate to the DS-User that the dialogue service provider has aborted the relationship with the peer DS-User. Any data in transit between the communicating DS-Users may be lost.

Note.— Table 4.2-2 lists the parameters used when invoking the services.

Table 4.2-2. Parameters of the Dialogue Service primitives

Service	Parameters
D-START	Called Peer ID Calling Peer ID DS-User Version Number Security Requirements Quality-of-Service Result Reject Source User Data
D-DATA	User Data
D-END	Result User Data
D-ABORT	Originator User Data
D-P-ABORT	(no parameters)

4.2.3 Service Definition

4.2.3.1 Sequence of Primitives

4.2.3.1.1 Implementations which claim to support the DS functionality shall exhibit behaviour allowing two communicating DS-Users to:

- a) establish a dialogue;
- b) exchange user data;
- c) terminate a dialogue in an orderly or abnormal fashion; and
- d) be informed of DS abnormal dialogue termination due to the underlying communications failure;

consistent with the appropriate use of the corresponding service primitives.

Note 1.— Either DS-User may send data at any time after the initial D-START exchange, by using the D-DATA service. Under normal circumstances, a dialogue is released by a DS-User invoking the D-END service. A dialogue is abnormally released with the D-ABORT service. If the underlying service provider abnormally releases the dialogue, the DS-Users which are aware of the dialogue will be notified with the D-P-ABORT service.

Note 2.— For the purposes of this service definition, it is only valid for the DS-User to issue and be prepared to receive primitives for one Dialogue according to the permitted sequences of DS primitives shown in Table 4.2-3, where intersections marked “Y” show possible primitives which may occur after the primitive in the column heading.

Table 4.2-3. Sequence of DS primitives for one Dialogue at one DS-User

<i>The DS primitive X -> may be followed by the DS primitive Y</i>	1	2	3	4	5	6	7	8	9	10	11	12	13
1 D-START req													
2 D-START cnf (accepted)	Y												
3 D-START ind								Y		Y	Y	Y	Y
4 D-START rsp (accepted)			Y										
5 D-DATA req		Y		Y	Y	Y			Y				
6 D-DATA ind		Y		Y	Y	Y	Y						
7 D-END req		Y		Y	Y	Y							
8 D-END cnf (accepted)							Y						
9 D-END ind		Y		Y	Y	Y							
10 D-END rsp (accepted)									Y				
11 D-ABORT req	Y	Y	Y	Y	Y	Y	Y		Y				

The DS primitive X -> may be followed by the DS primitive Y	1	2	3	4	5	6	7	8	9	10	11	12	13
12 D-ABORT ind	Y	Y	Y	Y	Y	Y	Y		Y				
13 D-P-ABORT ind	Y	Y	Y	Y	Y	Y	Y		Y				

Note 3.— For compactness, each DS primitive is given a number in the column headings in Table 4.2-3; the numbers have the meanings assigned in the row headings. For simplicity, where D-START and D-END response and confirmation primitives are used, Table 4.2-3 only shows the case where the D-START or D-END request is accepted by the peer. If a D-START request is rejected, then a DS-User may not issue or receive any other primitives apart from D-START request or indication. If a D-END request is rejected, then a DS-User may continue to issue and receive primitives as if the dialogue had just been established. A D-START request results in a new instance of communication with the peer DS-User, so could occur at any time. Table 4.2-3 only applies to a single instance of communication.

4.2.3.2 The D-START service

4.2.3.2.1 The behaviour defined by the D-START service primitive shall be provided to enable the setting up of a dialogue between two DS-Users.

Note 1.— D-START is a confirmed service which is invoked by a DS-User (the dialogue-initiator) to start a dialogue with a peer DS-User. D-START request, indication, response and confirmation primitives are defined, as illustrated in Figure 4.2-1.

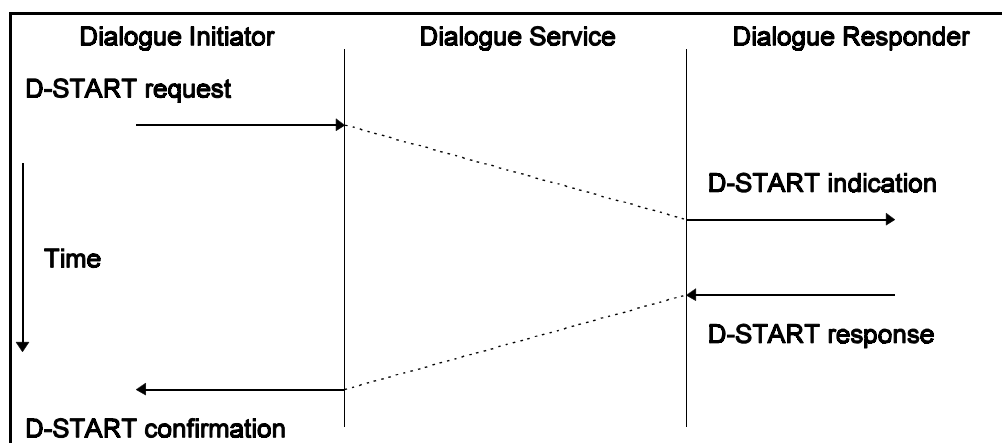


Figure 4.2-1. D-START sequence diagram

Note 2.— The initiating DS-User issues a D-START request primitive. It is not then valid to issue any other primitives (except D-ABORT) until a D-START confirmation is received. When the responding DS-User receives the D-START indication primitive, it must decide whether or not to accept this instantiation of the dialogue service. It may issue only a D-START response or a D-ABORT request primitive. The D-START response and confirmation primitives contain a Result parameter which defines whether the responding DS-User accepts or rejects the request. If the responding DS-User accepts the request, then the dialogue is established. If it rejects the request, then no dialogue exists. The parameters of the D-START primitives are specified in Table 4.2-4.

Table 4.2-4. D-START parameters

Parameter Name	Req	Ind	Rsp	Cnf
Called Peer ID	M			
Calling Peer ID	U	C(=)		
DS-User Version Number	U	C(=)	U	C(=)
Security Requirements	U	C(=)	U	C(=)
Quality Of Service	M	M(=)	U	M(=)
Result			M	M
Reject Source				C
User Data	U	C(=)	U	C(=)

Note 3.— The Called Peer ID parameter is used in the D-START service to specify the name of the intended peer DS-User, and takes an abstract value corresponding to either a 24-bit ICAO aircraft-id or an ICAO facility designator.

Note 4.— The Calling Peer Id parameter is optionally used in the D-START service to specify the name of the initiating DS-User, and is either absent or takes an abstract value corresponding to either a 24-bit ICAO aircraft-id or an ICAO facility designator. Its presence in the indication primitive is conditional upon it being specified by the DS-User in the request primitive.

Note 5.— The DS-User version number allows peer DS-Users to exchange version information. The parameter is optional in the request and response primitives. Its presence in the indication primitive is conditional upon it being specified by the DS-User in the request primitive, and its presence in the confirmation primitive is conditional upon it being specified by the DS-User in the response primitive. If present, it may take any abstract value in the range 1 to 255.

Note 6.— The Security Requirements parameter allows the DS-Users to exchange requirements for security. The parameter is optional in the request and response primitives. Its presence in the indication primitive is conditional upon it being specified by the DS-User in the request primitive, and its presence in the confirmation primitive is conditional upon it being specified by the DS-User in the response primitive.

Note 7.— The Quality Of Service parameter allows the initiating DS-User to specify in the request primitive its requirements for the quality of service (QOS) to be provided for the dialogue. For ATN, the parameter is not modified by the DS-provider, so the value in the indication primitive is equal to the value in the request. The only QOS component which may be modified by the DS-User in the response primitive is Residual Error Rate (see Note 10). Otherwise, the QOS parameter in the response primitive is assumed by the CF to be equal to the value in the indication primitive. The value of the QOS parameter in the confirmation primitive is equal to that present or assumed in the response primitive. The following QOS parameters may be specified:

- a) Routing Class - valid values are defined in Table 5.6-1

- b) Priority - valid values are defined in Table 1-2
- c) Residual Error Rate (RER) - valid values are “low” and “high”.

Note 8.— If the Routing Class parameter is not provided by the DS-User in the D-START Request primitive, and the DS-User is an ATS application as specified in 2.1 - 2.4, then the default value “ATSC: No Traffic Type Policy Preference” is assumed. If the DS-User is not an ATS application as specified in 2.1 - 2.4, then the default traffic type “General Communications” is assumed.

Note 9.— If a Priority value is not provided by the DS-User in the D-START Request primitive, then the default value “network/systems administration” is assumed.

Note 10.— For the RER parameter, “low” means a low error rate, i.e. a high quality connection, and “high” means a higher error rate, i.e. a lower quality connection. The high RER allows non-use of the transport checksum in the ATN. A limited negotiation is possible, such that if the RER value received in the indication primitive is “high”, the DS-User may set the value in the response primitive to either “low” or “high”.

Note 11.— The Result parameter specifies whether the requested dialogue start has been accepted. It can take one of the abstract values:

- a) accepted;
- b) rejected (transient); or
- c) rejected (permanent).

Note 12.— The Reject Source parameter is present if the Result parameter has one of the values “rejected (transient)” or “rejected (permanent)”. It specifies who rejected the start of the dialogue, and can have one of the abstract values:

- a) DS user; or
- b) DS provider.

Note 13.— The User Data parameter allows the peer DS-Users to exchange data during the D-START service invocation. Its presence in the indication primitive is conditional upon it being specified by the DS-User in the request primitive, and its presence in the confirmation primitive is conditional upon it being specified by the DS-User in the response primitive.

4.2.3.3 The D-DATA service

4.2.3.3.1 The behaviour defined by the D-DATA service primitive shall be provided to enable the exchange of information between two DS-Users.

Note 1.— D-DATA is an unconfirmed service which provides data transfer between peer DS-Users. The D-START service must first have been successfully completed to establish the communication relationship between the peers. Request and indication primitives are defined, as illustrated in Figure 4.2-2.

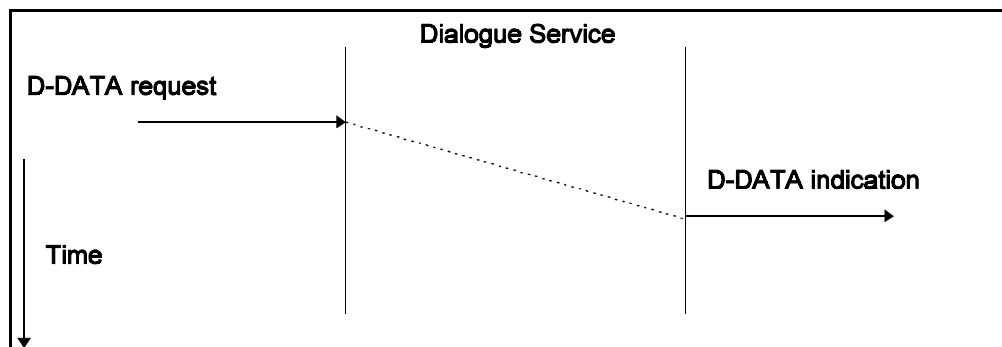


Figure 4.2-2. D-DATA sequence diagram

Note 2.— The parameters of the D-DATA primitives are specified in Table 4.2-5.

Table 4.2-5. D-DATA parameters

<i>Parameter Name</i>	<i>Req</i>	<i>Ind</i>
<i>User Data</i>	<i>M</i>	<i>M(=)</i>

Note 3.— The User Data parameter contains the data to be transferred from a DS-User to its peer, using an existing dialogue.

4.2.3.4 The D-END service

4.2.3.4.1 The behaviour defined by the D-END service primitive shall be provided to enable the orderly termination of a dialogue between two DS-Users.

Note 1.— D-END is a confirmed service which causes the end of a dialogue. It may be invoked by either of the communicating partners. When the D-END service is invoked, the DS performs an orderly release, whereby any service previously invoked is completed before the dialogue is terminated. The D-END service defines request, indication, response and confirmation primitives, as illustrated in Figure 4.2-3.

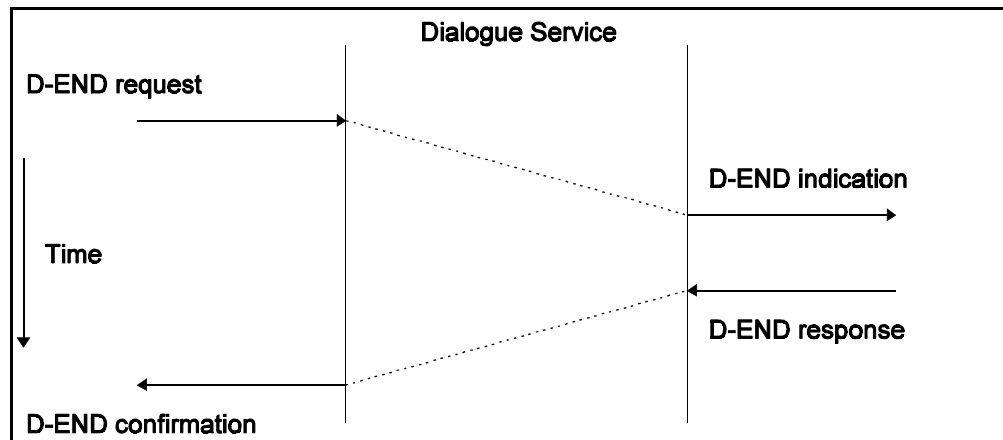


Figure 4.2-3. D-END sequence diagram

Note 2.— The DS-User which wishes to terminate the dialogue issues a D-END request primitive. After issuing a D-END request primitive, the DS-User must not then issue any other service primitive (except D-ABORT if required), until a D-END confirmation is received. After issuing a D-END request primitive, the DS-User must be prepared to continue receiving D-DATA indications from the peer user, until a D-END confirmation primitive is received.

Note 3.— If the D-END confirmation contains a result code of “accepted” then the dialogue no longer exists. If the D-END confirmation contains a result code of “rejected” then the peer DS-User does not wish to terminate the dialogue, and both DS-Users are then free to use the dialogue as if the D-END service had never been invoked.

Note 4.— When a DS-User receives a D-END indication primitive, it may continue to send data using the D-DATA service, but it may at some time issue a D-END response primitive, with a result code of “accepted” or “rejected”. After issuing a D-END response primitive with result “accepted”, a DS-User must not issue any other service primitive, as the dialogue no longer exists. After issuing a D-END response primitive with result “rejected”, a DS-User may issue any other service primitive, as if the D-END service had never been invoked.

Note 5.— The parameters of the D-END primitives are specified in Table 4.2-6.

Table 4.2-6. D-END parameters

Parameter Name	Req	Ind	Rsp	Cnf
Result			M	M(=)
User Data	U	C(=)	U	C(=)

Note 6.— The Result parameter specifies whether the requested dialogue end has been accepted. It can take one of the abstract values: “accepted” or “rejected”.

Note 7.— The User Data parameter contains the data to be transferred from a DS-User to its peer, using an existing dialogue. Its presence in the confirmation primitive is conditional upon it being specified by the DS-User in the response primitive.

Note 8.— In the event of service disruption (e.g. by D-P-ABORT), the invoker of the D-END response primitive will never know that any associated User Data failed to be delivered, as the service is already terminated.

Note 9.— A D-END collision occurs when both peers issue a D-END request primitive near-simultaneously, such that neither peer has yet received the D-END indication primitive corresponding to the remote peer's D-END request. The collision is handled by the CF on behalf of the DS-User. However, one result of the collision handling is that any User Data present in the D-END request will be delivered to the peer in a D-END confirmation primitive, rather than the usual D-END indication. This means that the peer will be unable to react to the contents of the User Data parameter, as the dialogue will have terminated. When a DS-User application is designed such that either peer may terminate the dialogue, then the application can not require a response to any User Data which is sent on a D-END request primitive. The following sequence diagram illustrates the D-END collision from the viewpoint of the two DS-Users:

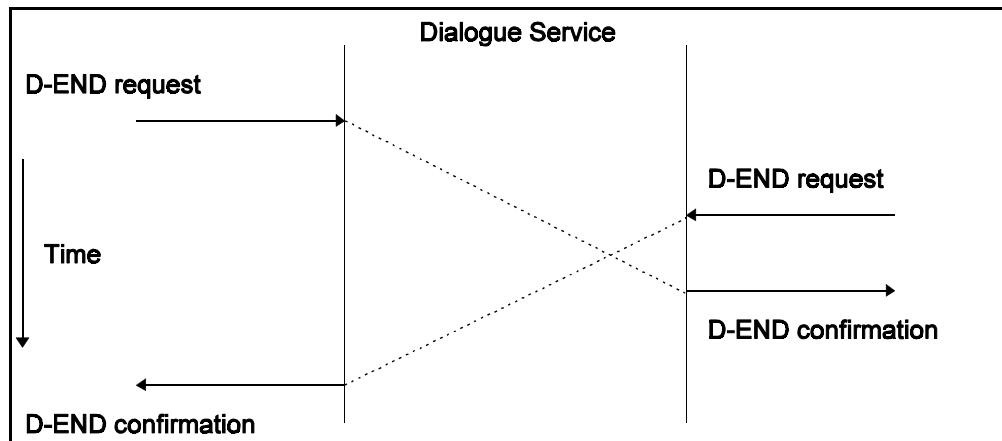


Figure 4.2-4. D-END collision sequence diagram (DS-User view)

4.2.3.5 The D-ABORT service

4.2.3.5.1 The behaviour defined by the D-ABORT service primitive shall be provided to enable the abnormal release of a dialogue between two DS-Users, by either DS-User.

Note 1.— The D-ABORT service request and indication primitives are as illustrated in Figure 4.2-5.

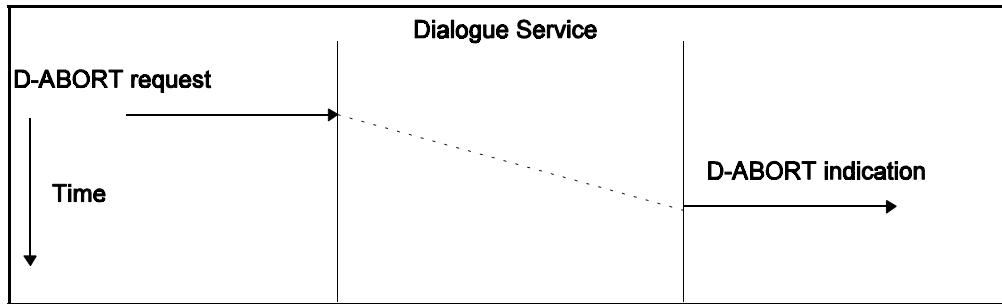


Figure 4.2-5. D-ABORT sequence diagram

Note 2.— When a dialogue is aborted, data in transfer may be lost. The parameters of the D-ABORT primitives are specified in Table 4.2-7.

Table 4.2-7. D-ABORT parameters

<i>Parameter Name</i>	<i>Req</i>	<i>Ind</i>
<i>Originator</i>	<i>U</i>	<i>C(=)</i>
<i>User Data</i>	<i>U</i>	<i>C(=)</i>

Note 3.— The Originator parameter is used to distinguish the source of the abort. Its presence in the indication primitive is conditional upon it being specified by the DS-User in the request primitive. It can take one of the following abstract values:

- a) User — the abort originated from the Application-user; or*
- b) Provider — the abort originated in the ATN-App AE (including the ATN-App ASE).*

Note 4.— If the D-ABORT Originator parameter is not specified, the default value “Provider” is assumed.

Note 5.— The User Data parameter contains the data to be transferred from a DS-User to its peer, using an existing dialogue. Its presence in the indication primitive is conditional upon it being specified by the DS-User in the request primitive. There is no guarantee that the peer will receive the User Data; the sender will not be informed if the User Data is not delivered.

4.2.3.6 The D-P-ABORT service

4.2.3.6.1 The behaviour defined by the D-P-ABORT service primitive shall be provided to indicate an abnormal release of a dialogue by the supporting communications service.

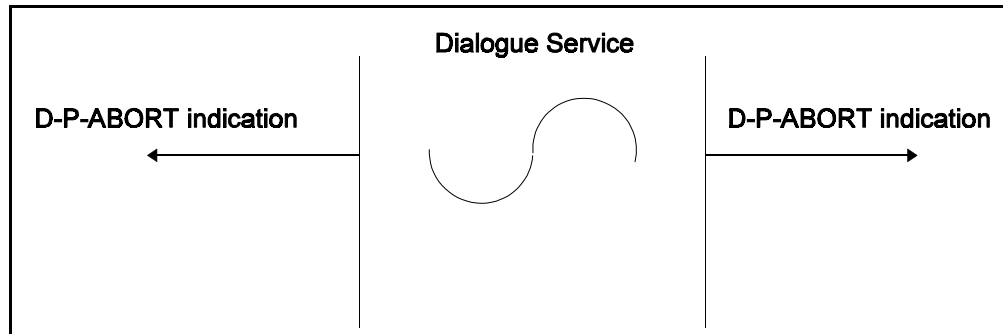


Figure 4.2-6. D-P-ABORT sequence diagram

Note 1.— For the D-P-ABORT service, only an indication primitive is defined, as illustrated in Figure 4.2-6.

Note 2.— The D-P-ABORT service allows the supporting communications service to indicate to the DS-Users that it aborted the dialogue. When the dialogue is aborted, any data in transit may be lost. The D-P-ABORT primitive has no parameters.

4.3 APPLICATION ENTITY (AE) DESCRIPTION

4.3.1 Introduction

4.3.1.1 The ATN-App AE shall exhibit external behaviour as if implemented according to the model shown in Figure 4.3-1, with the protocols defined in ACSE and the ATN-App ASE specifications.

Note 1.— As indicated in 4.1, the AE is described in terms of the Service which it displays to the Application-user, and in terms of the Control Function (CF) which mediates the interactions of the components of the AE.

Note 2.— Figure 4.3-1 also indicates which paragraph describes the behaviour of the CF in response to events at various service boundaries.

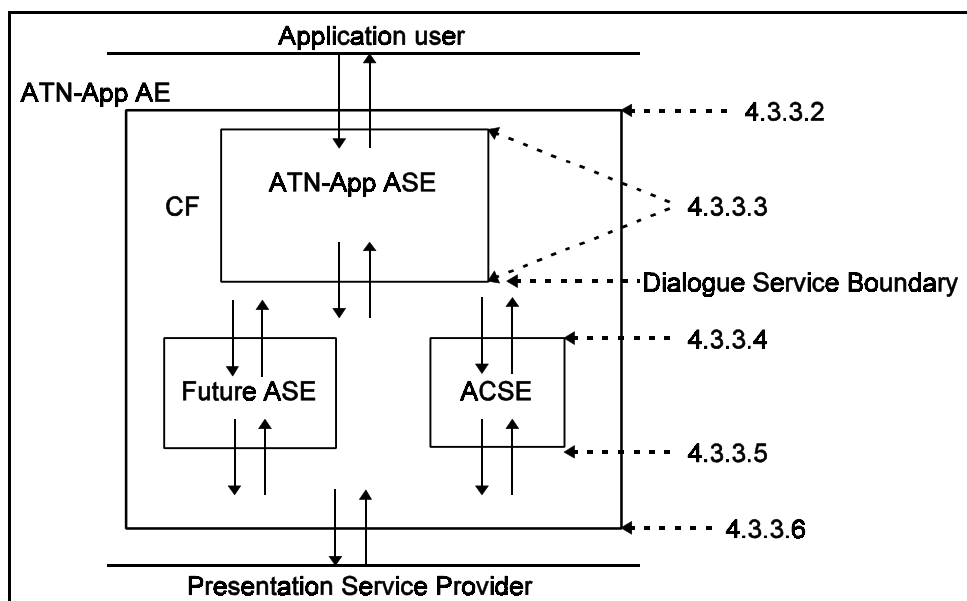


Figure 4.3-1. Components of ATN-App AE

Note 3.— The “Future ASE” component in Figure 4.3-1 indicates how additional functionality can be added in future versions of the ATN Upper Layer Architecture. ACSE provides the basic mechanisms for establishing and releasing an application association. The service provided by the ATN-App AE represents an abstract description of the Application Programming Interface (API) seen by the Application user. The CF of the ATN-App AE specifies how the interactions at the ATN-App AE Service boundary invoke the appropriate service primitives of the constituent ASEs, which in turn generate the actual protocol. The CF also specifies how the constituent ASEs interact with the supporting communications service.

Note 4.— A CF specification is not a service definition of the ATN-App AE or its components. It only defines the actions of the CF as a result of service invocations visible to the CF. Thus, the specification is organised around specifying the response of the CF to these inputs. 4.3.3.2 specifies the actions that result from the inputs of the Application-user. 4.3.3.4 and 4.3.3.5 specify the actions that result from the service invocations of the ACSE component ASE. 4.3.3.3 specifies the actions that result from the service invocations of the ATN-App ASE component ASE. 4.3.3.6 specifies the actions that result from the inputs from the supporting service.

Note 5.— The CF specification describes the overall behaviour of the ATN-App AE. It is not a requirement that an identifiable CF entity be realised in an implementation.

Note 6.— This CF specification assumes that the embedded ASEs (ATN-App ASE and ACSE) are modelled as atomic entities, such that when an input event is invoked by the CF, that event is processed to completion by the ASE and the CF responds to any resulting output events from the ASE, all within the same logical processing thread. This model avoids the need to specify further transient states within the CF. It does not imply any particular implementation architecture.

Note 7.— In the current version of the ATN Upper Layers, the service interface presented to the Application-user is a simple pass-through to the ATN-App ASE. That is, the Application-user passes request and response primitives directly to, and receives indication and confirmation primitives directly from, the ATN-App ASE.

Note 8.— The CF described here supports the four air-ground applications currently defined, and the MHS pass-through application. The specification of the CF for the AIDC application is included in the AIDC specification.

Note 9.— For the purposes of this specification, the ATN-App AE is modelled such that a new instance of communication (effectively a new AE invocation) is implicitly created (a) for each request from the AE-User that will require a new association (i.e., that will result in a D-START request being invoked), and (b) for each indication from the underlying communications service that a new connection is requested. The AE invocation ceases to exist when the underlying communications service connection is disconnected and the CF is idle (i.e., in the NULL state).

4.3.2 Application Level Naming and Context Definition

4.3.2.1 ATN Naming Hierarchy

Note 1.— Names, in the form of object identifiers (OIDs), are assigned here to the defined ATN entities.

Note 2.— ISO/IEC 9834-1 specifies the top of the hierarchical OID name space. At the first level, provision is made for ISO, International Telecommunication Union - Telecommunication Standardisation Sector (ITU-T) and joint ISO/ITU-T sub-name spaces. The ISO name space is further subdivided into:

- a) standard (0)*
- b) registration-authority (1)*
- c) member-body (2)*
- d) identified-organisation (3)*

Note 3.— ICAO has requested and obtained the allocation of an International Code Designator (ICD), according to ISO 6523. The ICD obtained, name and number “icao (27)”, uniquely identifies ICAO and allows ICAO to establish its own object identifier name space within the International Organisation arc using the prefix: { iso (1) identified-organisation (3) icao (27) }. Similarly, IATA has obtained an ICD of “iata (19)”; values assigned under the IATA name space are out of scope.

4.3.2.1.1 Within the ICAO name space, the initial allocation of object identifiers shall follow the structure and values defined here.

Note 1.— In the future, it is likely that the ATN object identifier tree will have further levels of structure, and that fully location-independent values will be assigned.

Note 2.— The ATN naming hierarchy is illustrated in Figure 4.3-2.

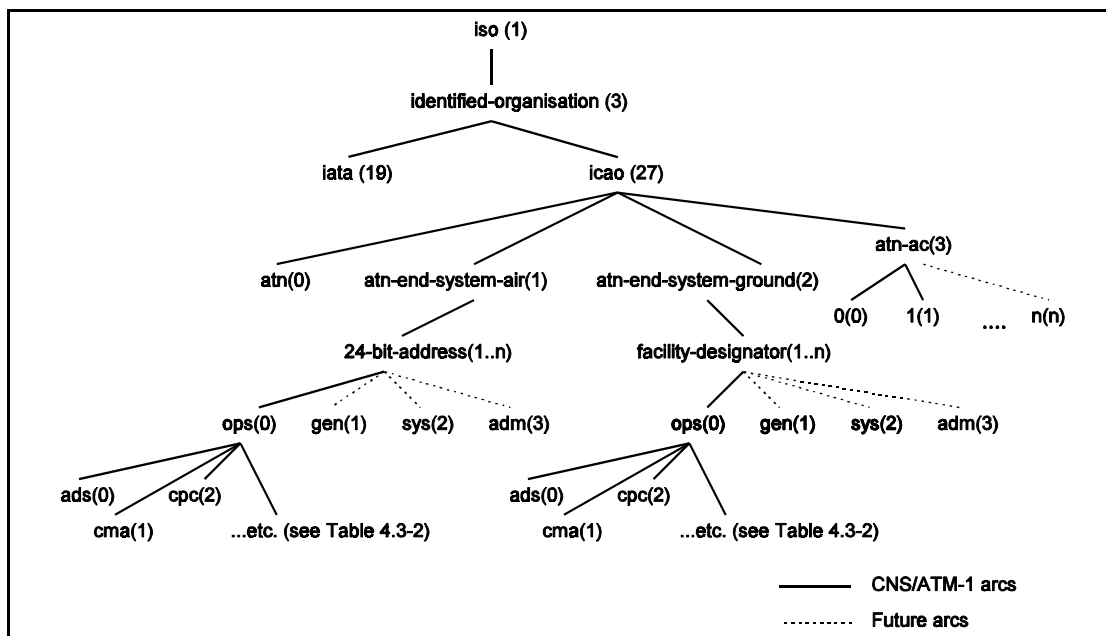


Figure 4.3-2. ATN Naming Hierarchy

4.3.2.1.2 Immediately under the ICAO arc, the values specified in Table 4.3-1 shall be used to specify the next level of the naming hierarchy.

Table 4.3-1. Top-level ICAO Identifiers

Name and numeric value	Description
atn (0)	General ATN identifiers
atn-end-system-air (1)	ATN aircraft end systems. The following OID component beneath this arc is a 24-bit ICAO aircraft identifier
atn-end-system-ground (2)	ATN ground end systems. The following OID component beneath this arc is an ICAO facility designator
atn-ac (3)	ATN application context names

4.3.2.2 Application Process titles

Note.— Application process titles are allocated underneath either of the Object Identifier arcs:

{ atn-end-system-air (1) } or { atn-end-system-ground (2) }.

Immediately subordinate to this arc is an arc whose value is an INTEGER derived from either the 24-bit ICAO aircraft address or the ICAO facility designator, as described in 4.3.2.4. Immediately beneath that arc is an arc whose value is determined by the category of the ATN application. For the present, only the following name and value are defined for the application category:

{ operational (0) }.

4.3.2.2.1 Each application category on each ATN end system shall be assigned an unambiguous application process title (AP-title).

4.3.2.2.2 The AP-title shall be an Object Identifier type (i.e. an AP-title-form2 as defined in ISO/IEC 8650-1).

4.3.2.2.3 Application Process titles shall be of the form:
either:

{ iso (1) identified-organisation (3) icao (27) atn-end-system-air (1) <end-system-id> (n) operational (0) }

or:

{ iso (1) identified-organisation (3) icao (27) atn-end-system-ground (2) <end-system-id> (n) operational (0) }

where:

<end-system-id> is the ICAO 24-bit address for aircraft end systems, or the ICAO facility designator for ground end systems.

(n) is an INTEGER value derived from the <end-system-id>.

Note.— The algorithm for deriving the INTEGER n from the <end-system-id> is defined in 4.3.2.4.

4.3.2.3 Application Entity Titles

4.3.2.3.1 Each ATN application entity shall be assigned an unambiguous application entity title (AE Title).

4.3.2.3.2 For ATN, an AE Title shall be an Object Identifier type as defined in ISO/IEC 8824-1 (i.e. an AE-title-form2 as defined in ISO/IEC 8650-1).

Note.— The AE Title is composed of an Application Process title (AP-title) and an AE-qualifier.

4.3.2.3.3 The AE-qualifier component of the AE Title shall be an INTEGER type (i.e. an AE-qualifier-form2 as defined in ISO/IEC 8650-1).

4.3.2.3.4 The AE-qualifier value arc of the AE Title object identifier represents the ATN application type (e.g. “ADS” or “CMA”), and shall take one of the values specified in Table 4.3-2.

Table 4.3-2. Assigned AE-Qualifier values

ATN ASE type	ATN AE-Qualifier name and numeric value
Automatic Dependent Surveillance	ADS (0)
Context Management Application	CMA (1)
Controller Pilot Data Link Communication	CPC (2)
Automatic Terminal Information Services (ATIS)	ATI (3)
Type A Gateway	GWA (4)
Systems Management Application (SMA)	SMA (5)
ATS Inter-Facility Data Communications (AIDC)	IDC (6)
ATS Message Application	AMS (7)
AFTN-AMHS Gateway	GWB (8)
ATS Message User Agent	AUA (9)
ADS Report Forwarding	ARF (10)
Aviation Routine Weather Report (METAR)	MET (11)
Generic ATN Communication Service AE (GACS)	GAC (12)

4.3.2.3.5 Thus, AE-titles conforming to this definition shall be of the form:
either:

{ iso (1) identified-organisation (3) icao (27) atn-end-system-air (1) <end-system-id> (n) operational (0) <ae-qualifier> (m) }

or

{ iso (1) identified-organisation (3) icao (27) atn-end-system-ground (2) <end-system-id> (n) operational (0) <ae-qualifier> (m) }

where:

<end-system-id> is the ICAO 24-bit address for aircraft end systems, or the ICAO facility designator for ground end systems.

(n) is an INTEGER value derived from the <end-system-id>

<ae-qualifier> is the name form of the AE qualifier from Table 4.3-2

(m) is the number form of the AE qualifier from Table 4.3-2.

Note.— The algorithm for deriving the INTEGER n from the <end-system-id> is defined in 4.3.2.4.

4.3.2.4 Encoding of End System Identifiers

Note 1.— Where <end-system-id> appears as a component of an Object Identifier, the encoding of the OID subidentifier value is obtained as defined in the following text.

Note 2.— For ground stations, the <end-system-id> is derived from an eight-letter facility designator, e.g., “LFPODLHX”. The syntax of the first four letters is defined in ICAO Doc 7910 “Location Indicators”; the syntax of the remaining letters is defined in ICAO Doc 8585 “Designators for Aircraft Operating Agencies, Aeronautical Authorities and Services.”

4.3.2.4.1 For aircraft, the <end-system-id> naming arc shall be the binary value of the 24-bits comprising the ICAO aircraft identifier, expressed as an INTEGER in the range $0..(2^{24}-1)$ and encoded as an Object Identifier subidentifier as defined in ISO/IEC 8825-1.

4.3.2.4.2 For ground stations, the encoding of the <end-system-id> naming arc shall be derived from the ICAO facility designator, which is a sequence of characters from the restricted character set (A..Z), as follows:

- a) Each character is encoded into one octet where:
 - 1) the most significant bit (bit 8) indicates whether the character is the last in the sequence: it is set to zero in the last octet and one in each preceding octet;
 - 2) the next most significant bit (bit 7) is set to zero;
 - 3) the six least significant bits (bits 6 - 1) contain the character encoded as a 6-bit value such that A is encoded as the binary value 000001, B is encoded as 000010, and so on up to Z which is encoded as 011010

Note.— This coding gives compatibility with the Basic Encoding Rules for an Object Identifier subidentifier in ISO/IEC 8825-1. The character coding is equivalent to the “6-bit ASCII” subset of International Alphabet Number 5 (IA5) defined by ITU, which is adopted in SSR Mode S specifications. If required, the encoding can be extended to include numeric characters, with 0 to 9 encoded as binary values 110000 to 111001 respectively, and the space character can be encoded as binary value 100000.

- b) The <end-system-id> is the concatenation of these octets.

Note.— Conceptually, bits 7 - 1 from each octet are concatenated to form an unsigned binary number whose most significant bit is bit 7 of the first octet and whose least significant bit is bit 1 of the last octet.

4.3.2.5 Application Context Names

Note 1.— The Application Context describes the ASE/ASO types which are present in the AE, including those aspects not distinguished by ASO type (e.g. version and policy aspects). The abstract syntax of the APDUs and the control function are described here. The Application Context name is an identifier which is used to refer to a defined Application Context. The syntax of the Application Context name is defined in ISO/IEC 8650-1 as an Object Identifier.

Note 2.— The application context name is used here only to distinguish between different versions of an application context within the scope of a given AE type, as identified by the AE Title.

4.3.2.5.1 The Application Context name shall be used to indicate the version and policy aspects relative to the AE with which it is associated.

4.3.2.5.2 Each Application Context shall be assigned an Application Context name.

4.3.2.5.3 Application Context names shall have the following structure:

{iso (1) identified-organisation (3) icao (27) atn-ac (3) version-<n> (n)}
where n is an INTEGER in the range 0..255.

Note.— The value n = 0 is reserved for use by the CF.

4.3.2.6 Presentation Context Identification

Note.— The Null Encoding presentation protocol option has been selected for the most efficient encoding of presentation PDUs, as defined in 4.5. As a consequence, the conventional presentation protocol mechanisms which enable users of the presentation service to distinguish the presentation context of received APDUs are not available. Therefore, an alternative, application layer, mechanism is defined here.

4.3.2.6.1 All User Data which is passed across the presentation service boundary shall be encoded using the unaligned variant of the Packed Encoding Rules (PER) for ASN.1 (ISO/IEC 8825-2).

4.3.2.6.2 When in the data transfer phase, in order to be able to distinguish APDUs which are defined in different abstract syntax modules, the presentation User Data encoding shall assume the Full Encoding option of ISO/IEC 8823-1, augmented with the PER-visible constraints defined in ISO/IEC 8823-1: 1994/Amd. 1: 1997 as follows:

Note 1.— ISO/IEC 8823-1 specifies two choices for the encoding of User-data:

User-data ::= CHOICE {
 [APPLICATION 0] IMPLICIT Simply-encoded-data,
 [APPLICATION 1] IMPLICIT Fully-encoded-data }

Simply-encoded-data ::= OCTET STRING

Fully-encoded-data ::= SEQUENCE SIZE (1, ...) OF PDV-list

-- contains one or more presentation-data-value-list (PDV-list) values

PDV-list ::= SEQUENCE

{ transfer-syntax-name Transfer-syntax-name OPTIONAL,
presentation-context-identifier Presentation-context-identifier,
presentation-data-values CHOICE

 { single-ASN1-type [0] ABSTRACT-SYNTAX.&Type
 (CONSTRAINED BY {
 -- Type corresponding to presentation context identifier -- }) ,

 octet-aligned [1] IMPLICIT OCTET STRING,

 arbitrary [2] IMPLICIT BIT STRING }

 -- contains one or more presentation data values from the same
 -- presentation context.

}

Transfer-syntax-name ::= OBJECT IDENTIFIER -- not used for ATN Upper Layers

Presentation-context-identifier ::= INTEGER (1..127, ...)

Note 2.— The use of Full Encoding is specified in order to overcome the fact that: (a) the use of presentation protocol efficiency enhancements removes the ability of presentation layer to perform the necessary demarcation, and (b) the use of ASN.1 Packed Encoding Rules means that it would not have been possible to assign unique ASN.1 tag values to individual APDUs to distinguish between them, as PER does not encode tags.

4.3.2.6.3 Only the presentation-context-identifier and presentation-data-values fields shall be present in the encoded presentation User Data.

4.3.2.6.4 Only the “arbitrary” (BIT STRING) choice for presentation-data-values in the PDV-list SEQUENCE shall be used in the encoded presentation User Data.

4.3.2.6.5 The values of Presentation-context-identifier which are pre-defined in Table 4.3-3 shall be used in the encoding of presentation User Data; the presentation-context-identifiers are not dynamically assigned by the presentation service.

Table 4.3-3. Presentation Context identifiers

Presentation-context-identifier value	Short name	Description
1	acse-apdu	ACSE abstract syntax as defined in ISO/IEC 8650-1
2		reserved for future use
3	user-ase-apdu	abstract syntax as defined in individual ATN application specifications
other		reserved for future use

4.3.2.6.6 With the sole exception of the P-CONNECT service (which is used exclusively by ACSE), upon receiving User Data from the presentation service, the CF shall:

- a) decode the Fully-encoded-data and use the presentation-context-identifier value to determine the target ASE.
- b) if the target ASE is ACSE, decode the header of the embedded presentation-data-value to determine the APDU type, and
- c) if the decoding in a) and b) fails for any reason (presentation-context-identifier not recognised, presentation-data-value does not use the “arbitrary” CHOICE value, or unrecognised APDU type) then issue a P-U-ABORT request to the supporting service and behave as if a P-U-ABORT indication with no parameters has been received;
- d) otherwise, pass the presentation-data-value (i.e., acse-apdu or user-ase-apdu) to the target ASE by invoking the appropriate indication or confirmation primitive at the lower ASE service boundary, as specified in 4.3.3.

4.3.2.6.7 Except for P-CONNECT primitives issued by ACSE, when an ASE issues a request or response primitive at its lower service boundary which would otherwise map onto a presentation service primitive, the CF shall:

- a) embed the User Data into a Fully-encoded-data type, using the presentation-context-identifier value corresponding to the source ASE
- b) pass the Fully-encoded User Data to the presentation service by invoking the appropriate primitive, as specified in 4.3.3.

4.3.3 Control Function Specification

4.3.3.1 ATN-App CF State Definitions

4.3.3.1.1 The ATN-App AE shall behave as if it has a Control Function which can exist only in one of the following states:

- a) Null (STA0) — This is the state of the CF when there is no association in existence.
- b) Association Pending (STA1) — The CF enters this state either when the ATN-App ASE has made a request to establish a dialogue and is waiting for notification from its peer OR an indication has been received that the peer has made a request to establish a dialogue.
- c) Data Transfer (STA2) — The CF enters this state once the establishment phase is complete. An association has successfully been established and the communicating partners are free to send and receive data.
- d) Release Pending (STA3) — The CF enters this state when the ATN-App ASE has requested the termination of the dialogue OR an indication has been received that the peer has made a request to terminate the dialogue.
- e) Release Collision (STA4) — The CF enters this state when both communicating partners have requested the termination of the dialogue near-simultaneously.

4.3.3.1.2 CF State Table

4.3.3.1.2.1 The ATN-App AE CF shall behave as if it has a control function in accordance with the state table specified in Table 4.3-4, which shows diagrammatically the state transitions and actions performed by the CF in response to incoming events.

Note.— The following conventions are used in Table 4.3-4:

- a) *Incoming events are shown in the first two columns of the state table, and are enumerated in Table 4.3-6.*
- b) *When an input event occurs and the state table indicates an action, the CF performs that action.*
- c) *Each cell in the state table shows:*
 - 1) *optionally, one or more predicates, denoted “pN”, where N is an integer. The state and action which follow the predicate are only valid if the predicate is TRUE. The inverse (logical NOT) of a predicate is indicated by the prefix “~” (tilde character).*

- 2) *the new state that the CF enters after the action has been performed*
- 3) *the action, if any, which the CF performs. The possible actions are outlined in Table 4.3-7.*
- d) *Blank cells indicate error conditions.*
- e) *When an input event occurs and the state table indicates a state transition, the CF enters the new state after any associated action has been performed.*

4.3.3.1.2.2 For the purpose of specifying CF behaviour, embedded ASEs (ATN-App ASE and ACSE) shall be treated as atomic entities, such that when an input event is invoked by the CF, that event is processed to completion by the ASE and the CF responds to any resulting output events from the ASE, all within the same logical processing thread.

Note.— This provision avoids the need to specify further transient states within the CF. It does not imply any particular implementation architecture.

4.3.3.1.2.3 The following combinations of input events and CF states shall be treated as error conditions:

- a) The occurrence of an input event other than those listed in Table 4.3-6; or
- b) A combination of input event and CF state which corresponds to a blank cell in Table 4.3-4; or
- c) A combination of input event and CF state which corresponds to a cell in Table 4.3-4 containing one or more predicates, none of which evaluates to TRUE.

4.3.3.1.2.4 The error handling shall result in the association being aborted, if one exists, and a notification being given to the Application user.

4.3.3.1.2.5 In the event of a conflict between the actions implied by the state table and the text in the following paragraphs, the text shall take precedence.

Table 4.3-4. ATN-App CF State Table

Event Source	State-->	STA0	STA1	STA2	STA3	STA4
	Event	Null	Assoc. Pending	Data Transfer	Release Pending	Release Collision
From ATN-App	ATN-APP function req	STA0 ATN-App ASE req	STA1 ATN-App ASE req	STA2 ATN-App ASE req	STA3 ATN-App ASE req	STA4 ATN-App ASE req
	ATN-APP function rsp	STA0 ATN-App ASE rsp	STA1 ATN-App ASE rsp	STA2 ATN-App ASE rsp	STA3 ATN-App ASE rsp	STA4 ATN-App ASE rsp
From ATN-App ASE (upper)	ATN-APP function ind	STA0 ATN-App ind	STA1 ATN-App ind	STA2 ATN-App ind	STA3 ATN-App ind	STA4 ATN-App ind
	ATN-APP function cnf	STA0 ATN-App cnf	STA1 ATN-App cnf	STA2 ATN-App cnf	STA3 ATN-App cnf	STA4 ATN-App cnf
From ATN-App ASE (lower)	D-START req	p0: STA1 A-ASSOC req				
	D-START rsp+		~p1: STA1 A-ASSOC rsp+			
	D-START rsp-		~p1: STA1 A-ASSOC rsp-			
	D-DATA req			STA2 P-DATA req (User)	~p2: STA3 P-DATA req (User)	
	D-END req			STA3 A-RELEASE req		
	D-END rsp+				~p2: STA3 A-RELEASE rsp+	
	D-END rsp-				~p2: STA3 A-RELEASE rsp-	
	D-ABORT req		STA1 A-ABORT req	STA2 A-ABORT req	STA3 A-ABORT req	STA4 A-ABORT req

Event Source	State-->	STA0	STA1	STA2	STA3	STA4
	Event	Null	Assoc. Pending	Data Transfer	Release Pending	Release Collision
From ACSE (upper)	A-ASSOCIATE ind		STA1 D-START ind			
	A-ASSOCIATE cnf+		STA2 D-START cnf+			
	A-ASSOCIATE cnf-		STA0 D-START cnf-			
	A-RELEASE ind				STA3 D-END ind	p1: STA4 A-RELEASE rsp+ ~p1: STA4
	A-RELEASE cnf+				STA0 D-END cnf+ P-U-ABORT req	p1: STA0 D-END cnf+ P-U-ABORT req ~p1: STA4 D-END cnf+ A-RELEASE rsp+
	A-RELEASE cnf-				STA2 D-END cnf-	
	A-ABORT ind		STA0 D-ABORT ind	STA0 D-ABORT ind	STA0 D-ABORT ind	STA0 D-ABORT ind
	A-P-ABORT ind		STA0 D-P-ABORT ind	STA0 D-P-ABORT ind	STA0 D-P-ABORT ind	STA0 D-P-ABORT ind
From ACSE (lower)	P-CONNECT req		STA1 P-CONN req			
	P-CONNECT rsp+		STA2 P-CONN rsp+			
	P-CONNECT rsp-		STA0 P-CONN rsp-			
	P-RELEASE req				STA3 P-DATA req (RLRQ)	
	P-RELEASE rsp+				STA0 P-DATA req (RLRE+)	p1: STA4 P-DATA req (RLRE+) ~p1: STA0 P-DATA req (RLRE+)
	P-RELEASE rsp-				STA2 P-DATA req (RLRE-)	
	P-U-ABORT req (data)	STA0 P-U-ABORT req	STA0 P-U-ABORT req	STA0 P-DATA req (ABRT)	STA0 P-U-ABORT req	STA0 P-U-ABORT req
	P-U-ABORT req (no data)		STA0 P-U-ABORT req	STA0 P-U-ABORT req	STA0 P-U-ABORT req	STA0 P-U-ABORT req

Event Source	State-->	STA0	STA1	STA2	STA3	STA4
	Event	Null	Assoc. Pending	Data Transfer	Release Pending	Release Collision
From supporting service	P-CONNECT ind	p0: STA1 P-CONN ind				
	P-CONNECT cnf+		STA1 P-CONN cnf+			
	P-CONNECT cnf-		STA1 P-CONN cnf-			
	P-DATA ind (RLRQ)	p3: STA0		STA3 P-RELEASE ind	p2: STA4 P-RELEASE ind	
	P-DATA ind (RLRE+)	p3: STA0			STA3 P-RELEASE cnf+	STA4 P-RELEASE cnf+
	P-DATA ind (RLRE-)	p3: STA0			STA3 P-RELEASE cnf-	
	P-DATA ind (ABRT)	p3: STA0 P-U-ABORT req ~p3: STA0		STA2 P-U-ABORT ind P-U-ABORT req	STA3 P-U-ABORT ind P-U-ABORT req	STA4 P-U-ABORT ind P-U-ABORT req
	P-DATA ind (User)	p3: STA0		STA2 D-DATA ind	p2: STA3 D-DATA ind	
	P-U-ABORT ind	STA0	STA1 P-U-ABORT ind	STA2 P-U-ABORT ind	STA3 P-U-ABORT ind	STA4 P-U-ABORT ind
	P-P-ABORT ind	STA0	STA1 P-P-ABORT ind	STA2 P-P-ABORT ind	STA3 P-P-ABORT ind	STA4 P-P-ABORT ind

Table 4.3-5. Predicates used in Table 4.3-4

Predicate	Meaning
p0	This is a new instance of communication, i.e., no previous association exists (effectively, a new AE invocation is created).
p1	This CF is the initiator CF, i.e., the CF which issued an A-ASSOCIATE request primitive.
~p1	This CF is the responder CF, i.e., the CF which received an A-ASSOCIATE indication primitive.
p2	This CF is the Release Initiator, i.e., the CF issued an A-RELEASE request primitive.
~p2	This CF is the Release Responder, i.e., the CF received an A-RELEASE indication primitive.
p3	This CF is the "Abort+Data" initiator, i.e., the CF issued a P-DATA request containing an ABRT APDU and is awaiting disconnection by the peer.
~p3	This CF has not initiated an Abort containing user data.

Table 4.3-6. Incoming Event List

Abbreviated name	Source	Description
ATN-APP function req	upper AE service boundary	Application-specific Request primitive issued by the Application User
ATN-APP function rsp		Application-specific Response primitive issued by the Application User
ATN-APP function ind	ATN-App ASE (upper service boundary)	Application-specific Indication primitive issued by the Application ASE
ATN-APP function cnf		Application-specific Confirmation primitive issued by the Application ASE
D-START req	ATN-App ASE (lower service boundary)	D-START Request primitive issued by DS-User
D-START rsp+		D-START Response primitive issued by DS-User, with Result = accepted
D-START rsp-		D-START Response primitive issued by DS-User, with Result = rejected (transient) or rejected (permanent)
D-DATA req		D-DATA Request primitive issued by DS-User
D-END req		D-END Request primitive issued by DS-User
D-END rsp+		D-END Response primitive issued by DS-User, with Result = accepted
D-END rsp-		D-END Response primitive issued by DS-User, with Result = rejected
D-ABORT req		D-ABORT Request primitive issued by DS-User
A-ASSOCIATE ind	ACSE (upper service boundary)	A-ASSOCIATE Indication primitive issued by ACSE service
A-ASSOCIATE cnf+		A-ASSOCIATE Confirmation primitive issued by ACSE service, with Result = accepted
A-ASSOCIATE cnf-		A-ASSOCIATE Confirmation primitive issued by ACSE service, with Result = rejected (transient) or rejected (permanent)
A-RELEASE ind		A-RELEASE Indication primitive issued by ACSE service
A-RELEASE cnf+		A-RELEASE Confirmation primitive issued by ACSE service, with Result = affirmative
A-RELEASE cnf-		A-RELEASE Confirmation primitive issued by ACSE service, with Result = negative
A-ABORT ind		A-ABORT Indication primitive issued by ACSE service
A-P-ABORT ind		A-P-ABORT Indication primitive issued by ACSE service
P-CONNECT req	ACSE (lower service boundary)	P-CONNECT Request primitive issued by ACSE Protocol Machine (ACPM)
P-CONNECT rsp+		P-CONNECT Response primitive issued by ACPM, with Result = acceptance
P-CONNECT rsp-		P-CONNECT Response primitive issued by ACPM, with Result = user-rejection or provider-rejection
P-RELEASE req		P-RELEASE Request primitive issued by ACPM
P-RELEASE rsp+		P-RELEASE Response primitive issued by ACPM, with Result = affirmative
P-RELEASE rsp-		P-RELEASE Response primitive issued by ACPM, with Result = negative

Abbreviated name	Source	Description
P-U-ABORT req (data)		P-U-ABORT Request primitive issued by ACPM, with the User Data parameter present.
P-U-ABORT req (no data)		P-U-ABORT Request primitive issued by ACPM, with the User Data parameter empty or absent.
P-CONNECT ind	supporting service	P-CONNECT Indication primitive issued by presentation service provider
P-CONNECT cnf+		P-CONNECT Confirmation primitive issued by presentation service provider, with Result = acceptance
P-CONNECT cnf-		P-CONNECT Confirmation primitive issued by presentation service provider, with Result = user-rejection or provider-rejection
P-DATA ind (RLRQ)		P-DATA Indication primitive issued by presentation service provider, with a RLRQ APDU as User-Data
P-DATA ind (RLRE+)		P-DATA Indication primitive issued by presentation service provider, with a RLRE APDU as User-Data, with the reason field set to "normal"
P-DATA ind (RLRE-)		P-DATA Indication primitive issued by presentation service provider, with a RLRE APDU as User-Data, with the reason field set to "not-finished"
P-DATA ind (ABRT)		P-DATA Indication primitive issued by presentation service provider, with an ABRT APDU as User-Data
P-DATA ind (User)		P-DATA Indication primitive issued by presentation service provider, with an ATN-APP APDU (e.g. an ADS-ASE protocol data unit) as User-Data
P-U-ABORT ind		P-U-ABORT Indication primitive issued by presentation service provider
P-P-ABORT ind		P-P-ABORT Indication primitive issued by presentation service provider

Table 4.3-7. Outgoing Event List

Abbreviated name	Target	Description
ATN-App ind	upper AE service boundary	Application-specific Indication primitive mapped transparently from the upper service boundary of the ATN-App ASE.
ATN-App cnf		Application-specific Confirmation primitive mapped transparently from the upper service boundary of the ATN-App ASE.
ATN-App ASE req	upper ATN-App ASE service boundary	Application-specific Request primitive mapped transparently from the upper AE service boundary
ATN-App ASE rsp		Application-specific Response primitive mapped transparently from the upper AE service boundary.
D-START ind	DS-User	D-START Indication primitive issued.
D-START cnf+		D-START Confirmation primitive issued, with the Result parameter set to the abstract value "accepted"
D-START cnf-		D-START Confirmation primitive issued, with the Result parameter set to the abstract value "rejected (transient)" or "rejected (permanent)", according to the A-ASSOCIATE Confirmation primitive which was received.
D-DATA ind		D-DATA Indication primitive issued.
D-END ind		D-END Indication primitive issued.

Abbreviated name	Target	Description
D-END cnf+		D-END Confirmation primitive issued, with the Result parameter set to the abstract value “accepted”.
D-END cnf-		D-END Confirmation primitive issued, with the Result parameter set to the abstract value “rejected”.
D-ABORT ind		D-ABORT Indication primitive issued.
D-P-ABORT ind		D-P-ABORT Indication primitive issued.
A-ASSOC req	ACSE service provider	A-ASSOCIATE Request primitive issued
A-ASSOC rsp+		A-ASSOCIATE Response primitive issued, with Result = “accepted”
A-ASSOC rsp-		A-ASSOCIATE Response primitive issued, with Result = “rejected (transient)” or “rejected (permanent)”, according to the D-START response primitive which was received.
A-RELEASE req		A-RELEASE Request primitive issued.
A-RELEASE rsp+		A-RELEASE Response primitive issued, with Result = “affirmative” and Reason = “normal”
A-RELEASE rsp-		A-RELEASE Response primitive issued, with Result = “negative” and Reason = “not-finished”
A-ABORT req		A-ABORT Request primitive issued.
P-CONN ind	lower ACSE service boundary	P-CONNECT Indication primitive invoked.
P-CONN cnf+		P-CONNECT Confirmation primitive invoked, with the Result parameter set to “acceptance”.
P-CONN cnf-		P-CONNECT Confirmation primitive invoked, with the Result parameter set to “user-rejection”.
P-RELEASE ind		P-RELEASE Indication primitive invoked.
P-RELEASE cnf+		P-RELEASE Confirmation primitive invoked, with the Result parameter set to “affirmative”.
P-RELEASE cnf-		P-RELEASE Confirmation primitive invoked, with the Result parameter set to “negative”.
P-U-ABORT ind		P-U-ABORT Indication primitive invoked.
P-P-ABORT ind		P-P-ABORT Indication primitive invoked.
P-CONN req	supporting service	P-CONNECT Request primitive issued.
P-CONN rsp+		P-CONNECT Response primitive issued, with the Result parameter set to “acceptance”.
P-CONN rsp-		P-CONNECT Response primitive issued, with the Result parameter set to “user-rejection”.
P-DATA req (RLRQ)		P-DATA Request primitive issued. The User Data parameter contains a RLRQ APDU.
P-DATA req (RLRE+)		P-DATA Request primitive issued. The User Data parameter contains a RLRE APDU, with the reason field set to “normal”.
P-DATA req (RLRE-)		P-DATA Request primitive issued. The User Data parameter contains a RLRE APDU, with the reason field set to “not-finished”.

Abbreviated name	Target	Description
P-DATA req (ABRT)		P-DATA Request primitive issued. The User Data parameter contains an ABRT APDU, with a non-empty user-information field.
P-DATA req (User)		P-DATA Request primitive issued. The User Data parameter contains an ATN-App ASE APDU (e.g. an ADS-ASE protocol data unit)
P-U-ABORT req		P-U-ABORT Request primitive issued.

4.3.3.2 Services Invoked by the Application User

Note 1.— The actions that result from inputs generated by the user of this ATN-App AE (see Figure 4.3-1) are defined here.

Note 2.— The service primitives available to the Application User are specific to the ATN application. This service is detailed in the individual application specifications.

4.3.3.2.1 When Invoked

4.3.3.2.1.1 Invocations of Application User Request and Response primitives by the Application-user shall be allowed when the CF is in any valid state.

4.3.3.2.2 Action Upon Invocation

4.3.3.2.2.1 When the Application User Request or Response primitive is issued, the CF shall:

- a) Invoke the equivalent primitive of the ATN-App ASE service, with a one-to-one mapping of parameters; and
- b) Remain in its current state.

4.3.3.3 Services Invoked by ATN-App ASE

4.3.3.3.1 ATN-App ASE Indication and Confirmation primitives

4.3.3.3.1.1 When Invoked

4.3.3.3.1.1.1 Invocations of ATN-App ASE Indication and Confirmation primitives by the ATN-App ASE shall be allowed when the CF is in any valid state.

4.3.3.3.1.2 Action Upon Invocation

4.3.3.3.1.2.1 When the ATN-App ASE Indication or Confirmation primitive is issued, the CF shall:

- a) Invoke the equivalent primitive of the Application-user service with a one-to-one mapping of parameters; and
- b) Remain in its current state.

4.3.3.3.2 D-START Request primitive

4.3.3.3.2.1 When Invoked

4.3.3.3.2.1.1 When the D-START Request primitive is invoked by the ATN-App ASE, a new instance of communication shall be created, with its CF initially in the NULL state.

4.3.3.3.2.2 Action Upon Invocation

4.3.3.3.2.2.1 When the D-START Request is validly invoked, the CF shall:

- a) Retrieve the AE-qualifier as defined for the ATN-App AE,
- b) Construct the Application Context name, with the value of the final arc set equal to the DS-User Version Number parameter if provided, and set to zero otherwise.
- c) Retrieve the calling Presentation address
- d) Look up the called Presentation address from the Called Peer Id parameter.
- e) If the Calling Peer Id parameter is present, then retrieve the Calling AP Title and Calling AE-qualifier. If it is not present, then Calling AP Title and Calling AE-qualifier are not used in the A-ASSOCIATE request (and they will not then be included in the resulting A-ASSOCIATE-REQUEST (AARQ) APDU).

Note.— The way that the Calling AP Title and the Calling AE-Qualifier are retrieved is a local implementation matter.

- f) If the Security Requirements parameter is not present, make no use of the A-ASSOCIATE parameter “ACSE Requirements”. If the Security Requirements parameter is present, set the ACSE Requirements parameter to the symbolic value “authentication”; and map the Security Requirements value to the A-ASSOCIATE Authentication-value parameter.
- g) Construct an A-ASSOCIATE Request primitive with the following parameters:

Table 4.3-8.

A-ASSOCIATE Request parameter	ISO Status	ATN value
Mode	U	Not used (default value)
Application Context Name	M	As derived in b) above
Application Context Name List	C	Not used
Calling AP Title	U	As derived in e) above
Calling AE Qualifier	U	As derived in e) above
Calling AP Invocation-identifier	U	Not used
Calling AE Invocation-identifier	U	Not used

A-ASSOCIATE Request parameter	ISO Status	ATN value
Called AP Title	U	Not used
Called AE Qualifier	U	Not used
Called AP Invocation-identifier	U	Not used
Called AE Invocation-identifier	U	Not used
ACSE Requirements	U	As derived in f) above
Authentication-mechanism Name	U	Not used
Authentication-value	U	As derived in f) above
User Information	U	D-START User Data parameter
Calling Presentation Address	M	Derived as in c) above
Called Presentation Address	M	Derived as in d) above
Presentation Context Definition List	U	Not used
Default Presentation Context Name	U	Not used
Quality of Service	M	See following subsection
Presentation Requirements	U	Not used (default value)
Session Requirements	M	No Orderly Release (NOR), Duplex
Initial Synchronization Point Serial No	C	Not used
Initial Assignment of Tokens	C	Not used
Session-connection Identifier	U	Not used

- h) Invoke the A-ASSOCIATE Request primitive
- i) Enter the ASSOCIATION PENDING state as an initiator CF.

4.3.3.3.2.3 Quality of Service parameter mappings

Note.— The following paragraphs specify how the Quality of Service parameters in D-START Request and Response primitives are conveyed to the ATN Internet.

4.3.3.3.2.3.1 The Routing Class component of the quality of service parameter in D-START Request and Response primitives shall be conveyed to the ATN Internet and mapped to ATN Security Label by local means, using the values for Security Tag Value specified in Table 5.6-1.

Note.— 5.2.7.3.1 states that the mechanism by which the connection initiator provides the appropriate ATN Security Label is a local matter. For example, it may be identified by an extension to the transport service interface, be implicit in the choice of a given Transport Service Access Point (TSAP), or be identified using a Systems Management function.

4.3.3.3.2.3.2 If no value for Routing Class is specified in the D-START Request primitive, then a default value shall be assigned as follows:

- a) If the ATN-App AE is one of the ATS applications specified in 2.1 - 2.4, the value corresponding to “ATSC: No Traffic Type Policy Preference” is assigned;
- b) otherwise, the traffic type defaults to General Communications, and no Security Tag Value is conveyed.

4.3.3.3.2.3.3 The Routing Class value conveyed to the ATN Internet when the D-START Response primitive is invoked shall be the same as that which was passed to the DS-User in the D-START Indication primitive.

4.3.3.3.2.3.4 The Priority component of the quality of service parameter in D-START Request and Response primitives shall be provided to the TS-Provider, by implementation-specific means, using the values for “Transport Layer Priority” specified in Table 1-2.

Note.— Although transport priority and network priority are semantically independent of each other, 5.5.1.2 requires that the Transport Service (TS)-user specifies the Application Service Priority, which in turn is mapped into the resulting Connectionless Network Protocol (CLNP) PDUs according to Table 1-2, which defines the fixed relationship between transport priority and the network priority.

4.3.3.3.2.3.5 If no value for Priority is specified in the D-START Request primitive, then the value corresponding to “Network/systems administration” shall be used.

4.3.3.3.2.3.6 The Priority value conveyed when the D-START Response primitive is invoked shall be the same as that which was passed to the DS-User in the D-START Indication primitive.

4.3.3.3.2.3.7 The residual error rate (RER) component of the quality of service parameter in D-START Request and Response primitives shall map to the residual error rate component of the A-ASSOCIATE Quality of Service parameter, and is used to convey requests for the use or non-use of transport checksum to the TS-Provider.

Note.— 5.5.1.2 requires that the TS-User specifies the required residual error rate to determine whether or not the transport checksum is required.

4.3.3.3.2.3.8 If no valid value for RER is specified in the D-START Response primitive, then the value shall be taken to be the same as that which was passed to the DS-User in the D-START Indication primitive.

Note.— If the RER value in the D-START Indication was “high”, then valid values in the response are “low” and “high”. If the RER value in the D-START Indication was “low”, then the only valid value in the response is “low”.

4.3.3.3.3 D-START Response primitive

4.3.3.3.3.1 When Invoked

4.3.3.3.3.1.1 The D-START Response primitive may be validly invoked by the ATN-App ASE when the CF is the responder CF (see 4.3.3.6.1.2.1) and is in the ASSOCIATION PENDING state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.3.3.2 Action Upon Invocation

4.3.3.3.3.2.1 When a D-START Response primitive is validly invoked, the CF shall :

- a) Construct the Application Context name, with the value of the final arc set equal to the DS-User Version Number parameter if provided, and set to zero otherwise.
- b) Retrieve the responding Presentation address
- c) If the Security Requirements parameter is not present, make no use of the A-ASSOCIATE parameter “ACSE Requirements”. If the Security Requirements parameter is present, set the ACSE Requirements parameter to the symbolic value “authentication”; and map the Security Requirements value to the A-ASSOCIATE Authentication-value parameter.
- d) Construct an A-ASSOCIATE Response primitive with the following parameters:

Table 4.3-9.

A-ASSOCIATE Response parameter	ISO Status	ATN Value
Application Context Name	M	As derived in a) above
Application Context Name List	C	Not used
Responding AP Title	U	Not used
Responding AE Qualifier	U	Not used
Responding AP Invocation-identifier	U	Not used
Responding AE Invocation-identifier	U	Not used
ACSE Requirements	C	As derived in c) above
Authentication-mechanism Name	U	Not used
Authentication-value	U	As derived in c) above
User Information	U	D-START User Data parameter
Result	M	D-START Result parameter
Diagnostic	U	Not used
Responding Presentation Address	M	Derived as in b) above
Presentation Context Definition Result List	C	Not used

A-ASSOCIATE Response parameter	ISO Status	ATN Value
Default Presentation Context Result	C	Not used
Quality of Service	M	As for D-START Request (see preceding section)
Presentation Requirements	U	Not used (default value)
Session Requirements	M	No Orderly Release (NOR), Duplex
Initial Synchronization Point Serial No	C	Not used
Initial Assignment of Tokens	C	Not used
Session-connection Identifier	U	Not used

- e) If the D-START Response *Result* parameter has the abstract value “accepted”, invoke an A-ASSOCIATE Response primitive with the Result parameter set to “accepted”, and remain in the ASSOCIATION PENDING state.
- f) If the D-START Response *Result* parameter has the abstract value “rejected (permanent)” or “rejected (transient)”, invoke an A-ASSOCIATE Response primitive with the Result parameter set to the same abstract value, and remain in the ASSOCIATION PENDING state.

4.3.3.3.4 D-END Request primitive

4.3.3.3.4.1 When Invoked

4.3.3.3.4.1.1 The D-END Request primitive may be validly invoked by the ATN-App ASE when the CF is in the DATA TRANSFER state; if it is in any other state then appropriate error recovery action shall be taken.

Note.— For example, if the CF is in the RELEASE PENDING state, then the D-END Request is rejected locally, with an appropriate result code.

4.3.3.3.4.2 Action Upon Invocation

4.3.3.3.4.2.1 When a D-END Request primitive is validly invoked, the CF shall :

- a) Construct an A-RELEASE Request primitive with the following parameter values:

Table 4.3-10.

A- RELEASE Request parameter	ISO Status	ATN Value
Reason	U	“normal”
User Information	U	D-END User Data parameter

- b) Invoke the A-RELEASE Request primitive; and

- c) Enter the RELEASE PENDING state as the Release Initiator CF.

4.3.3.3.5 D-END Response primitive

4.3.3.3.5.1 When Invoked

4.3.3.3.5.1.1 The D-END Response primitive may be validly invoked by the ATN-App ASE when the CF is the Release Responder CF and is in the RELEASE PENDING state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.3.5.2 Action Upon Invocation

4.3.3.3.5.2.1 When a D-END Response primitive is validly invoked and the *Result* parameter has the value “accepted”, the CF shall:

- a) Construct an A-RELEASE Response primitive with parameter values as follows:

Table 4.3-11.

A- RELEASE Response parameter	ISO Status	ATN Value
Reason	U	“normal”
User Information	U	D-END User Data parameter
Result	M	“affirmative”

- b) Invoke the A-RELEASE Response primitive

- c) Remain in the RELEASE PENDING state.

4.3.3.3.5.2.2 When a D-END Response primitive is validly invoked and the *Result* parameter has the abstract value “rejected” the CF shall:

- a) Construct an A-RELEASE Response primitive with parameter values as follows:

Table 4.3-12.

A- RELEASE Response parameter	ISO Status	ATN Value
Reason	U	“not finished”
User Information	U	D-END User Data parameter
Result	M	“negative”

- b) Invoke the A-RELEASE Response primitive; and

- c) Remain in the RELEASE PENDING state.

4.3.3.3.6 D-DATA Request primitive

4.3.3.3.6.1 When Invoked

4.3.3.3.6.1.1 The D-DATA Request primitive may be validly invoked by the ATN-App ASE when the CF is in the DATA TRANSFER state, or (if it is the Release Responder) in the RELEASE PENDING state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.3.6.2 Action Upon Invocation

4.3.3.3.6.2.1 When a D-DATA Request primitive is validly invoked, the CF shall :

- a) Using the definition of presentation-user-data in 4.3.2.6, encode the D-DATA Request User Data parameter with presentation-context-identifier value corresponding to “user-ase-apdu”;
- b) Invoke a P-DATA Request primitive with the resulting encoding as User Data; and
- c) Remain in the same state.

4.3.3.3.7 D-ABORT Request primitive

4.3.3.3.7.1 When Invoked

4.3.3.3.7.1.1 Invocations of the D-ABORT Request primitive by the ATN-App ASE shall be allowed when the CF is in any valid state, except the NULL state; if an invocation occurs when the CF is in the NULL state then an error has occurred (see 4.3.3.1.2.4).

4.3.3.3.7.2 Action Upon Invocation

4.3.3.3.7.2.1 When a D-ABORT Request primitive is validly invoked, the CF shall:

- a) If the Originator parameter of the D-ABORT has the symbolic value “User”, then set Diagnostic to “No reason given”. If the Originator parameter is absent or has any symbolic value other than “User”, then set Diagnostic to “Protocol error”.
- b) Construct an A-ABORT Request primitive with the following parameter values:

Table 4.3-13.

A-ABORT Request parameter	ISO Status	ATN Value
Diagnostic	U	derived as in a) above
User Information	U	D-ABORT User Data parameter, if present and not empty.

- c) Invoke the A-ABORT Request primitive; and
- d) Remain in the same state.

4.3.3.4 ACSE Services delivered to the CF

Note.— Events which occur at the upper service boundary of ACSE, i.e. Indication and Confirmation primitives which are generated by the ACPM and which require handling by the CF, are defined here.

4.3.3.4.1 A-ASSOCIATE Indication primitive

4.3.3.4.1.1 When Invoked

4.3.3.4.1.1.1 The A-ASSOCIATE Indication primitive may be validly invoked by the ACSE Protocol Machine (ACPM) when the CF is in the ASSOCIATION PENDING state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.4.1.2 Action Upon Invocation

4.3.3.4.1.2.1 When an A-ASSOCIATE Indication primitive is validly invoked, the CF shall:

- a) If the final component of the Application Context Name parameter is non-zero, then use it as the DS-User Version Number in the D-START Indication primitive. If it has the value zero, then omit the DS-User Version Number parameter in the D-START Indication.
- b) If the Calling AP Title parameter is present, extract the Calling Peer Id from it.
- c) If the ACSE Requirements parameter is present, and it indicates that the authentication functional unit is requested, then extract the Authentication-value parameter.
- d) Construct a D-START Indication primitive, with the following parameter values:

Table 4.3-14.

D-START Indication parameter	Value
Calling Peer ID	Derived as in b) above
DS-User Version Number	Derived as in a) above
Security Requirements	Derived as in c) above
Quality Of Service	See following subsection
User Data	A-ASSOCIATE User Information parameter

- e) Invoke the D-START Indication primitive; and

- f) Remain in the ASSOCIATION PENDING state.

4.3.3.4.1.3 Quality of Service parameter mappings

Note.— The following paragraphs specify how the Quality of Service parameters in A-ASSOCIATE Indication and Confirmation primitives are conveyed to the DS-User as parameters of the D-START Indication and Confirmation primitives.

4.3.3.4.1.3.1 The Routing Class component of the quality of service parameter in D-START indication and confirmation primitives shall be obtained from the ATN Internet by local means, using the abstract values for Security Tag Values as specified in Table 5.6-1.

4.3.3.4.1.3.2 The Priority component of the quality of service parameter in D-START indication and confirmation primitives shall be taken from information provided by the TS-Provider, by implementation-specific means, using the abstract values for “Transport Layer Priority” specified in Table 1-2.

4.3.3.4.1.3.3 The RER component of the quality of service parameter in D-START indication and confirmation primitives shall be taken from the residual error rate component of the A-ASSOCIATE Quality of Service parameter.

4.3.3.4.2 A-ASSOCIATE Confirmation primitive

4.3.3.4.2.1 When Invoked

4.3.3.4.2.1.1 The A-ASSOCIATE Confirmation primitive may be validly invoked by the ACPM when the CF is in the ASSOCIATION PENDING state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.4.2.2 Action Upon Invocation

4.3.3.4.2.2.1 When an A-ASSOCIATE Confirmation primitive is validly invoked, and the *Result* parameter has the abstract value “accepted” the CF shall:

- a) If the final component of the Application Context Name parameter is non-zero, then use it as the DS-User Version Number in the D-START Confirmation primitive. If it has the value zero, then omit the DS-User Version Number parameter in the D-START Confirmation.
- b) If the ACSE Requirements parameter is present, and it indicates that the authentication functional unit is selected, then extract the Authentication-value parameter.
- c) Construct a D-START Confirmation primitive, with parameter values as follows:

Table 4.3-15.

D-START Confirmation parameter	Value
DS-User Version Number	Derived as in a) above
Security Requirements	Derived as in b) above
Quality Of Service	As for A-ASSOCIATE Indication (see preceding section)
Result	“accepted”
Reject Source	Not used
User Data	A-ASSOCIATE User Information parameter

- d) Invoke the D-START Confirmation primitive
- e) Enter the DATA TRANSFER state as the initiator CF.

4.3.3.4.2.2.2 When an A-ASSOCIATE Confirmation primitive is validly invoked, and the *Result* parameter has the abstract value “rejected (permanent)” or “rejected (transient)” the CF shall:

- a) If the final component of the Application Context Name parameter is non-zero, then use it as the DS-User Version Number in the D-START Confirmation primitive. If it has the value zero, then omit the DS-User Version Number parameter in the D-START Confirmation primitive.
- b) If the ACSE Requirements parameter is present, and it indicates that the authentication functional unit is selected, then extract the Authentication-value parameter.
- c) If the A-ASSOCIATE Confirmation *Result Source* parameter has the abstract value “ACSE service-user” form a Reject Source parameter with value “DS user”. If the A-ASSOCIATE Confirmation *Result Source* parameter has the abstract value “ACSE service-provider” or “presentation service-provider” form a Reject Source parameter with value “DS provider”.
- d) Construct a D-START Confirmation primitive, with parameter values as follows:

Table 4.3-16.

D-START Confirmation parameter	Value
DS-User Version Number	Derived as in a) above
Security Requirements	Derived as in b) above
Quality Of Service	As for A-ASSOCIATE Indication (see preceding section)
Result	“rejected (permanent)” or “rejected (transient)”, from the A-ASSOCIATE Result parameter
Reject Source	Derived as in c) above
User Data	A-ASSOCIATE User Information parameter

- e) Invoke the D-START Confirmation primitive
- f) Enter the NULL state.

4.3.3.4.3 A-RELEASE Indication primitive

4.3.3.4.3.1 When Invoked

4.3.3.4.3.1.1 The A-RELEASE Indication primitive may be validly invoked by the ACPM when the CF is in the RELEASE PENDING or the RELEASE COLLISION state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.4.3.2 Action Upon Invocation

4.3.3.4.3.2.1 When an A-RELEASE Indication primitive is validly invoked, and the CF is in the RELEASE PENDING state, it shall:

- a) Construct a D-END Indication primitive, with the User Data parameter set equal to the value of the User Information parameter of the A-RELEASE Indication primitive.
- b) Invoke the D-END Indication
- c) Remain in the RELEASE PENDING state.

4.3.3.4.3.2.2 When an A-RELEASE Indication primitive is validly invoked, and the CF is in the RELEASE COLLISION state, and it is the Initiator CF, it shall:

- a) Construct a D-END Confirmation primitive, with the User Data parameter set equal to the value of the User Information parameter of the A-RELEASE Indication primitive, if present.

Note.— The D-END Confirmation is not issued to the DS-User until the orderly release procedure is complete, and an A-RELEASE Confirmation is received.

- b) Construct an A-RELEASE response primitive with parameter values as follows:

Table 4.3-17.

A- RELEASE Response parameter	ISO Status	ATN Value
Reason	U	“normal”
User Information	U	Not present
Result	M	“affirmative”

- c) Invoke the A-RELEASE Response primitive; and
- d) Remain in the RELEASE COLLISION state.

4.3.3.4.3.2.3 When an A-RELEASE Indication primitive is validly invoked, and the CF is in the RELEASE COLLISION state, and it is the Responder CF, it shall:

- a) Construct a D-END Confirmation primitive, with the User Data parameter set equal to the value of the User Information parameter of the A-RELEASE Indication primitive, if present.

Note.— The D-END Confirmation is not issued to the DS-User until the orderly release procedure is complete, and an A-RELEASE Confirmation is received.

- b) Remain in the RELEASE COLLISION state.

4.3.3.4.4 A-RELEASE Confirmation primitive

4.3.3.4.4.1 When Invoked

4.3.3.4.4.1.1 The A-RELEASE Confirmation primitive may be invoked by the ACPM when the CF is in the RELEASE PENDING or RELEASE COLLISION state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.4.4.2 Action Upon Invocation

4.3.3.4.4.2.1 When an A-RELEASE Confirmation primitive is validly invoked, and the CF is in the RELEASE PENDING state, and the *Result* parameter has the abstract value “affirmative” the CF shall:

- a) Construct a D-END Confirmation primitive with the following parameter values.

Table 4.3-18.

D-END Confirmation parameter	Value
Result	“affirmative”
User Data	User Information parameter from the A-RELEASE Confirmation, if present

- b) Invoke the D-END Confirmation primitive.
- c) Issue a P-U-ABORT request primitive, with no parameters.

Note.— This will cause the release of the underlying transport connection.

- d) Enter the NULL state.

4.3.3.4.4.2.2 When an A-RELEASE Confirmation primitive is validly invoked, and the CF is in the RELEASE PENDING state, and the *Result* parameter has the abstract value “negative” the CF shall:

- a) Construct a D-END Confirmation primitive with the following parameter values.

Table 4.3-19.

D-END Confirmation parameter	Value
Result	“rejected”
User Data	User Information parameter from the A-RELEASE Confirmation, if present

- b) Invoke the D-END Confirmation primitive.
- c) Enter the DATA TRANSFER state.

4.3.3.4.4.2.3 When an A-RELEASE Confirmation primitive is validly invoked, and the *Result* parameter has the abstract value “affirmative”, and the CF is in the RELEASE COLLISION state, and it is the Initiator CF, it shall:

- a) Issue the D-END Confirmation primitive, which was previously formed in response to the reception of an A-RELEASE Indication primitive, to the DS-User.
- b) Issue a P-U-ABORT request primitive, with no parameters.

Note.— This will cause the release of the underlying transport connection.

- c) Enter the NULL state.

4.3.3.4.2.4 When an A-RELEASE Confirmation primitive is validly invoked, and the *Result* parameter has the abstract value “affirmative”, and the CF is in the RELEASE COLLISION state, and it is the Responder CF, it shall:

- a) Issue the D-END Confirmation primitive, which was previously formed in response to the reception of an A-RELEASE Indication primitive, to the DS-User.
- b) Construct an A-RELEASE Response primitive, with the *Result* parameter set to “affirmative”.
- c) Invoke the A-RELEASE Response.
- d) Remain in the RELEASE COLLISION state.

4.3.3.4.5 A-ABORT Indication primitive

4.3.3.4.5.1 When Invoked

4.3.3.4.5.1.1 Invocations of the A-ABORT Indication primitive by the ACPM shall be allowed when the CF is in any valid state, except the NULL state; if an invocation occurs when the CF is in the NULL state then an error has occurred (see 4.3.3.1.2.4).

4.3.3.4.5.2 Action Upon Invocation

4.3.3.4.5.2.1 When an A-ABORT Indication primitive is validly invoked, the CF shall:

- a) If the *Abort Source* parameter of the A-ABORT Indication is set to “ACSE service-user” and the Diagnostic parameter is set to “No reason given”, issue a D-ABORT Indication primitive to the DS-User, with the Originator parameter set to “User” and the User Data parameter set equal to the User Information parameter in the A-ABORT Indication, if present.
- b) If the *Abort Source* parameter of the A-ABORT Indication is set to “ACSE service-user” and the Diagnostic parameter is absent or is set to any value other than “No reason given”, then issue a D-ABORT Indication primitive to the DS-User, with the Originator parameter set to “Provider” and the User Data parameter set equal to the User Information parameter in the A-ABORT Indication, if present.
- c) If the *Abort Source* parameter of the A-ABORT Indication has the abstract value “ACSE service-provider”, then issue a D-ABORT Indication primitive to the DS-User, with the Originator parameter set to the abstract value “Provider”, and the User Data parameter set equal to the User Information parameter in the A-ABORT Indication, if present.
- d) Enter the NULL state.

4.3.3.4.6 A-P-ABORT Indication primitive

4.3.3.4.6.1 When Invoked

4.3.3.4.6.1.1 Invocations of the A-P-ABORT Indication primitive by the ACPM shall be allowed when the CF is in any valid state, except the NULL state; if an invocation occurs when the CF is in the NULL state then an error has occurred (see 4.3.3.1.2.4).

4.3.3.4.6.2 Action Upon Invocation

4.3.3.4.6.2.1 When an A-P-ABORT Indication primitive is validly invoked, the CF shall:

- a) issue a D-P-ABORT Indication primitive to the DS-User, and discard any Provider Reason parameter in the A-P-ABORT Indication; and
- b) Enter the NULL state.

4.3.3.5 Services used by ACSE

Note.— ACSE, edition 2 mandates the mapping of ACSE APDUs to the underlying presentation service provider. However, when the efficient encoding options of Session and Presentation protocols are used, the full Presentation service is no longer available. Therefore, invocations of presentation service primitives by ACSE are “intercepted” by the CF and re-mapped to the “actual” presentation service as appropriate.

4.3.3.5.1 P-CONNECT Request primitive

4.3.3.5.1.1 When Invoked

4.3.3.5.1.1.1 The P-CONNECT Request primitive may be validly invoked by the ACPM when the CF is in the ASSOCIATION PENDING state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.5.1.2 Action Upon Invocation

4.3.3.5.1.2.1 When a P-CONNECT Request primitive is validly invoked, the CF shall transparently invoke the equivalent presentation service primitive and remain in the same state.

4.3.3.5.2 P-CONNECT Response primitive

4.3.3.5.2.1 When Invoked

4.3.3.5.2.1.1 The P-CONNECT Response primitive may be validly invoked by the ACPM when the CF is in the ASSOCIATION PENDING state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.5.2.2 Action Upon Invocation

4.3.3.5.2.2.1 When the P-CONNECT Response primitive is validly invoked, the CF shall:

- a) transparently invoke the equivalent presentation service primitive.
- b) If the P-CONNECT Response Result parameter has the abstract value “acceptance” then enter the DATA TRANSFER state, otherwise enter the NULL state.

4.3.3.5.3 P-U-ABORT Request primitive

4.3.3.5.3.1 When Invoked

4.3.3.5.3.1.1 Invocations of the P-U-ABORT Request primitive by the ACPM shall be allowed when the CF is in any valid state.

4.3.3.5.3.2 Action Upon Invocation

4.3.3.5.3.2.1 When a P-U-ABORT Request primitive is validly invoked, the CF shall:

- a) If the P-U-ABORT request user data parameter is present, and the CF is in the DATA TRANSFER state:
 - 1) Encode the presentation user data as indicated in 4.3.2 with the P-U-ABORT user data parameter (an ABRT APDU) as the presentation data value and presentation context identifier value corresponding to “acse-apdu”.
 - 2) Invoke a P-DATA Request primitive with the resulting encoding as User Data.
- b) Otherwise, invoke a P-U-ABORT Request primitive with no parameters.

Note.— This will cause the underlying transport connection to be disconnected.

- c) Enter the NULL state.

4.3.3.5.4 P-RELEASE Request primitive

Note.— ACSE, edition 2 mandates the mapping of A-RELEASE APDUs (RLRQ and RLRE) to the P-RELEASE service. However, when the efficient encoding options of Session and Presentation protocols are used, the Session No-Orderly Release (NOR) functional unit is selected, and no mapping for the P-RELEASE service is available. In order to provide an orderly release service, the CF re-maps invocations of the P-RELEASE service at the lower service boundary of ACSE to invocations of the P-DATA service, with the release APDUs transferred as user information.

4.3.3.5.4.1 When Invoked

4.3.3.5.4.1.1 The P-RELEASE Request primitive may be validly invoked by the ACPM when the CF is in the RELEASE PENDING state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.5.4.2 Action Upon Invocation

4.3.3.5.4.2.1 When a P-RELEASE Request primitive is validly invoked, the CF shall:

- a) Encode the presentation user data as indicated in 4.3.2.6 with the P-RELEASE user data parameter (a RLRQ APDU) as the presentation data value and presentation context identifier corresponding to “acse-apdu”.
- b) Invoke a P-DATA Request primitive with the resulting encoding as User Data; and
- c) Remain in the RELEASE PENDING state.

4.3.3.5.5 P-RELEASE Response primitive

4.3.3.5.5.1 When Invoked

4.3.3.5.5.1.1 The P-RELEASE Response primitive may be validly invoked by the ACPM when the CF is in the RELEASE PENDING or RELEASE COLLISION state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.5.5.2 Action Upon Invocation

4.3.3.5.5.2.1 When a P-RELEASE Response primitive is validly invoked, and the CF is in the RELEASE PENDING state, and the *Result* parameter has the abstract value “affirmative” the CF shall:

- a) encode the presentation user data as indicated in 4.3.2 with the P-RELEASE user data parameter (a RLRE APDU) as the presentation data value and presentation context identifier corresponding to “acse-apdu”.
- b) Invoke a P-DATA Request primitive with the resulting encoding as User Data.
- c) Enter the NULL state.

Note.— The peer AEI is now expected to issue a P-U-ABORT request, which will cause the release of the underlying connection.

4.3.3.5.5.2.2 When a P-RELEASE Response primitive is validly invoked, and the CF is in the RELEASE PENDING state, and the *Result* parameter has the abstract value “negative” the CF shall:

- a) Encode the presentation user data as indicated in 4.3.2 with the P-RELEASE user data parameter (a RLRE APDU) as the presentation data value and presentation context identifier corresponding to “acse-apdu”;
- b) Invoke a P-DATA Request primitive with the resulting encoding as User Data; and
- c) Enter the DATA TRANSFER state.

4.3.3.5.5.2.3 When a P-RELEASE Response primitive is validly invoked, and the CF is in the RELEASE COLLISION state, and it is the Initiator CF, it shall:

- a) Encode the presentation user data as indicated in 4.3.2 with the P-RELEASE user data parameter (a RLRE APDU) as the presentation data value and presentation context identifier corresponding to “acse-apdu”;
- b) Invoke a P-DATA Request primitive with the resulting encoding as User Data; and
- c) Remain in the RELEASE COLLISION state.

4.3.3.5.5.2.4 When a P-RELEASE Response primitive is validly invoked, and the CF is in the RELEASE COLLISION state, and it is the Responder CF, it shall:

- a) Encode the presentation user data as indicated in 4.3.2 with the P-RELEASE user data parameter (a RLRE APDU) as the presentation data value and presentation context identifier corresponding to “acse-apdu”;
- b) Invoke a P-DATA Request primitive with the resulting encoding as User Data; and
- c) Enter the NULL state.

Note.— The peer AEI is now expected to issue a P-U-ABORT request, which will cause the release of the underlying connection.

4.3.3.5.5.2.5 **Recommendation.** - *After entering the NULL state, implementations should release the underlying connection (e.g., by issuing the P-U-ABORT request) if the communication peer does not cause the connection to be released as expected, after a period of time not less than twice the anticipated end-to-end transit time.*

4.3.3.6 Supporting Services delivered to the CF

Note 1.— The mapping by the CF of presentation service indication and confirmation primitives, which are invoked by the presentation service provider, is defined in the following paragraphs.

Note 2.— The following provisions describe the behaviour to be exhibited by the ATN-App AE when the supporting communications service exhibits behaviour modelled by the passing of indication or confirmation primitives to the application layer.

4.3.3.6.1 P-CONNECT Indication primitive

4.3.3.6.1.1 When Invoked

4.3.3.6.1.1.1 When the P-CONNECT Indication primitive is invoked by the supporting service, a new instance of communication shall be created, with its CF initially in the NULL state.

4.3.3.6.1.2 Action Upon Invocation

4.3.3.6.1.2.1 When a P-CONNECT Indication primitive is validly invoked, the CF shall:

- a) transparently invoke the equivalent presentation service primitive at the lower ACSE service boundary; and
- b) enter the ASSOCIATION PENDING state as the responder CF.

4.3.3.6.2 P-CONNECT Confirmation primitive

4.3.3.6.2.1 When Invoked

4.3.3.6.2.1.1 The P-CONNECT Confirmation primitive may be validly invoked by the supporting service when the CF is in the ASSOCIATION PENDING state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.6.2.2 Action Upon Invocation

4.3.3.6.2.2.1 When a P-CONNECT Confirmation primitive is validly invoked, the CF shall :

- a) transparently invoke the equivalent presentation service primitive at the lower ACSE service boundary; and
- b) Remain in the ASSOCIATION PENDING state.

4.3.3.6.3 P-U-ABORT Indication primitive

4.3.3.6.3.1 When Invoked

4.3.3.6.3.1.1 Invocations of the P-U-ABORT Indication primitive by the supporting service shall be allowed when the CF is in any valid state.

4.3.3.6.3.2 Action Upon Invocation

4.3.3.6.3.2.1 When a P-U-ABORT Indication primitive is validly invoked, the CF shall

- a) if the CF is in the NULL state, take no action;
- b) otherwise, transparently invoke the equivalent presentation service primitive at the lower ACSE service boundary, and remain in the same state.

4.3.3.6.4 P-P-ABORT Indication primitive

4.3.3.6.4.1 When Invoked

4.3.3.6.4.1.1 Invocations of the P-P-ABORT Indication primitive by the supporting service shall be allowed when the CF is in any valid state.

4.3.3.6.4.2 Action Upon Invocation

4.3.3.6.4.2.1 When a P-P-ABORT Indication primitive is validly invoked, the CF shall:

- a) if the CF is in the NULL state, then take no action;
- b) otherwise, transparently invoke the corresponding presentation service primitive at the lower ACSE service boundary; and
- c) remain in the same state.

4.3.3.6.5 P-DATA Indication primitive

4.3.3.6.5.1 When Invoked

4.3.3.6.5.1.1 Invocations of the P-DATA Indication primitive by the supporting service shall be allowed when the CF is in a valid state to receive the decoded APDU, as listed in 4.3.3.6.5.2; if an invocation occurs when the CF is not in a valid state then an error has occurred (see 4.3.3.1.2.4).

4.3.3.6.5.2 Action Upon Invocation

4.3.3.6.5.2.1 When a P-DATA Indication primitive is validly invoked, the CF shall decode the presentation user data as indicated in 4.3.2 to determine the destination ASE of the APDU, and extract the presentation data value.

Note.— The destination ASE is determined from the value of the presentation-context-identifier in the received User-data. Valid values are acse-apdu and user-ase-apdu, which correspond to destination ASEs of ACSE and ATN-App ASE, respectively.

4.3.3.6.5.2.2 ACSE APDU Received

4.3.3.6.5.2.2.1 If the destination ASE is ACSE then the CF shall determine the type of ACSE APDU present in the extracted presentation data value.

Note.—ACSE APDUs which may validly be received in a P-DATA indication are A-Release-Request (RLRQ), A-Release-Response (RLRE), and A-Abort (ABRT) APDUs.

4.3.3.6.5.2.2.2 If the received APDU is RLRQ, the CF shall:

- a) if in the DATA TRANSFER state, then invoke a P-RELEASE Indication primitive at the ACSE lower service boundary, with the RLRQ as User Data, and enter the RELEASE PENDING state as the Release Responder CF;
- b) if in the RELEASE PENDING state, and the CF is the Release Initiator, then invoke a P-RELEASE Indication primitive at the ACSE lower service boundary with the RLRQ as User Data, and enter the RELEASE COLLISION state;
- c) if in the NULL state, and this CF has previously issued an ABRT APDU and is awaiting disconnection by the peer, then take no action and remain in the NULL state;
- d) if none of the conditions a) to c) is satisfied, then take error handling action as described in 4.3.3.6.5.2.4.

4.3.3.6.5.2.2.3 If the received APDU is RLRE, the CF shall:

- a) if the Reason field in the RLRE has the value “not-finished”, and the CF is in the RELEASE PENDING state, then invoke a P-RELEASE Confirmation primitive at the ACSE lower service boundary, with the result parameter set to “negative”, and the RLRE as User Data; remain in the RELEASE PENDING state;
- b) if the Reason field in the RLRE has the value “normal”, and the CF is in the RELEASE PENDING or RELEASE COLLISION state, then invoke a P-RELEASE Confirmation primitive at the ACSE lower service boundary, with the result parameter set to “affirmative”, and the RLRE as User Data; remain in the same state;
- c) if the CF is in the NULL state, and this CF has previously issued an ABRT APDU and is awaiting disconnection by the peer, then take no action and remain in the NULL state;
- d) if none of the conditions a) to c) is satisfied, then take error handling action in 4.3.3.6.5.2.4.

4.3.3.6.5.2.2.4 If the received APDU is ABRT, the CF shall:

- a) if the CF is in the state DATA TRANSFER, or RELEASE PENDING, or RELEASE COLLISION, then invoke a P-U-ABORT Indication primitive at the ACSE lower service boundary, with the ABRT as User Data, and issue a P-U-ABORT request with no parameters to the underlying service; remain in the same state;

- b) if the CF is in the NULL state, then take no action unless this CF has previously issued an ABRT APDU and is awaiting disconnection by the peer, in which case issue a P-U-ABORT request to the underlying service; remain in the same state;
- c) if neither of the conditions a) and b) is satisfied, then take error handling action as described in 4.3.3.6.5.2.4.

4.3.3.6.5.2.3 ATN-App APDU Received

4.3.3.6.5.2.3.1 If the destination ASE is ATN-App ASE, then the CF shall:

- a) if the CF is in the DATA TRANSFER state, or the CF is in the RELEASE PENDING state and is the Release Initiator CF, then issue a D-DATA Indication primitive to the DS-User, with the received presentation data value as the user data parameter, and remain in the same state;
- b) if the CF is in the NULL state, and this CF has previously issued an ABRT APDU and is awaiting disconnection by the peer, then take no action and remain in the same state;
- c) if neither of the conditions a) and b) is satisfied, then take error handling action as described in 4.3.3.6.5.2.4.

4.3.3.6.5.2.4 Error conditions

4.3.3.6.5.2.4.1 If the destination ASE is invalid (i.e. neither ACSE nor ATN-App ASE), or an unrecognised APDU is received, or a valid APDU is received when the CF is not in the correct state (as defined in 4.3.3.6.5.2.2 and 4.3.3.6.5.2.3), then the CF shall:

- a) if not in the NULL state then issue a P-U-ABORT request with no parameters to the supporting service: and
- b) regardless of CF state behave as if a P-U-ABORT indication had been received.

4.4 SESSION LAYER REQUIREMENTS

Note.— The session layer requirements are described in many cases by means of completed protocol implementation conformance statement (PICS) proforma tables. In such tables, the “Ref.” column contains a reference to the relevant section in the session layer PICS proforma, ISO/IEC 8327-2.

4.4.1 Protocol versions implemented

4.4.1.1 Session protocol versions shall be supported as specified in Table 4.4-1.

Table 4.4-1. Session Protocol Versions Supported

Ref.	Version	ISO Status	ATN Support
S.A.3/1	Version 1	O.1	-
S.A.3/2	Version 2	O.1	M

O.1: The ISO PICS requires that the implementation of one, and only one, version of the protocol is described.

4.4.2 Session Functional units

4.4.2.1 Session functional units (S-FUs) shall be selected as specified in Table 4.4-2.

Table 4.4-2. Selection of Session functional units

Ref.	Functional unit	ISO Status	ATN Support
S.A.6.1/1	Kernel	M	M
S.A.6.1/2	Negotiated release	O	X
S.A.6.1/3	Half Duplex (HD)	O.2	X
S.A.6.1/4	Duplex	O.2	M
S.A.6.1/5	Expedited Data (EX)	O	X
S.A.6.1/6	Typed Data	O	X
S.A.6.1/7	Capability Data Exchange	C1	X
S.A.6.1/8	Minor Synchronize (SY)	O	X
S.A.6.1/9	Symmetric Synchronize (SS)	O	X
S.A.6.1/10	Data Separation	C2	X
S.A.6.1/11	Major Synchronize	O	X
S.A.6.1/12	Resynchronise	O	X
S.A.6.1/13	Exceptions	C3	X
S.A.6.1/14	Activity Management (ACT)	O	X
See note	No-orderly release (NOR)	O	M
See note	Special User-data	O	X

Note.— Functional units added by efficiency enhancement amendment.

O.2: The ISO standard requires at least one of the functional units Duplex and Half Duplex to be implemented.

C1:if [S-FU(ACT)] then O else N/A

C2:if [S-FU(SY) or S-FU(SS)] then O else N/A

C3:if [S-FU(HD)] then O else N/A

4.4.3 Protocol mechanisms

4.4.3.1 Session protocol mechanisms shall be supported as specified in Table 4.4-3.

Table 4.4-3. Session Protocol Mechanisms Supported

Ref.	Mechanism	ISO Status	ATN Support	Associated mnemonic
S.A.6.2/1	Use of transport expedited data (Extended control Quality of Service)	C4	X	S-EXP_T
S.A.6.2/2	Reuse of transport connection	O	O	S-REUSE_T
S.A.6.2/3	Basic concatenation	M	M (Note 2)	
S.A.6.2/4	Extended concatenation (sending)	O	X	
S.A.6.2/5	Extended concatenation (receiving)	O	X	S-XCONC_RCV
S.A.6.2/6	Segmenting (sending)	O	X	S-SEG_SDR
S.A.6.2/7	Segmenting (receiving)	O	X	S-SEG_RCV
S.A.6.2/8	Max. size of SS-user-data (S-CONNECT) > 512	O	O	S-MAXSIZE_512
S.A.6.2/9	Max. size of SS-user-data (S-CONNECT) > 10240	O	O	S-MAXSIZE_10240
S.A.6.2/10	Max. size of SS-user-data (S-ABORT) >9	O	X	S-MAXSIZE_9
See note 1	Null-encoding protocol option	-	M	
See note 1	Short-connect protocol option	-	M	
See note 1	Short-encoding protocol option	-	X	

Note 1.— Protocol options added by efficiency enhancement amendment.

Note 2.— Only Category 1 SPDUs are used for this ATN profile. By definition, these are never concatenated. Therefore, Basic concatenation is not applicable to this specification, but is supported to the extent necessary for compliance with the ISO PICS.

C4:if [S-FU(EX)] then M else O

4.4.3.2 The session protocol shall implement the efficiency enhancements in ISO/IEC 8327-1:1996/Amd. 1:1997 as specified, together with all approved amendments and defect report resolutions.

4.4.3.3 If the null encoding protocol option is offered by the initiating Session Protocol machine (SPM), the responding SPM shall select only the kernel, full-duplex and no-orderly release functional units for use on this connection.

4.4.3.4 Session Protocol Data Units (SPDUs) associated with the Short-connect protocol option (i.e. Short Connect (SCN), Short Accept (SAC), Short Accept Continue (SACC), Short Refuse (SRF) and Short Refuse Continue (SRFC)) shall be transferred as User-data on the Transport layer T-CONNECT primitives, where possible.

Note.— This is only possible if the complete SPDUs, including any User-data, meet any size restrictions of the T-CONNECT User-data.

4.4.4 Supported Roles

4.4.4.1 Session Connection

4.4.4.1.1 The roles for Session Connection shall be supported as specified in Table 4.4-4.

Table 4.4-4. Session Connection Roles Supported

Ref.	Role	ISO Status	ATN Support	Mnemonic
S.A.7.1.1.1/1	Connection initiator	O.3	M	S-CON_initiator
S.A.7.1.1.1/2	Connection responder	O.3	M	S-CON_responder

O.3: The ISO standard requires a conforming implementation to support at least one of these roles as required by the implementation.

4.4.4.1.2 When a connection establishment request is accepted, the SHORT-CPA PPDU in the SS-User-data of the positive S-CONNECT response/confirmation primitive shall map to the User-data parameter of a SAC SPDU.

4.4.4.1.3 When a connection establishment request is refused, the SHORT-CPR PPDU in the SS-User-data of the negative S-CONNECT response/confirmation primitive shall map to the User-data parameter of a SRF SPDU.

4.4.4.2 Orderly release

4.4.4.2.1 The roles for Session Orderly Release shall be supported as specified in Table 4.4-5.

Table 4.4-5. Session Orderly Release Roles Supported

Ref.	Role	ISO Status	ATN Support	Mnemonic
S.A.7.1.1.2/1	Requestor	O.4	N/A (See note)	S-REL_requestor
S.A.7.1.1.2/2	Acceptor	O.4	N/A (See note)	S-REL_acceptor

O.4: The ISO standard requires a conforming implementation to support at least one of these roles as part of the Kernel functional unit. However, selection of the No Orderly Release functional unit removes this requirement.

Note.— Not applicable, as the No Orderly Release functional unit is selected. For ATN applications, orderly release is provided by the CF as described in 4.3.

4.4.4.3 Normal Data Transfer

4.4.4.3.1 The roles for Session Normal Data Transfer shall be supported as specified in Table 4.4-6.

Table 4.4-6. Session Normal Data Transfer Roles Supported

Ref.	Role	ISO Status	ATN Support	Mnemonic
S.A.7.1.1.3/1	Requestor	O.5	M	S-DATA_requestor
S.A.7.1.1.3/2	Acceptor	O.5	M	S-DATA_acceptor

O.5: The ISO standard requires a conforming implementation to support at least one of these roles.

4.4.5 Supported SPDUs

Note.— This section specifies the SPDUs associated with the supported Session functional units. There are no additional SPDUs associated with the Duplex functional unit, or with the No Orderly Release functional unit.

4.4.5.1 Support for the SPDUs associated with the Kernel functional unit

4.4.5.1.1 Support for SPDUs shall be as specified in Table 4.4-7.

Table 4.4-7. Supported Session Protocol Data Units

Ref.	SPDU	Sender		Receiver		Mnemonics
		ISO Status	ATN Support	ISO Status	ATN Support	
S.A.7.1.2/1	Connect (CN)	C5	N/A (Note 4)	C6	N/A (Note 4)	
S.A.7.1.2/2	Overflow Accept (OA)	C7	N/A (Note 4)	C8	N/A (Note 4)	S-OA_SDR / S-OA_RCV
S.A.7.1.2/3	Connect Data Overflow (CDO)	C9	N/A (Note 4)	C10	N/A (Note 4)	S-CDO_SDR / S-CDO_RCV
S.A.7.1.2/4	Accept (AC)	C6	N/A (Note 4)	C5	N/A (Note 4)	
S.A.7.1.2/5	Refuse (RF)	C6	N/A (Note 4)	C5	N/A (Note 4)	
S.A.7.1.2/6	Finish (FN)	C11	N/A (Note 2)	C12	N/A (Note 2)	
S.A.7.1.2/7	Disconnect (DN)	C12	N/A (Note 2)	C11	N/A (Note 2)	
S.A.7.1.2/8	Abort	M	N/A (Note 3)	M	N/A (Note 3)	
S.A.7.1.2/9	Abort Accept (AA)	O	N/A (Note 3)	M	N/A (Note 3)	
S.A.7.1.2/10	Data Transfer (DT)	C13	N/A (Note 3)	C14	N/A (Note 3)	
S.A.7.1.2/11	Prepare (PR)	C15	X	C15	X	S-PR_SDR / S-PR_RCV
See note 1	Short Connect (SCN)	C17	M	C17	M	
See note 1	Short Accept (SAC)	C17	M	C17	M	
See note 1	Short Refuse (SRF)	C17	M	C17	M	

		Sender		Receiver		
Ref.	SPDU	ISO Status	ATN Support	ISO Status	ATN Support	Mnemonics
See note 1	Null (NL)	C18	M	C18	M	
See note 1	Short Connect Continue (SCNC)	C16	N/A	C16	N/A	
See note 1	Short Accept Continue (SACC)	C17	M	C17	M	
See note 1	Short Refuse Continue (SRFC)	C17	M	C17	M	
See note 1	Short Finish (SFN)	C16	N/A	C16	N/A	
See note 1	Short Disconnect (SDN)	C16	N/A	C16	N/A	
See note 1	Short Data Transfer (SDT)	C16	N/A	C16	N/A	
See note 1	Short Abort (SAB)	C16	N/A	C16	N/A	

Note 1.— PDUs defined in efficiency enhancement amendment.

Note 2.— Not applicable, as the no-orderly-release functional unit is selected.

Note 3.— Not applicable, as the null-encoding protocol option is selected.

Note 4.— Not applicable, as the short-connect protocol option is selected.

C5: if [S-CON_initiator] then M else N/A

C6: if [S-CON_responder] then M else N/A

C7: if [S-V1 or (NOT S-CON_responder)] then N/A else if [S-MAXSIZE_10240] then M else O

C8: if [NOT S-V1 and S-CON_responder and S-MAXSIZE_10240] then M else N/A

C9: if [S-V1 or (NOT S-CON_initiator)] then N/A else if [S-MAXSIZE_10240] then M else O

C10: if [NOT S-V1 and S-CON_initiator and S-MAXSIZE_10240] then M else N/A

C11: if [S-REL_requestor] then M else N/A

C12: if [S-REL_acceptor] then M else N/A

C13: if [S-DATA_requestor] then M else N/A

C14: if [S-DATA_acceptor] then M else N/A

C15: if [NOT S-V1 and S-MAXSIZE_9 and S-EXP_T] then M else N/A

C16: used only if the short-encoding protocol option is selected.

C17: used if short-encoding or null-encoding is used.

C18: used only if the null-encoding protocol option is supported.

4.4.5.1.2 SCN, SAC, SRF, SACC and SRFC SPDUs shall be encoded such that the parameter bit of the SI&P octet is set to the value 0, indicating that all following octets are User-information (i.e. no SPDU parameters are present).

Note.— This is a requirement of the null-encoding protocol option.

4.4.5.2 Support for the SPDUs associated with Token Exchange

4.4.5.2.1 Support for Session protocol data units associated with Token exchange shall be as specified in Table 4.4-8.

Table 4.4-8. SPDUs associated with Token Exchange

Ref.	SPDU	Sender		Receiver	
		ISO Status	ATN Support	ISO Status	ATN Support
S.A.7.1.3/1	Give Tokens (GT)	M	- (See note 2)	M	- (See note 2)
S.A.7.1.3/2	Please Tokens (PT)	M	- (See note 2)	M	- (See note 2)

Note 1.— The ISO PICS states that these two SPDUs are used for Token Exchange, but they are also used as category 0 SPDUs in basic concatenation. Therefore, their implementation is mandatory even if no token is supported (reference ISO/IEC 8327-1 clauses 7.16 and 7.17). However, if the null-encoding protocol option is selected, their encoding will be null, i.e. not present.

Note 2.— Not applicable, as the null-encoding protocol option is selected.

4.4.6 Use of null-encoding and short-connect protocol options

4.4.6.1 The null-encoding and short-connect session protocol options shall be selected for use, with the requirements as specified in Table 4.4-9.

Table 4.4-9. Use of the null encoding and short-connect Session protocol options

Ref.	Requirement	Base Status	ATN Requirement
a	The calling and called session selectors are null	C1	M
b	The session-requirements parameter in the S-CONNECT service includes the kernel, full-duplex and no-orderly-release functional units only.	C1	M

C1: The SPMs may use the short-connect protocol option to establish a session connection using the null-encoding option. The null-encoding protocol option is available for use on an established connection only if the conditions a and b in Table 4.4-9 are both true.

4.4.7 Mapping to the ATN Internet Transport Service

4.4.7.1 The use of the connection-oriented transport service provided by the ATN Internet shall be as specified in Clause 6 of ISO/IEC 8327-1, except as stated in this section.

4.4.7.2 The called and calling Transport Service Access Point (TSAP) address shall be provided to the TS-Provider on a per Transport Connection basis, using the called and calling Presentation Service Access Point (PSAP) addresses as provided to ACSE in the A-ASSOCIATE request, with null presentation and session selectors.

4.4.7.3 The TS-user shall indicate in all T-CONNECT requests that the transport expedited flow is not required.

4.4.7.4 Information on the use or non-use of the transport checksum shall be conveyed between the TS-User and TS-Provider via the “residual error rate” component of the T-CONNECT quality of service parameter.

Note 1.— 5.5.1.2 requires that the TS-user specifies the required residual error rate to determine whether or not the transport checksum is required. In the ATN, the Quality of Service provided to applications is maintained using capacity planning techniques that are outside of the scope of this specification. Network administrators are responsible for designing and implementing a network that will meet the QOS requirements of the CNS/ATM applications that use it.

Note 2.— If the TS-User requests the use of checksum (RER = “low”) in the request primitive, the peer can only accept the use of checksum for this Transport Connection. If the TS-User proposes non-use of checksum (RER = “high”) in the request primitive, the peer can either accept the non-use of checksum or force the use of checksum for this Transport Connection.

4.4.7.5 The use or non-use of the transport checksum shall be negotiated by the TS-provider on a per Transport Connection basis, based on TS-User requests in the T-CONNECT request and response primitives, as follows:

- a) If the required residual error rate in the T-CONNECT request has the abstract value “low”, then the TS-provider uses best endeavours to obtain the lowest available residual error rate, including the use of the transport checksum in all Transport Protocol Data Units (TPDUs). The residual error rate in the T-CONNECT indication is set to the abstract value “low”, and the responder can only accept this value in the T-CONNECT response.
- b) If the required residual error rate in the T-CONNECT request has the abstract value “high”, then the TS-provider proposes non-use of the transport checksum. The residual error rate in the T-CONNECT indication is set to the abstract value “high”, and the responder can either accept this value, or request “low” in the T-CONNECT response. In the former case, transport checksum is not used, and in the latter case the TS-provider uses the transport checksum for all TPDUs.

4.4.7.6 The Application Service Priority shall be provided to the TS-Provider on a per Transport Connection basis, by implementation-specific means, and using the values for “Transport Layer Priority” specified in Table 1-2.

Note.— Although transport priority and network priority are semantically independent of each other, it is required (in 5.5.1.2), that the TS-user specifies the Application Service Priority, which in turn is mapped into the resulting CLNP PDUs according to Table 1-2, which defines the fixed relationship between transport priority and the network priority.

4.4.7.7 The ATN Security Label shall be provided to the TS-Provider on a per Transport Connection basis.

4.4.7.8 The ATN Security Label value shall be encoded according to 5.6.2.2.2.2 b), and passed between TS-User and TS-Provider by implementation-specific means.

4.4.7.9 The QOS parameter “Routing Class” shall be conveyed as the Security Tag field of the security tag set for Traffic Type and Associated Routing Policies within the ATN Security Label.

Note 1.— 5.2.7.3.1 states: “The mechanism by which the [transport] connection initiator provides the appropriate ATN Security Label is a local matter. For example, it may be identified by an extension to the transport service interface, be implicit in the choice of a given TSAP, or be identified using a Systems Management function.”

Note 2.— 5.5.1.2 requires the TS-User to provide the ATN Security Label as specified in Figure 5.6-1 and 5.6.2.2.2.2 b). The encoding of the ATN Security Label is summarised below. The D-START QOS parameter “Routing Class” maps to the field labelled “Traffic Type & category”.

ATN Security Label field	Value (Hex)	Length (Octets)
Security Registration ID Length	6	1
Security Registration ID = OID {1.3.27.0.0}	06, 04, 2B, 1B, 00, 00	6
Security Information Length	4	1
Security information:		
Tag Set Name Length	1	1
Tag Set Name = “Traffic Type & Associated Routing Policies”	0F	1
Tag Set Length	1	1
Security Tag Value = Traffic Type & category (from Table 5.6-1)	01 (for example)	1
Total:		12 Octets

4.4.7.10 No Transport Service quality of service parameters other than those specified in the preceding subsections shall be specified when establishing a transport connection.

4.5 PRESENTATION LAYER REQUIREMENTS

Note.— The presentation layer requirements are described in many cases by means of completed protocol implementation conformance statement (PICS) proforma tables. In such tables, the “Ref.” column contains a reference to the relevant section in the presentation layer PICS proforma ISO/IEC 8823-2.

4.5.1 Protocol mechanisms

4.5.1.1 The Presentation protocol mechanisms supported shall be as specified in Table 4.5-1.

Table 4.5-1. Presentation Protocol Mechanisms Supported

Ref.	Protocol Mechanism	ISO Status	ATN Support	Mnemonic
P.A.6.1/2	Normal mode	O.1	M	
P.A.6.1/1	X.410-1984 mode	O.1	X	
See note	Nominated context	O	N/A	
See note	Short encoding	O	N/A	
See note	Packed encoding rules	O	N/A	
See note	Short-connect	O	M	
See note	Null-encoding	O	M	

Note.— Optional protocol mechanisms defined in efficiency enhancement amendment.

O.1: The ISO standard requires that either Normal mode or X.410 (1984) mode or both be supported.

4.5.1.2 The presentation protocol shall implement the efficiency enhancements in ISO/IEC 8823-1: 1994/Amd. 1: 1997 as specified, together with all approved amendments and defect report resolutions.

4.5.2 Use of null-encoding and short-connect protocol options

4.5.2.1 The null-encoding and short-connect presentation protocol options shall be selected for use, with the requirements as specified in Table 4.5-2.

Table 4.5-2. Use of the null encoding and short-connect Presentation protocol options

Ref.	Requirement	Base Status	ATN Requirement
a	The presentation context definition list contains precisely one item in which the abstract syntax is known to the responding Presentation Protocol Machine (PPM) by bilateral agreement.	C1	N/A
b	The presentation context definition list is empty and the default context is known by bilateral agreement	C1	M
c	The presentation context definition list is empty and the abstract syntax of the default context is known to the responding PPM by bilateral agreement and is specified in ASN.1	C1	M
d	The calling and called presentation selectors are null	C2	M
e	The presentation-requirements parameter in the P-CONNECT service includes the kernel functional unit only.	C2	M

C1: The null-encoding protocol option is available for use on an established connection only if at least one of the conditions a, b and c in Table 4.5-2 is true.

C2: The short-connect protocol option is used only in connection establishment to establish a connection on which the null-encoding option will be used; it can only be used if both of the conditions d and e in Table 4.5-2 is true.

4.5.3 Mapping of Presentation Primitives to the Null Encoding option

Note.— When the null-encoding presentation protocol option is selected, no presentation protocol control information is present once the connection has been established. Thus, no presentation PDUs are supported. The presentation connection is only terminated by the termination of the supporting session and transport connections.

4.5.3.1 The user of the presentation service shall not issue any presentation primitives other than P-CONNECT request, P-CONNECT response, P-DATA request and P-U-ABORT request.

4.5.3.2 When it is required to release the presentation connection, the presentation service user shall issue a P-U-ABORT request.

4.5.3.3 Any user data in a P-U-ABORT request shall be ignored by the presentation service provider.

4.5.4 Functional units

4.5.4.1 The Presentation functional units selected shall be as specified in Table 4.5-3.

Table 4.5-3. Selection of Presentation functional units

Ref.	Presentation functional unit	ISO Status	ATN Support	Mnemonic
P.A.6.2/1	Kernel	M	M	
P.A.6.2/2	Presentation Context Management	O	X	P-FU(CM)
P.A.6.2/3	Presentation Context Restoration	C1	X	P-FU(CR)

C1: if Presentation Context Management (2) is supported then O else N/A

4.5.4.2 The Presentation pass-through functional units selected shall be as specified in Table 4.5-4.

Table 4.5-4. Selection of Presentation pass-through functional units

Ref.	Pass-through to Session functional units	ISO Status	ATN Support	Mnemonic
P.A.6.2/4	Negotiated release	O	X	S-FU(NR)
P.A.6.2/5	Half Duplex	O.2	X	S-FU(HD)
P.A.6.2/6	Duplex	O.2	M	S-FU(FD)
P.A.6.2/7	Expedited Data	O	X	S-FU(EX)
P.A.6.2/8	Typed Data	O	X	S-FU(TD)
P.A.6.2/9	Capability Data Exchange	C1	X	S-FU(CD)
P.A.6.2/10	Minor Synchronize	O	X	S-FU(SY)
P.A.6.2/11	Symmetric Synchronize	O	X	S-FU(SS)
P.A.6.2/12	Data Separation	O	X	S-FU(DS)
P.A.6.2/13	Major Synchronize	O	X	S-FU(MA)

Ref.	Pass-through to Session functional units	ISO Status	ATN Support	Mnemonic
P.A.6.2/14	Resynchronise	O	X	S-FU(RESYNC)
P.A.6.2/15	Exceptions	C2	X	S-FU(EXCEP)
P.A.6.2/16	Activity Management	O	X	S-FU(ACT)
See note	No-orderly release (NOR)	-	M	S-FU(NOR)

Note.— The NOR Session functional unit is defined in the ISO Session service efficiency enhancement amendment.

O.2: The ISO standard requires that pass-through for at least one of the Session functional units Duplex and Half Duplex be supported.

C1: if [S-FU(ACT)] then O else N/A

C2: if [S-FU(HD)] then O else N/A

4.5.5 Elements of procedure

4.5.5.1 Supported roles

4.5.5.1.1 Presentation Connection

4.5.5.1.1.1 The supported roles for establishing Presentation connections shall be as specified in Table 4.5-5.

Table 4.5-5. Presentation Connection roles

Ref.	Role	ISO Status	ATN Support	Mnemonic
P.A.7.1.1.1/1	Initiator	O.3	M	P-CON_initiator
P.A.7.1.1.1/2	Responder	O.3	M	P-CON_responder

O.3: The ISO standard requires a conforming implementation to support at least one of these roles.

4.5.5.1.1.2 When a connection establishment request is accepted, the AARE (accepted) in the User-data of the positive P-CONNECT response/confirmation primitive shall map to the User-data parameter of a SHORT-CPA PPDU.

4.5.5.1.1.3 When a connection establishment request is refused, the AARE (rejected) in the User-data of the negative P-CONNECT response/confirmation primitive shall map to the User-data parameter of a SHORT-CPR PPDU.

4.5.5.1.2 Orderly release

4.5.5.1.2.1 The supported roles for the orderly release of Presentation connections shall be as specified in Table 4.5-6.

Table 4.5-6. Presentation Connection orderly release roles

Ref.	Role	ISO Status	ATN Support	Mnemonic
P.A.7.1.1.3/1	Requestor	O	N/A	P-REL_requestor
P.A.7.1.1.3/2	Acceptor	O	N/A	P-REL_acceptor

4.5.5.1.3 Normal Data

4.5.5.1.3.1 The supported roles for Normal Data shall be as specified in Table 4.5-7.

Table 4.5-7. Presentation Normal Data roles

Ref.	Role	ISO Status	ATN Support	Mnemonic
P.A.7.1.1.2/1	Requestor	O	M	P-DATA_requestor
P.A.7.1.1.2/2	Acceptor	O	M	P-DATA_acceptor

4.5.6 Supported Presentation Protocol Data Units (PPDUs)

Note.— This section specifies the PPDUs associated with the supported Presentation functional units. There are no additional PPDUs or additional pass-through functionality associated with the supported Session functional units.

4.5.6.1 Supported PPDUs associated with the Kernel services

The Presentation Protocol Data Units supported shall be as specified in Table 4.5-8.

Table 4.5-8. Supported Presentation Protocol Data Units

		Sender		Receiver		
Ref.	PPDU	ISO Status	ATN Support	ISO Status	ATN Support	Mnemonics
P.A.7.1.2/1	Connect presentation (CP)	C3	N/A (Note 2)	C4	N/A (Note 2)	
P.A.7.1.2/2	Connect presentation accept (CPA)	C4	N/A (Note 2)	C3	N/A (Note 2)	S-OA_SDR / S-OA_RCV
P.A.7.1.2/3	Connect presentation reject (CPR)	C4	N/A (Note 2)	C3	N/A (Note 2)	S-CDO_SDR / S-CDO_RCV
P.A.7.1.2/4	Abnormal release provider (ARP)	M	N/A (Note 2)	M	N/A (Note 2)	
P.A.7.1.2/5	Abnormal release user (ARU)	O	N/A (Note 2)	M	N/A (Note 2)	
P.A.7.1.2/6	Presentation Data (TD)	C5	N/A (Note 2)	C6	N/A (Note 2)	
Note 1	Short Connect (SHORT-CP)	O	M	O	M	
Note 1	Short Connect Accept (SHORT-CPA)	O	M	O	M	
Note 1	Short Connect Reject (SHORT-CPR)	O	M	O	M	

Note 1.— PDUs defined in efficiency enhancement amendment.

Note 2.— PPDUs not applicable, as the short-connect and null-encoding protocol options are selected.

C3: if [P-CON_initiator] then M else N/A

C4: if [P-CON_responder] then M else N/A

C5: if [P-DATA_requestor] then M else N/A

C6: if [P-DATA_acceptor] then M else N/A

4.5.6.2 Structure and encoding of PPDUs

4.5.6.2.1 The SHORT-CP, SHORT-CPA and SHORT-CPR PPDUs shall have the encoding-choice bit-field set to “unaligned PER”.

4.6 ACSE SPECIFICATION

Note.— The ACSE requirements are described in many cases by means of completed protocol implementation conformance statement (PICS) proforma tables. In such tables, the “Ref.” column contains a reference to the relevant section in the ACSE PICS proforma ISO/IEC 8650-2.

4.6.1 Protocol details

4.6.1.1 The specification of the ACSE protocol supported shall be as defined in Table 4.6-1.

Table 4.6-1. Identification of ACSE Protocol Specification

Identification of Protocol Specification	ATN Support	Comments
ISO/IEC 8650-1:1995	M	See note

Note.— This is the second edition of the ACSE protocol specification.

4.6.2 Protocol versions

4.6.2.1 The version of the ACSE protocol supported shall be as specified in Table 4.6-2.

Table 4.6-2. Identification of ACSE Protocol version

Ref.		ISO Status	ATN Support	Mnemonic
A.A.4.2/1	Version 1	O.1	M	A-V1
A.A.4.2/2	Version 2	O.1		

O.1: The ISO PICS requires support of the implementation of only one version of the protocol to be described.

4.6.3 Supported roles

4.6.3.1 Association establishment

4.6.3.1.1 The supported roles for Association Establishment shall be as specified in Table 4.6-3.

Table 4.6-3. ACSE Roles for Association Establishment

Ref.	Capability	ISO Status	ATN Support	Mnemonic
A.A.6.1/1	Association initiator	O.2	See text	A-CON_initiator
A.A.6.1/2	Association responder	O.2	See text	A-CON_responder

O.2: The ISO standard requires a conforming implementation to support at least one of the roles.

4.6.3.1.2 Either one or both of the ACSE roles “Association initiator” or “Association responder” shall be supported.

4.6.3.2 Normal Release procedure

4.6.3.2.1 The supported roles for the Normal Release procedure shall be as specified in Table 4.6-4.

Table 4.6-4. ACSE Roles for Normal Release

Ref.	Role	ISO Status	ATN Support	Mnemonic
A.A.6.2/1	Initiator	O	See text	A-REL_requestor
A.A.6.2/2	Responder	O	See text	A-REL_acceptor

4.6.3.2.2 Either one or both of the ACSE Normal Release roles “Initiator” or “Responder” shall be supported.

4.6.3.2.3 The ACSE Release Responder shall be allowed to give a negative response, despite the fact that the session Negotiated Release functional unit is not selected for the association.

Note.— The above provision waives the ISO/IEC 8649 requirement that the Responder may give a negative response only if session Negotiated Release is selected. This is possible because, for ATN, the ACSE release PDUs do not map directly to the Presentation release service; they are re-mapped by the CF to P-DATA.

4.6.3.3 Abnormal Release procedure

4.6.3.3.1 The supported roles for the Abnormal Release procedure shall be as specified in Table 4.6-5.

Table 4.6-5. ACSE Roles for Abnormal Release

Ref.	Role	ISO Status	ATN Support	Mnemonic
A.A.6.3/1	Initiator	M	M	
A.A.6.3/2	Responder	M	M	

4.6.4 Protocol mechanisms

4.6.4.1 The ACSE protocol mechanisms supported shall be as specified in Table 4.6-6.

Table 4.6-6. ACSE Protocol Mechanisms Supported

Ref.	Protocol Mechanism	ISO Status	ATN Support	Mnemonic
A.A.7/1	Normal mode	O.4	M	
A.A.7/2	X.410-1984 mode	O.4	X	
A.A.7/2	Rules for extensibility	M	M	
A.A.7/4	Supports operation of Session version 2	O	M	S-O-SESS-V2

O.4: The ISO standard requires either Normal mode or X.410-1984 mode or both to be supported.

4.6.4.2 Extensibility and Encoding

4.6.4.2.1 For the purposes of this specification, the abstract syntax module defined in clause 9 of the ACSE protocol specification shall be augmented with the ASN.1 extensibility notation, as specified in ISO/IEC 8650-1: 1996/Amd. 1: 1997

4.6.4.2.2 The system shall support that encoding which results from applying the ASN.1 packed encoding rules (basic, unaligned variant), as specified in ISO/IEC 8825-2, to the abstract syntax module specified in 4.6.4.2.1.

4.6.4.2.3 Packed encoding (basic, unaligned) shall be used for encoding all ACSE Protocol Control Information (PCI) for interchange.

4.6.5 ACSE Functional units

4.6.5.1 The ACSE functional units selected shall be as specified in Table 4.6-7.

Table 4.6-7. Selection of ACSE Functional Units

Ref.	Role	ISO Status	ATN Support	Mnemonic
A.A.8/1	Normal mode	M	M	
A.A.8/2	Authentication	O	C1	A-FU(AU)

C1: If the Dialogue Service user requires the use of the Security Requirements parameter of the D-START primitives, then M, else O.

4.6.6 Supported APDUs

4.6.6.1 The ACSE Protocol data units supported shall be as specified in Table 4.6-8.

Table 4.6-8 Supported ACSE Protocol Data Units

Ref.	APDU	Sender		Receiver		Comment
		ISO Status	ATN Support	ISO Status	ATN Support	
A.A.9/ 1	AARQ	C1	M	C2	M	
A.A.9/ 2	AARE	C2	M	C1	M	
A.A.9/ 3	RLRQ	C3	M	C4	M	
A.A.9/ 4	RLRE	C4	M	C3	M	
A.A.9/ 5	ABRT	C5	M	C5	M	

C1: if [A-CON_initiator] then M else N/A

C2: if [A-CON_responder] then M else N/A

C3: if [A-REL_requestor] then M else N/A

C4: if [A-REL_acceptor] then M else N/A

C5: if [S-O-SESS-V2] then M else N/A

4.6.6.2 Supported APDU parameters

4.6.6.2.1 A-Associate-request (AARQ)

4.6.6.2.1.1 The parameters in the AARQ APDU shall be supported as specified in Table 4.6-9.

Table 4.6-9. Supported AARQ Parameters

		Sender		Receiver	
Ref.	Parameter	ISO Status	ATN Support	ISO Status	ATN Support
A.A.10.1/1	Protocol Version	C6	X	C2	M
A.A.10.1/2	Application Context Name	C1	M	C2	M
A.A.10.1/3	Calling AP title	C6	M	C2	M
A.A.10.1/4	Calling AE qualifier	C6	M	C2	M
A.A.10.1/5	Calling AP invocation-identifier	C6	X	C2	M
A.A.10.1/6	Calling AE invocation-identifier	C6	X	C2	M

		Sender		Receiver	
Ref.	Parameter	ISO Status	ATN Support	ISO Status	ATN Support
A.A.10.1/7	Called AP title	C6	X	C2	M
A.A.10.1/8	Called AE qualifier	C6	X	C2	M
A.A.10.1/9	Called AP invocation-identifier	C6	X	C2	See text
A.A.10.1/10	Called AE invocation-identifier	C6	X	C2	See text
A.A.10.1/11	ACSE-requirements	C8	See text	C9	M
A.A.10.1/12	Authentication-mechanism name	C8	X	C9	N/A
A.A.10.1/13	Authentication-value	C8	See text	C9	M
A.A.10.1/14	Implementation information	C6	X	C7	O
A.A.10.1/15	User information	C6	M	C7	M

C1: if [A-CON_initiator] then M else N/A

C2: if [A-CON_responder] then M else N/A

C6: if [A-CON_initiator] then O else N/A

C7: if [A-CON_responder] then O else N/A

C8: if [A-CON_initiator and A-FU(AU)] then M else N/A

C9: if [A-CON_responder and A-FU(AU)] then M else N/A

4.6.6.2.1.2 The AARQ parameters “ACSE-Requirements” and “Authentication-value” shall be supported, for sending, only if the connection initiator role (A-CON_initiator) and the Authentication functional unit (A-FU(AU)) are supported.

Note.— The ATN specification is non-conformant to the ISO PICS proforma, in that the “Authentication-mechanism-name” parameter is not supported for sending.

4.6.6.2.1.3 The AARQ parameters “ACSE-Requirements” and “Authentication-value” shall be supported for receiving if the connection responder role (A-CON_responder) is supported, but are ignored if the Authentication functional unit (A-FU(AU)) is not supported by the responder.

Note.— The ATN specification is non-conformant to the ISO PICS proforma, in that the “Authentication-mechanism-name” parameter is “N/A” for receiving, if the Authentication functional unit is selected, and “ACSE-requirements” and “Authentication-value” are “M” for receiving, even if the Authentication functional unit is not supported.

4.6.6.2.1.4 The AARQ parameters “Called AP invocation-identifier” and “Called AE invocation-identifier” shall be supported, for receiving, if the Association Responder role is supported.

4.6.6.2.2 A-Associate-response (AARE)

4.6.6.2.2.1 The parameters in the AARE APDU shall be supported as specified in Table 4.6-10.

Table 4.6-10. Supported AARE Parameters

		Sender		Receiver	
Ref.	Parameter	ISO Status	ATN Support	ISO Status	ATN Support
A.A.10.2/1	Protocol Version	C7	X	C1	M
A.A.10.2/2	Application Context Name	C2	M	C1	M
A.A.10.2/3	Responding AP title	C7	X	C1	M
A.A.10.2/4	Responding AE qualifier	C7	X	C1	M
A.A.10.2/5	Responding AP invocation-identifier	C7	X	C1	M
A.A.10.2/6	Responding AE invocation-identifier	C7	X	C1	M
A.A.10.2/7	Result	C2	M	C1	M
A.A.10.2/8	Result source - diagnostic	C10	M	C11	M
A.A.10.2/9	ACSE-requirements	C9	See text	C8	See text
A.A.10.2/10	Authentication-mechanism name	C9	X	C8	N/A
A.A.10.2/11	Authentication-value	C9	See text	C8	See text
A.A.10.2/12	Implementation information	C7	X	C6	O
A.A.10.2/13	User information	C7	M	C6	M

C1: if [A-CON_initiator] then M else N/A

C2: if [A-CON_responder] then M else N/A

C6: if [A-CON_initiator] then O else N/A

C7: if [A-CON_responder] then O else N/A

C8: if [A-CON_initiator and A-FU(AU)] then M else N/A

C9: if [A-CON_responder and A-FU(AU)] then M else N/A

C10: if [A-CON_responder] then (if [A-FU(AU)] then M (with a value range of 0 to 14) else M (with a value range of 0 to 10) else N/A

C11: if [A-CON_initiator] then (if [A-FU(AU)] then M (with a value range of 0 to 14) else M (with a value range of 0 to 10) else N/A

4.6.6.2.2.2 The AARE parameters “ACSE-Requirements”, Authentication-mechanism-name” and “Authentication-value” shall be supported, for sending, only if the connection responder role (A-CON_responder) and the Authentication functional unit (A-FU(AU)) are supported.

4.6.6.2.2.3 The AARE parameters “ACSE-Requirements” and “Authentication-value” shall be supported, for receiving, only if the connection initiator role (A-CON_initiator) and the Authentication functional unit (A-FU(AU)) are supported.

4.6.6.2.3 A-Release-request (RLRQ)

4.6.6.2.3.1 The parameters in the RLRQ APDU shall be supported as specified in Table 4.6-11.

Table 4.6-11. Supported RLRQ Parameters

		Sender		Receiver	
Ref.	Parameter	ISO Status	ATN Support	ISO Status	ATN Support
A.A.10.3/1	Reason	C12	M	C4	M
A.A.10.3/2	User information	C12	M	C4	M

C4: if [A-REL_acceptor] then M else N/A

C12: if [A-REL_requestor] then O else N/A

4.6.6.2.4 A-Release-response (RLRE)

4.6.6.2.4.1 The parameters in the RLRE APDU shall be supported as specified in Table 4.6-12.

Table 4.6-12 Supported RLRE Parameters

		Sender		Receiver	
Ref.	Parameter	ISO Status	ATN Support	ISO Status	ATN Support
A.A.10.4/1	Reason	C13	M	C3	M
A.A.10.4/2	User information	C13	M	C3	M

C3: if [A-REL_requestor] then M else N/A

C13: if [A-REL_acceptor] then O else N/A

4.6.6.2.5 A-Abort (ABRT)

4.6.6.2.5.1 The parameters in the ABRT APDU shall be supported as specified in Table 4.6-13.

Table 4.6-13. Supported ABRT Parameters

		Sender		Receiver	
Ref.	Parameter	ISO Status	ATN Support	ISO Status	ATN Support
A.A.10.5/1	Abort source	M	M	M	M
A.A.10.5/2	Diagnostic	C14	M	C14	M
A.A.10.5/3	User information	O	M	M	M

C14: if [A-FU(AU)] then M else N/A

4.6.6.3 Supported parameter forms

4.6.6.3.1 AE title name form

4.6.6.3.1.1 The Application Entity Title parameter shall be supported in the forms specified in Table 4.6-14.

Table 4.6-14. AE Title Name Form

		Sender		Receiver	
Ref.	Syntax form	ISO Status	ATN Support	ISO Status	ATN Support
A.A.11.1/1	Form 1 (Directory name)	O.5	X	M	O
A.A.11.1/2	Form 2 (Object identifier and integer)	O.5	M	M	M

O.5: The ISO standard requires a conforming implementation to support at least one of the forms.

4.6.6.3.2 Authentication value form

4.6.6.3.2.1 The Authentication value parameter shall be supported in the forms specified in Table 4.6-15.

Table 4.6-15. Authentication Value Form

Prerequisite: A-FU(AU)

		Sender		Receiver	
Ref.	Authentication value form	ISO Status	ATN Support	ISO Status	ATN Support
A.A.11.2/1	GraphicString	O.6	See text	C14	M
A.A.11.2/2	BIT STRING	O.6	See text	C14	M
A.A.11.3/3	EXTERNAL	O.6	See text	C14	M

		Sender		Receiver	
Ref.	Authentication value form	ISO Status	ATN Support	ISO Status	ATN Support
A.A.11.4/4	Other	O.6	X	C14	N/A

O.6: The ISO standard requires a conforming implementation to support at least one of the forms.

C14: if [A-FU(AU)] then M else N/A

4.6.6.3.2.2 If the authentication functional unit is supported, at least one of the Authentication-value forms listed in Table 4.6-15 shall be implemented for sending.

4.6.6.3.3 User information form

4.6.6.3.3.1 User information reference

4.6.6.3.3.1.1 The User information parameter shall use the forms of reference specified in Table 4.6-16.

Table 4.6-16. User information reference

		Sender		Receiver	
Ref.	Parameter	ISO Status	ATN Support	ISO Status	ATN Support
	direct-reference	O	X	M	N/A
	indirect-reference	O	O (See Note)	M	M
	data-value-descriptor	O	X	M	N/A

Note.— Indirect-reference contains a presentation-context-id value as specified in Table 4.3-3 when the single-ASN-1-type encoding form is used, and is absent otherwise.

4.6.6.3.3.2 User information encoding type

4.6.6.3.3.2.1 The User information parameter encoding choice shall be as specified in Table 4.6-17.

Table 4.6-17. User information encoding choice

		Sender		Receiver	
Ref.	Parameter	ISO Status	ATN Support	ISO Status	ATN Support
	single-ASN1-type	O	O	M	M
	octet-aligned	O	X	M	N/A
	arbitrary	O	O	M	M

4.6.7 Mapping to the Presentation Service

4.6.7.1 The mapping of ACSE APDUs and parameters to presentation service primitives shall be performed by the CF as specified in 4.3, which takes precedence over the direct mapping defined in clause 8 of ISO/IEC 8650-1.

Sub-Volume V

Internet Communications Service

NOTE ON THE SECOND EDITION

The list below shows the parts of this sub-volume that are different from similar parts of the first edition. It also shows the parts of the first edition that have been deleted and thus no longer appear in this edition.

Reference	Nature of change
5.2.4.1.2	Modification
5.2.4.1.3	Addition
5.2.5.1.6	Deletion
5.2.5.1.6.1	Deletion
5.2.5.1.6.1 (Note 1)	Deletion
5.2.5.1.6.1 (Note 2)	Deletion
5.3.2.2.2.1 a) and b)	Modification
5.3.2.2.3.2.2.3	Modification
5.3.2.2.3.2.2.3 (Note)	Addition
5.3.2.2.3.2.2.4	Addition
5.3.5.1 (Note 6)	Addition
5.3.5.2.2.2	Deletion
5.3.5.2.2.2.1	Deletion
5.3.5.2.2.2.2	Deletion
5.3.5.2.2.2.2 (Note)	Deletion
5.3.5.2.3.2 (Note 1)	Modification
5.3.5.2.3.2.2	Addition
5.3.5.2.4.2	Addition
5.3.5.2.5.2	Addition
5.3.5.2.6.5	Addition
5.3.5.2.6.5 (Note 1)	Addition
5.3.5.2.6.5 (Note 2)	Addition
5.3.5.2.6.8	Addition
5.3.5.2.6.9	Addition
5.3.5.2.6.10	Modification
5.3.5.2.7.2 (Note)	Modification
5.3.5.2.10.5	Addition
5.3.5.2.12.3.2 a) and b)	Modification
5.3.5.2.13.7 b)	Modification
5.3.5.2.13.8 b) and c)	Modification
5.3.5.2.14	Addition
5.3.5.2.14 (Note 1)	Addition
5.3.5.2.14 (Note 2)	Addition
5.3.5.2.14 (Note 3)	Addition
5.3.5.2.14.1	Addition
5.3.5.2.14.2	Addition
5.3.5.2.14.3	Addition
5.3.5.2.14.3 (Note 1)	Addition
5.3.5.2.14.3 (Note 2)	Addition
5.3.5.2.14.3 (Note 3)	Addition

Reference	Nature of change
5.3.5.2.15	Addition
5.3.5.2.15.1	Addition
5.3.5.2.15.1 (Note 1)	Addition
5.3.5.2.15.1 (Note 2)	Addition
5.3.5.2.16.1	Modification
5.3.5.2.16.2	Modification
5.3.5.2.16.3	Modification
5.3.5.2.16.4	Modification
5.3.5.2.16.5	Modification
5.3.5.2.16.6	Modification
5.4.3.8.4.4	Modification
5.6.3.1	Modification
5.6.3.1.2 (Note)	Addition
5.6.3.1.3	Deletion
5.6.3.4	Addition
5.6.3.4.1	Addition
5.6.3.4.1 (Note)	Addition
5.6.3.4.2	Addition
5.6.3.4.3	Addition
5.6.3.4.4	Addition
5.6.3.4.5	Addition
5.6.4.2	Modification
5.6.4.4	Modification
5.6.4.6	Modification
5.6.4.9	Modification
5.6.4.10	Modification
5.6.4.12	Modification
5.6.4.14	Modification
5.6.4.15	Modification
5.6.4.16	Modification
5.6.4.17	Modification
5.7.3	Modification
5.7.3 (Note)	Modification
5.7.3.1	Modification
5.7.6.2.1.1.1	Modification
5.7.6.2.1.1.2	Deletion
5.7.6.2.1.1.2 (Note 1)	Addition
5.7.6.2.1.5.6	Modification
5.7.6.2.1.5.14	Addition
5.7.6.3.4.1.1 g)	Modification
Table 5.7-6 Heading	Addition
5.7.6.5.2.2 a)	Modification
5.7.6.5.2.9	Deletion
5.7.6.5.4.1.3 (Note 2)	Deletion
5.7.6.5.4.1.4	Addition
5.7.6.5.4.1.4 (Note)	Addition

Reference	Nature of change
5.7.6.5.4.2.5.3	Deletion
5.7.6.5.4.2.5.3 (Note)	Deletion
5.7.6.5.6.3	Addition
5.7.6.5.6.4	Addition
5.7.6.5.7.1.1	Modification
5.7.6.5.7.1.3 (Note 1)	Modification
5.7.6.5.8.2.3	Addition
5.7.6.5.8.3.2	Deletion
5.7.6.5.8.3.2 (Note)	Deletion
5.7.7.8.2	Modification
5.8.2.1.3	Addition
5.8.2.1.3.1	Addition
5.8.2.1.3.2	Addition
5.8.2.1.3.3	Addition
Figure 5.8-1	Addition
5.8.2.1.3.4	Addition
5.8.2.1.3.4.1	Addition
5.8.2.1.3.4.1 (Note)	Addition
5.8.2.1.3.4.2	Addition
5.8.2.1.3.4.3	Addition
5.8.2.1.3.4.3.1	Addition
5.8.2.1.3.4.3.2	Addition
5.8.2.1.3.4.3.3	Addition
5.8.2.1.3.4.3.4	Addition
5.8.2.1.3.4.3.5	Addition
5.8.2.1.3.4.3.5 (Note 1)	Addition
5.8.2.1.3.4.3.5 (Note 2)	Addition
5.8.2.1.3.4.3.6	Addition
5.8.2.1.3.4.3.6 (Note)	Addition
5.8.2.1.3.4.3.7	Addition
Table 5.8-1	Addition
Table 5.8-1 (Note)	Addition
5.8.2.2.1	Modification
Table 5.8-2	Modification
5.8.3.2.3.2.2	Modification
5.8.3.2.3.2.5	Modification
5.8.3.2.3.3.6	Modification
5.8.3.2.8.1	Modification
Table 5.8-5	Modification
5.8.3.2.14.1 (Note)	Addition
5.8.3.3.2.1.1	Modification
5.8.3.5.5	Modification

5.1 INTRODUCTION

5.1.1 This sub-volume defines the provisions that ATN compliant End Systems (ESs) and Intermediate Systems (ISs) must implement in order to provide the ATN SARPs compliant “Internet Communications Service” to the “User” i.e. the Upper Layer Architecture as defined in Section 4 of the ATN SARPs. For the protocols, the majority of such provisions are specified in a tabular fashion under the title of “ATN Protocol Requirements Lists” (APRLs).

5.1.2 This sub-volume comprises nine Chapters as introduced below.

Chapter 5.1, contains introductory material to the remainder of the Section.

Chapter 5.2, contains pertinent definitions of the Internet Routing Architecture and components including Routing Domains, Administrative Domains, Routing Domain Confederations, ATN Backbone, ATN Islands etc. Furthermore it contains system level provisions related to communications protocol support for ATN End Systems and Intermediate Systems, and SARPs related to security and priority handling within the ATN internet.

Chapter 5.3, contains provisions related to the deployment of ATN components within the ATN Internet, to the use of routing information, to the definition of routing policies, and to the procedures for initiating the exchange of routing information.

Chapter 5.4, contains provisions related to the ATN Internet addressing architecture and responsibilities related to the definition and allocation of ATN Internet address fields.

Chapter 5.5, contains “Transport Layer” provisions applicable to ATN End Systems. Provisions for the ISO Connection Oriented Transport Protocol (Class 4) and the Connectionless Transport Protocol are defined.

Chapter 5.6, contains “Inter-Network Layer” provisions, based on the ISO Connectionless Network Protocol (CLNP), applicable to ATN End Systems and ATN Intermediate Systems.

Chapter 5.7, contains provisions related to the use of the various candidate Ground/Ground and Air/Ground subnetworks of the ATN in order to ensure successful inter-operation of ATN Intermediate Systems and the subnetworks to which they are attached. Compression techniques are also defined to enable the efficient use of the limited bandwidth available over such Air/Ground subnetworks.

Chapter 5.8, contains provisions related to the exchange of routing information between ATN Intermediate Systems using the Inter Domain Routing Information Exchange Protocol (IDRP) and specific features of the ES-IS protocol.

Chapter 5.9, contains a recommendation regarding the implementation of Internet Systems Management.

5.2 DEFINITIONS AND CONCEPTS

5.2.1 Objectives and Goals

Note 1.— In computer data networking terminology, the infrastructure required to support the interconnection of automated ATM (Air Traffic Management) systems is referred to as an internet. Simply stated, an internet comprises the interconnection of computers with gateways or routers via real subnetworks. This allows the construction of a homogeneous virtual data network in an environment of administrative and technical diversity. Given the desire to interconnect an evolving and ever wider variety of aircraft- and ground-based computers to accomplish ATM automation, it is clear that the civil aviation community needs a global data internet. The internetworking infrastructure developed by ICAO (International Civil Aviation Organization) for this purpose is the ATN.

Note 2.— The ATN design allows communication services for different user groups, i.e. air traffic services (ATS), aeronautical operational control (AOC), aeronautical administrative communications (AAC) and aeronautical passenger communications (APC). The design provides for the incorporation of different Air/Ground subnetworks (e.g. SSR Mode S, AMSS, VDL) and different Ground/Ground subnetworks, resulting in a common data transfer service. These two aspects are the basis for interoperability of the ATN and will provide a reliable data transfer service for all users. Furthermore, the design is such that user communications services can be introduced in an evolutionary manner.

Note 3.— The ATN is capable of operating in a multinational environment with different data communication service providers. The ATN is capable of supporting Air Traffic Service Communication (ATSC) as well as Aeronautical Industry Service Communication (AINSC).

Note 4.— The ATN is capable of supporting the interconnection of End Systems (ESs) and Intermediate Systems (ISs) using a variety of subnetwork types.

5.2.2 Definitions

Note.— This specification makes extensive use of the definitions, concepts and terminology derived from the OSI Reference Model (ISO 7498 parts 1-4) and the OSI Routing Framework (ISO/IEC TR 9575).

5.2.2.1 The ATN Internet

5.2.2.1.1 The ATN shall consist of a set of interconnected Routing Domains (RDs), within the global OSI Environment (OSIE). Each such RD shall contain Air Traffic Service Communication (ATSC) and/or Aeronautical Industry Service Communication (AINSC) related Intermediate and End Systems.

5.2.2.1.2 A Routing Domain that declares itself to be a Transit Routing Domain (TRD) shall implement a Routing Policy that supports the relaying of Network Protocol Data Units (NPDUs) received from at least one other Routing Domain to destinations in another Routing Domain.

5.2.2.1.3 Otherwise, the Routing Domain shall be defined as an End Routing Domain (ERD).

5.2.2.2 ATN RDs

5.2.2.2.1 General

5.2.2.2.1.1 An ATN RD shall meet the requirements specified in ISO/IEC TR 9575 for a Routing Domain and shall include one or more ATN Routers.

5.2.2.2.1.2 Every ATN RD shall have at least one Routing Domain Identifier (RDI).

5.2.2.2.1.3 Each RDI shall unambiguously identify a single RD.

Note.— An RDI is a generic Network Entity Title (NET), and has the same syntax as an ATN NSAP Address; alias RDIs are permitted.

5.2.2.2.2 Fixed RDs

5.2.2.2.2.1 Each State and Organisation participating in the ATN shall operate one or more ATN RDs, comprising Air/Ground and Ground/Ground Routers as required to interconnect with Mobile RDs and other ground-based ATN RDs, respectively.

Note.— Adjacent States and/or Organisations may alternatively combine their RDs into a single RD.

5.2.2.2.3 Mobile RDs

5.2.2.2.3.1 Each ATN equipped Mobile platform (e.g. an aircraft) shall operate at least one ATN RD. This shall be an End Routing Domain.

5.2.2.2.3.2 This ERD shall include ATSC and AINSC related Intermediate and End Systems contained within this Mobile platform, and at least one Airborne Router (Router Class 6 or 7 as defined in Table 5.2.-1).

Note.— *An ATN Mobile platform may operate multiple ERDs.*

5.2.2.2.3.3 When more than one Airborne Router (BIS) is installed on board an aircraft, then each shall be in a separate Routing Domain.

5.2.2.2.3.4 **Recommendation.**— *ATSC and AINSC End Systems and Intermediate Systems (non-BISs) located within a Mobile platform should form a single Routing Domain including the airborne router (BIS) referred to in the above note, within the appropriate Administrative Domain.*

Note 1.— *A single routing domain minimizes the transfer of routing information over low-bandwidth Air/Ground subnetworks.*

Note 2.— *It is anticipated that other classes of Mobile platforms (e.g. airport surface vehicles, etc.) may be operated as ATN routing domains in the future.*

5.2.2.3 The Ground ATN Internet

5.2.2.3.1 General

5.2.2.3.1.1 The Ground ATN Internet shall consist of one or more ATN Island RDCs (Routing Domain Confederations).

5.2.2.3.2 ATN Island RDC

5.2.2.3.2.1 Each ATN Island shall comprise one or more ATN RDs forming a single ATN Island RDC.

5.2.2.3.2.2 An ATN Island RDC shall not contain any ATN Mobile RDs.

Note.— An example ATN Island RDC topology is presented in Figure 5.2-1.

5.2.2.3.3 The Fixed ATN RDC

5.2.2.3.3.1 The Fixed ATN RDC shall comprise all ATN RDs other than ATN Mobile RDs.

Note.— The Fixed ATN RDC enables a ground ATN Router to advertise a route to a Mobile, the destination of which is the entire fixed ATN, without having to enumerate the RDIs (Routing Domain Identifiers) of all ATN RDs in the RD_Path Attribute.

5.2.2.4 The Global ATN Backbone

5.2.2.4.1 General

5.2.2.4.1.1 The Global ATN Backbone shall comprise at least one ATN RD from each ATN Island, interconnected either directly or indirectly via other members of the Global ATN Backbone.

Note.— The purpose of the Global ATN Backbone is to provide a high availability core network of ATN Routers supporting ATN Mobile routing.

5.2.2.4.2 ATN Island Backbone RDCs

5.2.2.4.2.1 **Recommendation.**— Within each ATN Island, those ATN RDs that are members of the Global ATN Backbone should form a single RDC, which is referred to as the ATN Island Backbone RDC.

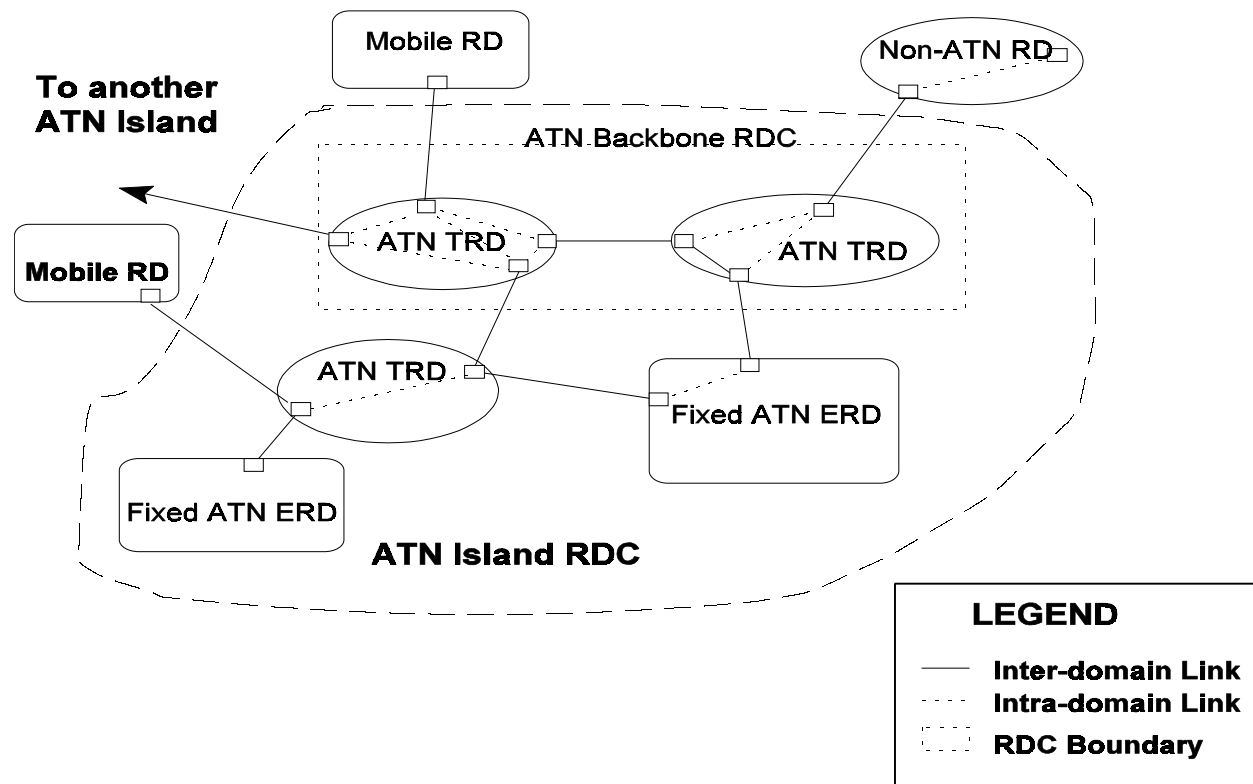


Figure 5.2-1 Example ATN Island Routing Domain Confederation Structure

5.2.2.4.2.2 An ATN Island Backbone RDC, when present, shall be nested within an ATN Island RDC.

Note 1.— The purpose of the ATN Island Backbone RDC is to permit more than one ATN RD to act as the default route provider for an ATN Island. It also provides a containment boundary to limit the impact of changes in routes to Mobile RDs to only the members of the Backbone RDC and not to the rest of the ATN Island.

Note 2.— This is only a recommended practice as in some regions, simpler, or other alternative structures may be more appropriate for an ATN Island.

5.2.2.5 The “Home” Domain

5.2.2.5.1 Aircraft for which inter-Island communications are required shall have a “Home” domain, which is a Routing Domain in an ATN Island.

Note 1.— This “home” needs not be in either the ATN Island through which the aircraft is currently reachable, or in the ATN Island with which communication is required.

Note 2.— The role of the “Home” domain is to advertise a default route to all the aircraft belonging to an airline, or the General Aviation aircraft of a given country of registration. This default route is advertised to the ATN Global Backbone in line with the routing policies specified in 5.3.7.

5.2.2.6 Administrative Domains and the ATN

5.2.2.6.1 The Administrative Domain of each administration, and aeronautical industry member that operates one or more ATN RDs shall comprise both their ATN RDs, and any non-ATN RDs that they operate.

Note 1.— The Routing Policies for communication between ATN and non-ATN RDs within the same Administrative Domain is a local matter.

Note 2.— While meeting the requirements of the SARPs, the distribution of end system and intermediate system functionality and the use of interworking processes exclusively within an Administrative Domain is a local matter.

5.2.2.7 Default Routes

5.2.2.7.1 The default route to all aircraft shall be a route in the context of IDRP that:

- a) is available to all traffic types (see 5.2.7.1.2), and
- b) has in its destination two NSAP Address prefixes. One of these is the NSAP Address prefix that is common to all AINSC Airborne Systems and only AINSC Airborne Systems, and the other is the NSAP Address prefix that is common to all ATSC Airborne Systems and only ATSC Airborne Systems.

5.2.2.7.2 The default route to all the aircraft belonging to an airline or the General Aviation Aircraft of a given country of registration shall be a route in the context of IDRP that:

- a) is available to all traffic types (see 5.2.7.1.2), and
- b) has in its destination an NSAP address prefix which is common to all Airborne Systems and only those Airborne Systems of the aircraft that belong to that airline or are registered in that country.

5.2.3 ATN End Systems

Note 1.— ATN End Systems are capable of communicating with other ATN End Systems, either directly or indirectly, to provide end-to-end communication service for Air/Ground or Ground/Ground applications, or both.

Note 2.— An ATN End System is a realisation of the OSI End System architectural entity.

Note 3.— An ATN End System supports one or more ATN Applications and supports their communication over the ATN by providing either the connection mode transport service, or the connectionless mode transport service, or both.

5.2.3.1 Physical and Data Link Layer

5.2.3.1.1 ATN End Systems shall implement the appropriate Physical and Data Link Layer functions for access to the ATN subnetwork(s) to which they are attached.

5.2.3.2 Network Layer

5.2.3.2.1 ATN End Systems shall implement:

- a) The End System provisions of ISO/IEC 8473, as specified in 5.6, as the Subnetwork Independent Convergence Function (SNICF).
- b) a Subnetwork Access Protocol (SNAcP) suitable for each underlying subnetwork.
- c) a Subnetwork Dependent Convergence Function (SND CF) providing byte and code independent service to the SNICF (i.e. ISO/IEC 8473) via the appropriate Subnetwork Access Protocol, as specified in 5.7.

5.2.3.2.2 **Recommendation.**— *ATN End Systems should implement the End System provisions of ISO/IEC 9542 to facilitate the exchange of routing information between the ES and any locally attached IS(s).*

5.2.3.3 Transport Layer

5.2.3.3.1 Depending on the requirements of the application and its supporting upper-layer protocols, ATN End Systems shall implement either one or both of the following:

- a) ISO/IEC 8073 as specified in 5.5.
- b) ISO/IEC 8602 as specified in 5.5.

5.2.3.4 Upper Layers

Note.— The requirements for session, presentation and application layer protocols to support end-user applications on ATN End Systems are defined in Section 4 of the ATN SARPs.

5.2.3.5 Applications

Note.— The requirements for Air/Ground and Ground/Ground applications are contained in Sections 2 and 3 of the ATN SARPs respectively.

5.2.4 ATN Routers

Note 1.— ATN Routers are capable of the relaying and routing of Network Layer protocol data units with other ATN Routers and with directly connected ATN End Systems.

Note 2.— An ATN Router is a realisation of the OSI Intermediate System architectural entity. ATN Routers that additionally implement ISO/IEC 10747 are also known as Boundary Intermediate Systems (BISs).

5.2.4.1 ATN Router Classes

5.2.4.1.1 The classes of ATN Router and the Routing Protocols supported, that are recognised by this specification, are listed below in Table 5.2-1.

Table 5.2-1 ATN Router Classes

Class	Name	Routing Protocols Supported
1.	Static Router	ISO/IEC 9542 (optional)
2.	Level 1 Router	ISO/IEC 9542 (optional) ISO/IEC 10589 Level 1 only
3.	Level 2 Router	ISO/IEC 9542 (optional) ISO/IEC 10589 Level 1 and Level 2
4.	Ground/Ground Router	ISO/IEC 9542 (optional) ISO/IEC 10589 (optional) ISO/IEC 10747
5.	Air/Ground -Router (ground based)	ISO/IEC 9542 ISO/IEC 10589 (optional) ISO/IEC 10747 Route Initiation Procedures (see 5.3.5.2)
6.	Airborne Router with IDRP	ISO/IEC 9542 ISO/IEC 10747 Route Initiation Procedures (see 5.3.5.2)
7.	Airborne Router without IDRP	ISO/IEC 9542 Route Initiation Procedures (see 5.3.5.2)

Note 1.— Classes 1, 2 and 3 are only for use within an ATN Routing Domain and their specification is a local matter.

Note 2.— The intra-domain parts of Router Classes 4 and 5 are also a local matter.

Note 3.— The intra-domain parts of Router Classes 6 and 7 are concerned with the interconnection of avionics to the airborne router and are the subject of aeronautical industry standards.

Note 4.— Router Classes 5, 6 and 7 describe routers that support at least one ATN Mobile Subnetwork.

5.2.4.1.2 All ATN Routers (i.e. Router Classes 1 to 7 inclusive) shall support the ISO/IEC 8473 Connectionless Network Protocol (CLNP) as specified in 5.6, including the use of the CLNP options security parameter, and shall interpret and obey the Routing Policy Requirements expressed therein, whilst routing the packet in accordance with any restrictions placed on the traffic types that may be carried over a given ATN Subnetwork, by forwarding CLNP NPDUs.

5.2.4.1.3 With the exception of Airborne Routers that implement the procedures for the optional non-use of IDRP (i.e. Router Class 7), all ATN Inter-Domain Routers (i.e. Router Classes 4 to 6 inclusive) shall support the ISO/IEC 10747 Inter-Domain Routing Protocol (IDRP) as specified in 5.8 for the exchange of inter-domain routing information according to 5.3.6 and 5.3.7.

5.2.4.1.4 An Airborne (Router Classes 6 or 7) or Air/Ground Router (Router Class 5) shall support the Mobile SNDCEF specified in 5.7 for the use of CLNP over an ATN Mobile Subnetwork, and the ISO/IEC 9542 ES-IS routing information exchange protocol, as specified in 5.8 for support of the route initiation procedures specified in 5.3.5.2.

5.2.4.2 Physical and Data Link Layers

5.2.4.2.1 ATN Routers shall implement the appropriate Physical and Data Link Layer functions for access to the ATN subnetwork(s) to which they are attached.

5.2.4.3 Network Layer

5.2.4.3.1 An ATN Router shall implement:

- a) the Intermediate System provisions of ISO/IEC 8473, as specified in 5.6, as the Subnetwork Independent Convergence Function (SNICF).
- b) a Subnetwork Access Protocol (SNACP) suitable for each underlying subnetwork.
- c) a Subnetwork Dependent Convergence Function (SNDCEF) providing byte and code independent service to the SNICF (i.e. ISO/IEC 8473) via the selected Subnetwork Access Protocol, as specified in 5.7.
- d) The routing protocols specified in Table 5.2-1 for the Router's Router Class, as specified in 5.8.
- e) The Route Initiation procedures appropriate to the Router Class, as specified in 5.3.
- f) Where an ATN Router is directly connected to one or more Mobile Subnetworks, it shall implement a sub-set of the ISO/IEC 9542 for operation over those subnetworks to facilitate the exchange of addressing information (BIS Network Entity Title) between the Router and its peer as specified in 5.3 (see 5.3.5.2) and in 5.8.

5.2.4.3.2 ATN Routers of class 5 (Air/Ground Routers) and of class 7 (Airborne Routers without IDRP) shall also implement the mechanisms necessary to support the “optional non-use of ISO/IEC 10747” as specified in 5.3.

5.2.4.3.3 **Recommendation.**— *All ATN Airborne Routers should support the use of ISO/IEC 10747 (i.e. Class 6 is the preferred Airborne Router Class).*

Note.— *Some States may elect to support the optional non-use of airborne IDRP procedures in their Air/Ground Routers; however, Regional Implementation Planning Groups must acknowledge the requirement for aircraft using IDRP within the Region to communicate with an Air/Ground Router, independent of how that is accomplished.*

5.2.5 ATN Subnetworks

Note.— An ATN Subnetwork is any fixed or Mobile data communications network that fulfils the following requirements.

5.2.5.1 Requirements for All ATN Subnetworks

5.2.5.1.1 Both fixed and Mobile ATN subnetworks shall conform to the following requirements.

5.2.5.1.2 Byte and Code Independence

5.2.5.1.2.1 Data shall be transferred through ATN Subnetworks in a byte and code independent manner.

Note.— If necessary, this byte and code independence may be ensured through the services of the SNDCE.

5.2.5.1.3 Subnetwork QoS

5.2.5.1.3.1 A Subnetwork service provider shall provide an indication of the Subnetwork Quality of Service (QoS) available, in order to support the internetwork routing decision process.

5.2.5.1.4 Subnetwork Addressing

5.2.5.1.4.1 An ATN subnetwork shall provide a mechanism for uniquely and unambiguously identifying each ATN router attached to that subnetwork.

5.2.5.1.5 Internal Subnetwork Routing

5.2.5.1.5.1 Routing between specified source and destination Subnetwork Point of Attachment (SNPA) addresses on an ATN subnetwork shall be carried out by mechanisms internal to the subnetwork.

5.2.5.2 Requirements for ATN Mobile Subnetworks

5.2.5.2.1 General

5.2.5.2.1.1 An ATN Mobile Subnetwork shall conform to the following requirements.

5.2.5.2.2 Invocation of Subnetwork Priority

5.2.5.2.2.1 When priority is implemented within that subnetwork, an ATN Mobile Subnetwork shall provide a SNAcP mechanism for invocation of subnetwork priority.

5.2.5.2.3 Invocation of Subnetwork Quality of Service for Mobile Subnetworks

5.2.5.2.3.1 **Recommendation.**—*ATN Mobile Subnetworks should provide a mechanism for invocation of subnetwork QoS.*

Note 1.— Subnetwork QoS parameters include transit delay, protection against unauthorized access, cost determination and residual error probability.

Note 2.— ATN Mobile Subnetworks may allocate subnetwork resources on a per user or per subnetwork connection basis in order to make available a different QoS.

5.2.5.2.4 Connection-Mode Subnetwork Service

5.2.5.2.4.1 An ATN Mobile Subnetwork shall provide a connection-mode service between SNPAs, with a well-defined start and end to a connection, and with reliable, sequenced SNSDU transfer over that connection.

5.2.5.2.4.2 When QoS is available on a per subnetwork connection basis, the SNACp shall provide mechanisms for selecting a specific QoS when the subnetwork connection is established.

Note 1.— A Mobile Subnetwork implementing ISO/IEC 8208 to provide a connection-mode service between SNPAs meets this requirement; however, where appropriate, an alternative protocol providing the same service may be used.

Note 2.— This requirement does not imply the need for a single Mobile SNACp.

5.2.5.2.5 Connectivity Status Changes

Note.— ATN Mobile Subnetworks may provide a mechanism for detection of change in media connectivity and for the conveyance of this information to connected ATN routers.

5.2.5.2.5.1 If a Mobile Subnetwork provides subnetwork connectivity information, the subnetwork shall convey this information to connected subnetwork service users (i.e. connected ATN routers), in order to initiate operation of the internetwork routing protocols as specified in 5.3.

Note.— It is desirable for the Intermediate System - Systems Management Entity (IS-SME) to be notified as soon as possible by the SN-SME when communication is possible with a newly attached BIS and for an immediate decision to be made as regards bringing up the link.

5.2.5.2.6 Segmentation/Reassembly Mechanism

5.2.5.2.6.1 **Recommendation.**— *An ATN Mobile Subnetwork should provide a mechanism that allows the conveyance of large SNSDUs greater than the subnetwork's internal packet size between SNPAs.*

Note.— It is the responsibility of the subnetwork to ensure that this data is efficiently segmented and/or concatenated for efficient transfer over the physical medium. If this capability is not present within an ATN Mobile Subnetwork, ISO/IEC 8473 can support segmentation of NPDUs for transit over subnetworks with small maximum SNSDU sizes.

5.2.6 Quality of Service Concept

Note 1.— In the ATN, the Quality of Service provided to applications is maintained using capacity planning techniques that are outside of the scope of this specification. Network Administrators are responsible for designing and implementing a network that will meet the QoS requirements of the ATN applications that use it.

Note 2.— Network Administrators may take advantage of the QoS requirements signalled by the ATSC Class (see 5.2.7), in order to ensure that only those parts of the ATN that support the QoS requirements of ATSC applications, need be designed to meet those requirements.

5.2.7 ATN Security Concept

Note 1.— ATN Security Functions are concerned with:

- a) *Protecting ATN Data Link applications from internal and external threats;*
- b) *Ensuring that application Quality of Service and Routing Policy Requirements are maintained, including service availability; and,*
- c) *Ensuring that Air/Ground subnetworks are used in accordance with ITU resolutions on frequency utilisation.*

Note 2.— Other than through the provision of physical security mechanisms, no security mechanisms are provided in the ATN Internet for protecting ATN Data Link applications. ATN Data Link applications need to develop their own security mechanisms for countering any threats to their proper operation. This may change in future versions of the specification.

Note 3.— There are currently no mechanisms for protecting the Routing Information Base from an attacker. However, the use of ISO/IEC 10747 type 2 authentication is under consideration for specification in future versions of this specification.

Note 4.— The ATN Internet does provide mechanisms to support items (b) and (c) above. These mechanisms are defined to take place in a common domain of trust, and use a Security Label in the header of each CLNP PDU to convey information identifying the “traffic type” of the data and the application’s routing policy and/or strong QoS Requirements. No mechanisms are provided to protect the integrity of this label or its binding to the application data.

Note 5.— In order to permit the later extension of the ATN to handle classified data, the Security Label in the CLNP PDU header can additionally be used to convey Security Classification information.

Note 6.— The Routing Information necessary to support this Security Label is maintained through information conveyed in the ISO/IEC 10747 Security Path Attribute about each route. ATN Routers of classes 4 and above reference this routing information during the NPDU forwarding process in order to meet the application’s requirements expressed through the NPDU’s Security Label and to enforce any applicable ITU resolutions on frequency utilisation.

5.2.7.1 The ATN Security Label

5.2.7.1.1 General

5.2.7.1.1.1 The ATN Security Label shall be encoded according to 5.6.2.2.2.

5.2.7.1.1.2 An ATN Security Label shall be provided as part of the header of every CLNP NPDU, except for those that convey General Communications applications data.

Note.— The above implies that any CLNP NPDU that does not contain an ATN Security Label contains General Communications data.

5.2.7.1.2 Traffic Types

5.2.7.1.2.1 A CLNP Data NPDU's Security Label shall identify the "Traffic Type" of its user data, as either:

- a) ATN Operational Communications
- b) ATN Administrative Communications
- c) ATN Systems Management Communications.

Note.— ATN Operational Communications traffic type is broken down into two categories: ATSC and AOC (see Table 5.6-1).

5.2.7.1.2.2 For the ATN Operational Communications traffic type and the ATSC traffic category, routing policy requirements shall be expressed through further information encoded into the Security Label, as either:

- a) A preferred ATSC Class, or
- b) no routing policy preference.

5.2.7.1.2.3 For the ATN Operational Communications traffic type and the AOC traffic category, routing policy requirements shall be expressed through further information encoded into the Security Label, as either no routing policy preference, or an ordered list of appropriate Air/Ground subnetworks to be used.

Note.— The possible ordering of Air/Ground subnetworks are specified in Table 1-1.

5.2.7.1.3 ATSC Class

Note 1.— The Transit Delay semantics of ATSC Class are defined in Table 1-1.

Note 2.— The semantics of the ATSC Class for other QoS metrics and availability are outside of the scope of this specification.

5.2.7.1.4 Security Classification

Note.— The Security Classification may be used to convey the confidentiality level of application data.

5.2.7.2 Applications Use of ATN Security Labels

5.2.7.2.1 ATN Data Link applications shall specify an ATN Security Label for each message category that they support. This ATN Security Label shall identify:

- a) the Traffic Type appropriate for the message; and,
- b) for ATN Operational Communications applications, the application's requirements for the routing of the message, if any, expressed as specified in 5.2.7.1.2.

5.2.7.2.2 When sent using the connection-mode transport service, a message shall only be conveyed over a transport connection which is associated with the same ATN Security Label as that specified for the message's message category.

5.2.7.2.3 When sent using the connectionless-mode transport service, the TSDU conveying that message shall be associated with the ATN Security Label of the specified message category.

5.2.7.3 Transport Layer Security

5.2.7.3.1 In the Connection Mode

5.2.7.3.1.1 Except when a transport connection is used to convey general communications data, each transport connection shall be associated with a single ATN Security Label.

5.2.7.3.1.2 The value of this label shall be determined when the connection is initiated.

Note.— It is not possible to change an ATN Security Label during the lifetime of a transport connection.

5.2.7.3.1.3 Every NSDU passed to the Network Layer that contains a TPDU from a transport connection associated with an ATN Security Label shall be associated with the same ATN Security Label.

Note.— The Network Layer functions may then encode this ATN Security Label in the NPDU header.

5.2.7.3.1.4 TPDUs from transport connections associated with different ATN Security Labels shall not be concatenated into the same NSDU.

5.2.7.3.1.5 When an incoming CR TPDU is received, the ATN Security Label, if any, encoded into the header of the NPDU that conveyed it, shall define the ATN Security Label that is associated with the transport connection.

Note 1.— The mechanism by which the connection initiator provides the appropriate ATN Security Label is a local matter. For example, it may be identified by an extension to the transport service interface, be implicit in the choice of a given TSAP, or be identified using a Systems Management function.

Note 2.— Some applications may reject incoming transport connections for which the ATN Security Label is inappropriate. Again, the mechanism by which the Transport provider passes to its user the ATN Security Label associated with an incoming transport connection is a local matter.

5.2.7.3.2 In the Connectionless Mode

5.2.7.3.2.1 In the connectionless mode, unless used to convey General Communications data, each incoming or outgoing TSDU shall be associated with an ATN Security Label.

5.2.7.3.2.2 For outgoing TSDUs, this ATN Security Label shall be encoded into the header of the resulting NPDU(s).

5.2.7.3.2.3 For incoming TSDUs, the associated ATN Security Label shall be the ATN Security Label that was encoded into the header of the incoming NPDU(s) that contained the TSDU.

Note.— The mechanism by which ATN Security Labels are associated with TSDUs is a local matter.

5.2.7.4 Network Layer Security

5.2.7.4.1 Service Provider to the Transport Layer

5.2.7.4.1.1 The Network Service shall provide a mechanism that permits an ATN Security Label to be associated with an NSDU:

- a) When passed from the Transport Layer to the Network Layer in an NS-UNITDATA.request. This ATN Security Label shall be encoded into the header of the corresponding CLNP NPDU(s) according to 5.6.2.2.2.
- b) When passed from the Network Layer to the Transport Layer in an NS-UNITDATA.indication. This ATN Security Label shall be that received in the header of the associated CLNP NPDU(s).

5.2.7.4.2 Routing Control

5.2.7.4.2.1 When present in an NPDU header, the network layer routing functions shall ensure that:

- a) The Routing Policy requirements, if any, encoded into the ATN Security Label are obeyed, and
- b) The NPDU is only routed over paths through the internetwork which both permit and are suitable for data of the traffic type identified by the ATN Security Label.

Note 1.— 5.3.2.2 specifies the forwarding procedures that ensure the above.

Note 2.— The Security Information conveyed in ISO/IEC 10747 (IDRP) routes is used to provide the forwarding process with the information needed to support the above.

5.2.7.4.3 Protection of the Routing Information Base

5.2.7.4.3.1 IDRP type 1 Authentication, as specified in ISO/IEC 10747, shall be used as a mechanism for ensuring the integrity of routing information exchange by IDRP.

Note.— A later extension to support type 2 authentication will enable the routing information base to be protected from attackers that try to modify routing information while in transit, or which attempt to masquerade as genuine ATN Routers.

5.2.7.5 Subnetwork Provisions

Note.— There are no requirements for security mechanisms in ATN Subnetworks. However, Administrations and other Organisations implementing ATN subnetworks are encouraged to ensure the general security and availability of ATN subnetworks through the use of physical security mechanisms.

5.2.8 ATN Use of Priority

Note 1.— The purpose of priority is to signal the relative importance and/or precedence of data, such that when a decision has to be made as to which data to action first, or when contention for access to shared resources has to be resolved, the decision or outcome can be determined unambiguously and in line with user requirements both within and between applications.

Note 2.— In the ATN, priority is signalled separately by the application in the transport layer and network layer, and in ATN subnetworks. In each case, the semantics and use of priority may differ. Figure 5.2-2 illustrates where priority is applied in the ATN, and where it is necessary to map the semantics and syntax of ATN priorities.

Note 3.— In the ATN Internet, priority has the essential role of ensuring that high priority safety related data is not delayed by low priority non-safety data, and in particular when the network is overloaded with low priority data.

5.2.8.1 Application Priority

Note.— Priority in ATN Application Protocols is used to distinguish the relative importance and urgency of application messages within the context of that application alone.

5.2.8.1.1 For the purpose of

- a) distinguishing the relative importance and urgency of messages exchanged by different ATN Applications, and
- b) distinguishing the relative importance and urgency of messages of the same application during their transit through the ATN,

application messages shall be grouped into one or more categories listed in Table 1-2 .

Note.— An ATN Application may include messages from more than one category.

5.2.8.1.2 When a message is sent between ATN Application Entities, the message shall be sent using either:

- a) a transport connection established using the Transport Connection Priority listed in Table 1-2 for the message's message category, or
- b) the connectionless transport service, signalling the Connectionless Transport Service Priority listed in Table 1-2 for the message's message category.

Note.— The priority of an individual transport connection cannot be changed during the lifetime of the connection. Therefore, if an application exchanges messages belonging to more than one message category using the connection mode transport service, then a separate transport connection needs to be established for each message category.

5.2.8.2 Transport Connection Priority

Note 1.— Transport connection priority is concerned with the relationship between transport connections and determines the relative importance of a transport connection with respect to (a) the order in which TCs are to have their QoS degraded, if necessary, and (b) the order in which TCs are to be broken in order to recover resources.

Note 2.— The transport connection priority is specified by the transport user either explicitly or implicitly, when the transport connection is established.

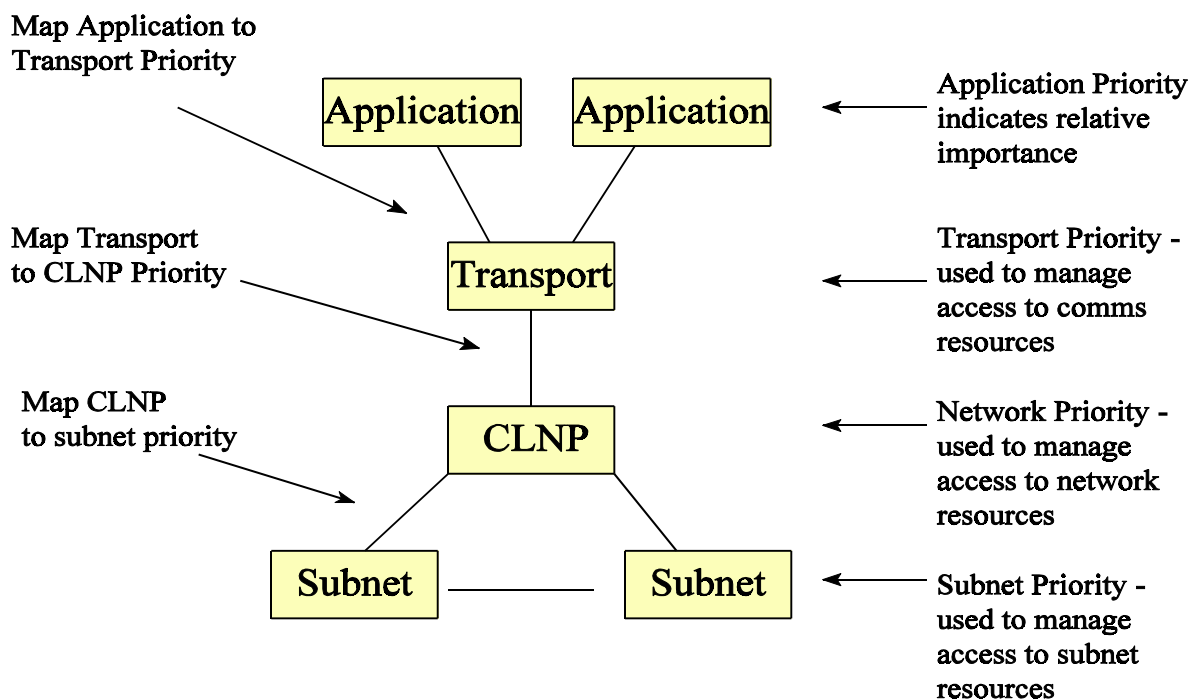


Figure 5.2-2 Use of Priority in the ATN

5.2.8.2.1 When an ATN Transport Layer entity is unable to satisfy a request for a transport connection from either a local or remote TSAP, and which is due to insufficient local resources available to the transport layer entity, then it shall terminate a lower priority transport connection, if any, in order to permit the establishment of a new higher priority transport connection.

Note.— Implementations may also use transport connection priority to arbitrate access to other resources (e.g. buffers). For example, this may be achieved by flow control applied to local users, by discarding received but unacknowledged TPDU's, by reducing credit windows, etc.

5.2.8.2.2 All TPDU's sent by an ATN Transport Layer Entity shall be transferred by the ATN Internet Layer, using the Network Protocol Priority that corresponds to the transport connection's priority according to Table 1-2.

5.2.8.3 Connectionless Transport Service Priority

Note.— There are no procedures required of the ATN Connectionless Transport Entity in respect of priority, except for mapping the TSDU priority supplied by the service user (i.e. an ATN Application), to the corresponding Network Layer Priority, and vice versa.

5.2.8.3.1 All UD TPDUs sent by an ATN Transport Layer Entity shall be transferred by the ATN Internet Layer using the Network Protocol Priority that corresponds to the TSDU priority provided by the service user according to Table 1-2.

5.2.8.4 ATN Internet Priority

Note.— In the ATN Internet Layer, an NPDU of a higher priority is given preferred access to resources. During periods of higher network utilisation, higher priority NPDUs may therefore be expected to be more likely to reach their destination (i.e. are less likely to be discarded by a congested router) and to have a lower transit delay (i.e. be more likely to be selected for transmission from an outgoing queue) than are lower priority packets.

5.2.8.4.1 ATN Internet Entities shall maintain their queues of outgoing NPDUs in strict priority order, such that a higher priority NPDU in an outgoing queue will always be selected for transmission in preference to a lower priority NPDU.

Note.— Priority zero is the lowest priority.

5.2.8.4.2 During periods of congestion, or when any other need arises to discard NPDUs currently held by an ATN Internet Entity, lower priority NPDUs shall always be discarded before higher priority NPDUs.

Note.— In addition to NPDUs containing user (i.e. transport layer) data, the Internet Layer also forwards routing information contained in CLNP Data PDUs (e.g. IDRP) and as distinct NPDUs (e.g. ES-IS). These must all be handled at the highest priority if changes to network topology are to be quickly actioned and the optimal service provided to users.

5.2.8.4.3 BISPDUs exchanged by IDRP shall be considered as Network/Systems Management category messages, and shall be sent using CLNP priority 14.

5.2.8.4.4 ES-IS (ISO/IEC 9542) PDUs shall be implicitly assumed to have priority 14 and shall be forwarded as if they were CLNP PDUs of priority 14.

Note.— The priority encoded in an ISH PDU conveys the priority of the sending system, and not the priority of the PDU.

5.2.8.5 ATN Subnetwork Priority

5.2.8.5.1 Connection-Mode Subnetworks

Note 1.— In a connection-mode ATN subnetwork, priority is used to distinguish the relative importance of different data streams (i.e. the data on a subnetwork connection), with respect to gaining access to communications resources and to maintaining the requested Quality of Service.

Note 2.— On some subnetworks (e.g. public data networks), not all data streams will be carrying ATN messages. Therefore, subnetwork priority is also used to distinguish ATN and non-ATN data streams.

Note 3.— So as not to incur the overhead and cost of maintaining too many simultaneous subnetwork connections, NPDUs of a range of Network Layer priorities may be sent over the same subnetwork connection.

5.2.8.5.1.1 When an ATN connection mode subnetwork does not support prioritisation of subnetwork connections, then the ATN Internet Entity shall not attempt to specify a subnetwork connection priority, and NPDUs of any priority may be sent over the same subnetwork connection.

5.2.8.5.1.2 When an ATN connection mode subnetwork does support prioritisation of subnetwork connections, then unless the relationship between ATN Internet Priority and subnetwork priority is explicitly specified by the subnetwork specification, the following shall apply:

- a) Subnetwork connections shall be established as either “High” or “Low” priority connections.
- b) For the “Low” priority connection type, the priority to gain a connection, keep a connection and for data on the connection shall be the defaults for routine use of the subnetwork.
- c) “High” priority connections shall be used to convey NPDUs of priority ten and above. “Low” priority connections shall be used to convey all other NPDUs.

Note.— The above does not apply to the AMSS Subnetwork, which has specified its own priority mapping scheme.

5.2.8.5.1.3 When a subnetwork connection is established between two ATN Internet Entities and no other subnetwork connection between these two entities exists over any subnetwork, then that subnetwork connection shall always be established at a priority suitable for conveying NPDUs of priority 14 (i.e. Network/Systems Management).

Note.— This is to ensure that routing information can be exchanged at the appropriate priority.

5.2.8.5.2 Connectionless-Mode Subnetworks

Note 1.— The purpose of priority on a connectionless-mode subnetwork is to provide higher priority NPDUs with preferred access to subnetwork resources.

Note 2.— The relationship between NPDUs priority and subnetwork priority is subnetwork specific.

5.2.8.5.2.1 When an NPDUs is sent over a connectionless-mode ATN Subnetwork which supports data prioritisation, then the subnetwork priority assigned to the transmitted packet shall be that specified by the subnetwork provider as corresponding to the NPDUs priority.

5.3 ATN ROUTING

5.3.1 Introduction

5.3.1.1 Scope

Note.— This chapter provides requirements and recommendations pertaining to the deployment of ATN components within the ATN Internet; use of routing information distributed according to ISO/IEC 10747 in order to support policy based and Mobile routing in the ATN; and the Route Initiation procedures for initiating the exchange of routing information using the ISO/IEC 10747 protocol. In the case of Air/Ground data links, route initiation also includes the use of the ISO/IEC 9542 protocol. This chapter is not concerned with compliancy with the ISO/IEC 10747 and ISO/IEC 9542 protocols. This is the subject of 5.8.

5.3.1.2 Applicability of Requirements

Note 1.— The classes of ATN Router referred to below are defined in 5.2.4.1.

Note 2.— The ATN RDs referred to below are defined in 5.2.2.2.

5.3.1.2.1 ATN Ground/Ground Routers shall comply with the provisions of 5.3.4 and 5.3.6.

5.3.1.2.2 When used as an ATN Router in an ATN RD that is a member of an ATN Island Backbone RDC, an ATN Ground/Ground Router shall also comply with the provisions of 5.3.7.1.

5.3.1.2.3 When used in any other ATN Transit Routing Domain, an ATN Ground/Ground Router shall also comply with the provisions of 5.3.7.3.

5.3.1.2.4 Otherwise, an ATN Ground/Ground Router shall comply with the provisions of 5.3.7.4.

5.3.1.2.5 ATN Air/Ground Routers shall comply with the provisions of 5.3.4 for Ground/Ground interconnection, 5.3.5 for Air/Ground interconnection and 5.3.6.

5.3.1.2.6 When used as an ATN Router in an ATN RD that is a member of an ATN Island Backbone RDC, an ATN Air/Ground Router shall also comply with the provisions of 5.3.7.1.

5.3.1.2.7 When used in any other ATN Transit Routing Domain, an ATN Air/Ground Router shall also comply with the provisions of 5.3.7.3.

5.3.1.2.8 ATN Airborne Routers shall comply with the provisions of 5.3.5, 5.3.6, and 5.3.7.2.

5.3.1.2.9 When an RD is declared to be an ATN RD, then it shall comply with the provisions of 5.2.2.2.

5.3.1.2.10 When an RD is declared to be a Mobile RD, then it shall comply with the provisions of 5.2.2.3.

5.3.1.2.11 When an RDC is declared to be an ATN Island RDC, then its member RDs shall comply with the provisions of 5.2.2.3.2.

5.3.1.2.12 When an RDC is declared to be an ATN Island Backbone RDC, then its member RDs shall comply with the provisions of 5.2.2.4.2.

5.3.2 Service Provided by an ATN Router

5.3.2.1 General

5.3.2.1.1 A route shall only be advertised by an ATN Router to an adjacent ATN RD when it can be ensured that data sent over that route by the RD to which the route is advertised is acceptable to every RD and RDC in the route's path, and will be relayed by them to the route's destination.

Note.— The acceptability of a route may be determined using a priori knowledge derived from interconnection agreements with other RDs.

5.3.2.2 Forwarding CLNP NPDUs

5.3.2.2.1 General

5.3.2.2.1.1 The forwarding processes for a CLNP NPDUs shall operate by selecting the FIB identified by the combination of the QoS Maintenance and Security Parameters found in the CLNP Header, and selecting from that FIB, the entry, if any, identified by the longest matching NSAP Address Prefix.

5.3.2.2.1.2 The next hop information found in this FIB entry shall then be used to forward the NPDUs.

Note.— Forwarding decisions that take into account the CLNP QoS Maintenance Parameter are a local matter and an ATN Router may hence ignore this parameter.

5.3.2.2.2 Forwarding a CLNP NPDUs when no Security Parameter is present in the PDU Header

Note.— This case applies for General Communications data (see 5.2.7.1).

5.3.2.2.2.1 When a CLNP NPDUs is received by an ATN Router and that NPDUs does not contain a Security Parameter in the PDU Header then that NPDUs shall be forwarded over the selected route to the NPDUs's destination with the longest matching NSAP Address Prefix and which, either:

- 1) contains a security path attribute comprising the ATN Security Registration Identifier and security information that does not contain an ATSC Class Security Tag indicating support for only ATSC traffic, and comprises:
 - a) either an Air/Ground Subnetwork Security Tag that has "General Communications" in its set of permissible Traffic Types, or
 - b) no Air/Ground Subnetwork Security Tag,
- or
- 2) does not contain any security path attribute.

5.3.2.2.2.2 If no such route can be found then the NPDUs shall be discarded.

5.3.2.2.3 Forwarding a CLNP NPDU when a Security Parameter is present in the PDU Header

5.3.2.2.3.1 General

5.3.2.2.3.1.1 When a CLNP NPDU is received by an ATN Router and that NPDU contains a Security Parameter in the Globally Unique Format, and encodes security related information according to 5.6.2.2 under the ATN Security Registration Identifier, then the NPDU shall be forwarded according to the procedures specified below.

Note 1.— The CLNP NPDU Header Security Parameter is used to indicate the Traffic Type of the application data contained in the NPDU, and the application's routing policy requirements.

Note 2.— The procedures for handling an NPDU with any other format of Security Parameter, or with any other Security Registration Identifier are outside the scope of this specification.

5.3.2.2.3.2 ATN Operational Communications Traffic Type - ATSC Traffic Category

Note.— In this case, either no Traffic Type policy preference may be specified, or an ATSC Class may be specified.

5.3.2.2.3.2.1 No Traffic Type Policy Preference

Note.— This case corresponds to a Traffic Type and Associated Routing Policy Security Tag value of 000 00001.

5.3.2.2.3.2.1.1 If the NPDU contains a CLNP NPDU Header Security Parameter in the globally unique format, and encodes:

- a) security related information according to 5.6.2.2 under the ATN Security Registration Identifier, and
- b) a traffic type of ATN Operational Communications and a traffic category of Air Traffic Service Communications, and
- c) no Traffic Type Policy Preference,

then the NPDU shall be forwarded over the selected route to the NPDU's destination with the longest matching NSAP Address Prefix, and which contains a security path attribute comprising the ATN Security Registration Identifier and security information that comprises:

- i. An Air/Ground Subnetwork Security Tag that has "ATN Operational Communications - Air Traffic Services Communications" in its set of permissible Traffic Types, or
- ii. no Air/Ground Subnetwork Security Tag,

and an ATSC Class Security Tag indicating support of the lowest class out of all such routes available.

Note 1.— The requirement in 5.3.2.2.1.1 always takes precedence over selection based on ATSC Class i.e. a route with a longer matching NSAP Address Prefix with a higher ATSC Class is always preferred over a route with a lower ATSC Class but with a shorter NSAP Address Prefix. This is essential for the avoidance of routing loops.

Note 2.— ATSC Class “H” is the lowest and Class “A” is the highest.

5.3.2.2.3.2.1.2 If no such route can be found, then the NPDU shall be discarded.

5.3.2.2.3.2.2 ATSC Class Specified

Note.— This case corresponds to Traffic Type and Associated Routing Policy Security Tag values 000 10000 to 000 10111 inclusive.

5.3.2.2.3.2.2.1 If the NPDU contains a CLNP Header Security Parameter in the globally unique format, and encodes:

- a) security related information according to 5.6.2.2 under the ATN Security Registration Identifier, and
- b) a traffic type of ATN Operational Communications and Air Traffic Service Communications traffic category, and
- c) a requirement to route the NPDU over a route of a specified ATSC Class,

then the NPDU shall be forwarded over the selected route to the NPDU's destination with the longest matching NSAP Address Prefix, and which contains a security path attribute comprising the ATN Security Registration Identifier and security information that comprises:

- i. An Air/Ground Subnetwork Security Tag that has “ATN Operational Communications - Air Traffic Services Communications” in its set of permissible Traffic Types, or
- ii. no Air/Ground Subnetwork Security Tag

and an ATSC Class Security Tag indicating:

- I. support of the required class, or a higher class, or
- II. if no such route is available then, the route with the highest ATSC Class available is chosen.

Note 1.— The requirement in 5.3.2.2.1.1 always takes precedence over selection based on ATSC Class i.e. a route with a longer matching NSAP Address Prefix with a higher ATSC Class is always preferred over a route with a lower ATSC Class but with a shorter NSAP Address Prefix. This is essential for the avoidance of routing loops.

Note 2.— ATSC Class “H” is the lowest and Class “A” is the highest.

5.3.2.2.3.2.2.2 If no such route can be found then the NPDU shall be discarded.

5.3.2.2.3.2.2.3 If multiple routes are available which meet or exceed the required ATSC Class, then the route with the lowest relative cost, i.e. actual monetary cost, shall be selected.

Note.— The actual monetary cost is determined through means outside the scope of this specification.

5.3.2.2.3.2.2.4 If the monetary cost is the same or unknown, then the hop count shall be used as the relative cost metric.

5.3.2.2.3.3 ATN Operational Communications Traffic Type - AOC Traffic Category

Note.— In this case, either no routing policy may be specified, or an Air/Ground Subnetwork type may be specified, or an Air/Ground subnetwork order of preference may be specified.

5.3.2.2.3.3.1 No Traffic Type Policy Preference

Note.— This case corresponds to a Traffic Type and Associated Routing Policy Security Tag value of 001 00001.

5.3.2.2.3.3.1.1 If the NPDU contains a CLNP Header Security Parameter in the globally unique format, and encodes:

- a) security related information according to 5.6.2.2 under the ATN Security Registration Identifier, and
- b) a traffic type of ATN Operational Communications and Aeronautical Operational Control traffic category, and
- c) no Traffic Type Policy Preference,

then the NPDU shall be forwarded over the selected route to the NPDU's destination with the longest matching NSAP Address Prefix, and which contains a security path attribute comprising the ATN Security Registration Identifier and security information that comprises:

- i. an Air/Ground Subnetwork Security Tag that has "ATN Operational Communications - Aeronautical Operational Control" in its set of permissible Traffic Types, or
- ii. no Air/Ground Subnetwork Security Tag;

and which does not contain an ATSC Class Security Tag indicating support for only ATSC traffic.

5.3.2.2.3.3.1.2 If no such route can be found, then the NPDU shall be discarded.

5.3.2.2.3.3.2 Air/Ground Subnetwork Type Specified

Note 1.— This case corresponds to Traffic Type and Associated Routing Policy Security Tag values 001 00010 through to 001 00110 inclusive.

Note 2.— The Air/Ground Subnetworks that may be specified are: Gatelink, VDL, AMSS, HF and Mode S.

5.3.2.2.3.3.2.1 If the NPDU contains a CLNP Header Security Parameter in the globally unique format, and encodes:

- a) security related information according to 5.6.2.2 under the ATN Security Registration Identifier, and
- b) a traffic type of ATN Operational Communications and Aeronautical Operational Control traffic category, and
- c) a requirement to route traffic only via a specific Air/Ground Subnetwork only,

then the NPDU shall be forwarded over the selected route to the NPDU's destination with the longest matching NSAP Address Prefix, and which contains a security path attribute comprising the ATN Security Registration Identifier and security information that comprises either

- i. an Air/Ground Subnetwork Security Tag that indicates that the route passes over that Air/Ground Subnetwork and has "ATN Operational Communications — Aeronautical Operational Control" in its set of permissible Traffic Types, or,
- ii. no Air/Ground Subnetwork Security Tag,

and which does not contain an ATSC Class Security Tag indicating support for only ATSC traffic.

5.3.2.2.3.3.2.2 If no such route can be found, then the NPDU shall be discarded.

5.3.2.2.3.3.3 Air/Ground Subnetwork Order of Preference Specified

Note 1.— This case corresponds to Traffic Type and Associated Routing Policy Security Tag values 001 00111 through to 001 01001 inclusive.

Note 2.— The Air/Ground Subnetworks for which an order of preference may be specified are: Gatelink, VDL, AMSS, HF and Mode S.

5.3.2.2.3.3.3.1 If the NPDU contains a CLNP Header Security Parameter in the globally unique format, and encodes:

- a) security related information according to 5.6.2.2 under the ATN Security Registration Identifier, and
- b) a traffic type of ATN Operational Communications and Aeronautical Operational Control traffic category, and

- c) a requirement to route traffic only via certain Air/Ground Subnetworks and with a specified order of preference,

then the NPDU shall be forwarded over the selected route to the NPDU's destination with the longest matching NSAP Address Prefix, and which contains a security path attribute comprising the ATN Security Registration Identifier and security information that comprises:

- i. an Air/Ground Subnetwork Security Tag that indicates that the route passes over the first preference Air/Ground Subnetwork and has "ATN Operational Communications - Aeronautical Operational Control" in its set of permissible Traffic Types, if present, or
- ii. an Air/Ground Subnetwork Security Tag that indicates that the route passes over the second preference Air/Ground Subnetwork and has "ATN Operational Communications - Aeronautical Operational Control" in its set of permissible Traffic Types, if present, and so on until a suitable route is found or no further preferences are specified, or
- iii. no Air/Ground Subnetwork Security Tag,

and which does not contain an ATSC Class Security Tag indicating support for only ATSC traffic.

5.3.2.2.3.3.3.2 If no such route can be found, then the NPDU shall be discarded.

5.3.2.2.3.3.3.3 If after applying the above procedures, a more specific route is available to the NPDU's destination, but

- 1) the route has an Air/Ground Subnetwork Security Tag that indicates that the route passes over a lower preference Air/Ground Subnetwork while
- 2) having "ATN Operational Communications - Aeronautical Operational Control" in its set of permissible Traffic Types, then
- 3) the more specific route shall be selected in preference to the less specific route.

Note.— The purpose of this requirement is to ensure that the NPDU is not forced to visit a default route provider only to find that a higher preference route does not actually exist to the NPDU's destination.

5.3.2.2.3.4 ATN Administrative Communications Traffic Type

Note.— This case corresponds to a Traffic Type and Associated Routing Policy Security Tag value of 001 10000.

5.3.2.2.3.4.1 If the NPDU contains a CLNP Header Security Parameter in the globally unique format, and encodes:

- a) security related information according to 5.6.2.2 under the ATN Security Registration Identifier, and

- b) a traffic type of ATN Administrative Communications,

then the NPDU shall be forwarded over the selected route to the NPDU's destination with the longest matching NSAP Address Prefix, and which contains a security path attribute comprising the ATN Security Registration Identifier and security information that comprises:

- i. either an Air/Ground Subnetwork Security Tag that has "ATN Administrative Communications" in its set of permissible Traffic Types, or
- ii. no Air/Ground Subnetwork Security Tag

and which does not contain an ATSC Class Security Tag indicating support for only ATSC traffic.

5.3.2.2.3.4.2 If no such route can be found, then the NPDU shall be discarded.

5.3.2.2.3.5 ATN Systems Management Communications Traffic Type

Note.— This case corresponds to a Traffic Type and Associated Routing Policy Security Tag value of 011 00000.

5.3.2.2.3.5.1 If the NPDU contains a CLNP Header Security Parameter in the globally unique format, and encodes:

- a) security related information according to 5.6.2.2 under the ATN Security Registration Identifier, and
- b) a traffic type of ATN Systems Management Communications,

then the NPDU shall be forwarded over the selected route to the NPDU's destination with the longest matching NSAP Address Prefix, and which:

- 1) contains a security path attribute comprising the ATN Security Registration Identifier and security information that comprises:
 - a) either an Air/Ground Subnetwork Security Tag that has "ATN Systems Management Communications" in its set of permissible Traffic Types, or
 - b) no Air/Ground Subnetwork Security Tag,

or

- 2) contains no security path attribute.

5.3.2.2.3.5.2 If no such route can be found, then the NPDU shall be discarded.

5.3.3 The Deployment of ATN Components

5.3.3.1 Interconnection of ATN RDs

5.3.3.1.1 General

5.3.3.1.1.1 ATN RDs shall be interconnected by real subnetworks permitting communication between ATN Routers for each of the interconnection scenarios specified below.

Note 1.— Examples of possible interconnections between ATN Routing Domains are illustrated in Figure 5.2-1.

Note 2.— There is no requirement for all ATN RDs to be fully interconnected.

5.3.3.1.1.2 Except for the interconnection of Mobile RDs with other ATN RDs, the real subnetwork(s) used for such an interconnection shall be chosen by bilateral agreement and may be any subnetwork that complies with the provisions of 5.2.5.1.

Note 1.— For example, the chosen subnetwork may be a point-to-point communications link, a public or private PSDN providing the CCITT X.25 network access service, an Ethernet or an ISDN, etc.

Note 2.— The dynamic procedures for the interconnection of two ground-based ATN Routers are specified in 5.3.4, and for interconnection of an Air/Ground and an Airborne Router in 5.3.5. The remainder of this section is concerned with static interconnection requirements.

5.3.3.1.2 Interconnection between Members of an ATN Island Backbone RDC

5.3.3.1.2.1 When there is more than one ATN RD in an ATN Island Backbone RDC, each Administration or aeronautical industry member that has elected to participate in that ATN Island's Backbone RDC, shall ensure that its RD is either:

- a) interconnected directly with all other ATN RDs within the ATN Island's Backbone RDC, over suitable and mutually agreeable real subnetwork(s), or
- b) interconnected directly as in a), with one or more ATN RDs that are also members of the ATN Island's Backbone RDC, and which are able and willing to provide routes to the remaining RDs within the Backbone RDC.

Note.— The existence of the ATN Backbone RDC prohibits routes between its member RDs via other ATN RDs in the same ATN Island.

5.3.3.1.3 Interconnection between Members of an ATN Island Backbone RDC and other ATN RDs within the ATN Island

5.3.3.1.3.1 ATN RDs within an ATN Island RDC that are not members of the ATN Island's Backbone RDC, shall ensure that they are either:

- a) interconnected directly with one or more ATN RDs that are members of the ATN Island's Backbone RDC, over suitable and mutually agreeable real subnetworks; or

- b) interconnected with one or more other ATN RDs that are members of the same ATN Island RDC and which are able and willing to provide routes to and from one or more ATN RDs within the same ATN Island's Backbone RDC, and to all destinations reachable via the ATN Island's Backbone RDC.

5.3.3.1.4 Interconnection of ATN Islands

5.3.3.1.4.1 ATN Islands shall only interconnect via ATN RDs which are members of each ATN Island's Backbone RDC.

5.3.3.1.4.2 When an ATN RD is a member of more than one ATN Island RDC, its routing policy shall not permit it to operate as a TRD between sources and destinations in different ATN Islands unless the RD is a member of each Island's Backbone RDC.

5.3.3.1.5 Interconnection of Mobile and Fixed RDs

Note.— A Mobile RD may interconnect concurrently with multiple ATN RDs which are attached to the common Mobile Subnetworks and which are accessible to the Mobile RD at any given time. The purpose of such interconnections is to provide data link communications services when required by ATN Data Link applications and other aeronautical or airline industry applications.

5.3.3.1.5.1 In order to meet the availability requirements of ATN Data Link applications, Airborne and Air/Ground Routers shall be capable of supporting multiple concurrent adjacencies with other Routers.

Note 1.— These adjacencies are supported by multiple subnetwork connections at the same or different priorities, using the same or different Air/Ground subnetworks.

Note 2.— Dynamically, such adjacencies may be established and released in a « make before break » fashion permitting continuous communications availability, and for the suitability of a newly available adjacency to be determined before a no longer needed adjacency is released.

Note 3.— It is not within the scope of this specification to set minimum requirements in respect of the number of adjacencies and subnetwork connections that an Airborne or Air/Ground Router must support. Such requirements are dependant on the published coverage and number of Air/Ground subnetworks, application availability requirements and additionally, in the case of Airborne Routers, on Airline operating policies. Implementors are advised to interpret « multiple » as, in the context of the above requirement, implying at least two adjacencies or connections, and, in practice, a larger number is anticipated as being the likely minimum requirement.

5.3.3.1.6 Interconnection of ATN RDs and non-ATN RDs

Note.— ATN RDs may interconnect with non-ATN RDs whether they are members of the same Administrative Domain or not.

5.3.4 Ground/Ground Interconnection

5.3.4.1 Interconnection Scenarios

Note 1.— Ground/Ground interconnection procedures apply to the interconnection of two Ground/Ground Routers, and to the interconnection of an Air/Ground Router and a Ground/Ground Router.

Note 2.— Formally, these procedures only apply to interconnection between ATN Routers in different Administrative Domains. However, in practice, they are also applicable to interconnection scenarios within the same Administrative Domain.

5.3.4.2 Ground/Ground Route Initiation

Note.— Route Initiation is defined to be the point at which routing information exchange can begin, and the route initiation procedures are those that permit the exchange of routing information to commence.

5.3.4.2.1 When the network administrators agree to the ground/ground interconnection of one or more ATN Routers within their respective Administrative Domains, they shall:

- a) Make available suitable subnetwork connectivity including, where necessary the physical installation of suitable communications equipment for end-to-end communications between the ATN Routers, and supporting the Quality of Service necessary for the applications data that will be routed over this interconnection.

Note.— The choice of appropriate subnetwork(s) to support the interconnection is a matter for bilateral agreement between network administrators, including agreement on responsibility for installation, operating and maintenance costs, and fault management.

- b) Using global or local Systems Management mechanisms, establish one or more subnetwork connections between the two ATN Routers, unless the subnetwork technology is connectionless in which case this step may be omitted.

Note 1.— Typically (e.g. with X.25), one ATN Router will be placed in a state where it will accept an incoming connection from the other ATN Router, and then the other ATN Router is stimulated to initiate one or more subnetwork connection(s) to the other ATN Router.

Note 2.— Multiple concurrent subnetwork connections over the same or different subnetworks may be required in order to meet throughput and other QoS requirements.

- c) Using global or local Systems Management mechanisms, ensure that the forwarding information base in each ATN Router, used to support the connectionless network protocol specified in 5.6, contains sufficient information to forward CLNP NPDUs addressed to the NET of the other ATN Router, over the newly established subnetwork connection(s).

Note.— This step is necessary to ensure that the connectionless network service can be used to exchange the BISPDUs of IDRP.

- d) Using global or local Systems Management mechanisms, append the NET of the remote ATN Router to the **externalBISNeighbor** attribute of the BIS's **idrpConfig MO**,

- e) Using global or local Systems Management mechanisms, create an **AdjacentBIS** Managed Object (MO) in each ATN Router to represent the other ATN Router, and
- f) Using global or local Systems Management mechanisms, invoke the start event action on each such MO, in order to initiate a BIS-BIS connection between the two ATN Routers.

Note.— As a matter for the bilateral agreement of the network administrators, either (a) both ATN Routers will attempt to open the BIS-BIS connection, or (b) one and only one acts as the BIS-BIS connection initiator.

5.3.4.3 Ground/Ground Routing Information Exchange

5.3.4.3.1 Routing information shall be exchanged using the ISO/IEC 10747 Inter-Domain Routing Protocol according to the profile specified in 5.8.

5.3.4.3.2 In support of Air/Ground communications, the exchange of routing information shall be subject to appropriate routing policies specified in 5.3.7.1, 5.3.7.3, or 5.3.7.4, depending upon the role of the Routing Domain in which each ATN Router is located.

5.3.4.4 Ground/Ground Route Termination

Note 1.— Route Termination is defined to be the point at which routing information ceases to be exchanged between two ATN Routers, and, in consequence, the routes made available over the adjacency cease to be useable and must be withdrawn. The route termination procedures are those procedures which terminate the exchange of routing information.

Note 2.— Route Termination may result from a failure in the underlying subnetwork(s) causing a loss of communication between the two ATN Routers. Alternatively, it may result from a deliberate decision of network administrators to terminate the interconnection, either temporarily or permanently.

Note 3.— No special recovery procedures are specified if route termination is due to a network fault. Once the fault has been repaired, the procedures of 5.3.4.2 may be re-invoked, as appropriate to re-establish communication, and to exchange routing information.

5.3.4.4.1 When a network administrator decides to temporarily or permanently terminate an interconnection between two ATN Routers then, using global or local Systems Management mechanisms applied to either or both of the two ATN Routers, the deactivate action shall be invoked on the AdjacentBIS MO that represents the remote ATN Router with which the BIS-BIS connection is to be terminated.

5.3.4.4.2 If the adjacency is to be permanently terminated, then the AdjacentBIS MO shall also be deleted, and the forwarding information base shall be updated to remove the route to the NET of the remote ATN Router.

5.3.4.4.3 For either temporary or permanent termination, and if required, by using global or local Systems Management mechanisms, the network administrator(s) shall also terminate the underlying subnetwork connections.

5.3.5 Air/Ground Interconnection

5.3.5.1 Interconnection Scenarios

Note 1.— Air/Ground interconnection applies to the interconnection between an ATN Airborne Router and an ATN Air/Ground Router over one or more Mobile Subnetworks.

Note 2.— The significant difference between Air/Ground and Ground/Ground Interconnection is that in the former case interconnection is automatic and consequential on the availability of communications and local policy, while, in the latter case, interconnection is a deliberate and planned action with the direct involvement of network administrators.

Note 3.— While IDRP is also intended to be used over Air/Ground Interconnections, as an interim measure, the optional non-use of IDRP over Air/Ground Interconnections is permitted by this specification and according to 5.3.5.2.12.

Note 4.— For the purposes of this specification, the functional model of an ATN Router illustrated in Figure 5.3-1 is assumed. This model illustrates the basic functional entities in an ATN Air/Ground (Class 5 Router) and ATN Airborne Router with IDRP (Class 6 Router), the data flow between them as solid lines, and the flow of certain events and control information, by dashed lines.

Note 5.— Figure 5.3-1 introduces a new architectural entity, the Intermediate System - Systems Management Entity (IS-SME). As specified below, this plays an important role in the realisation of Route Initiation in Air/Ground Operations, by responding to changes in subnetwork connectivity and thereby controlling the route initiation and termination procedures.

Note 6.— The ATSC Class assigned to an Air/Ground Subnetwork and the traffic type(s) allowed to pass over this Air/Ground Subnetwork are known a priori to the Air/Ground Router attached to each such subnetwork. They are communicated to an Airborne Router using the options part of an ISO/IEC 9542 ISH PDU which is uplinked to the Airborne Router as part of the route initiation procedure as described in 5.3.5.2.

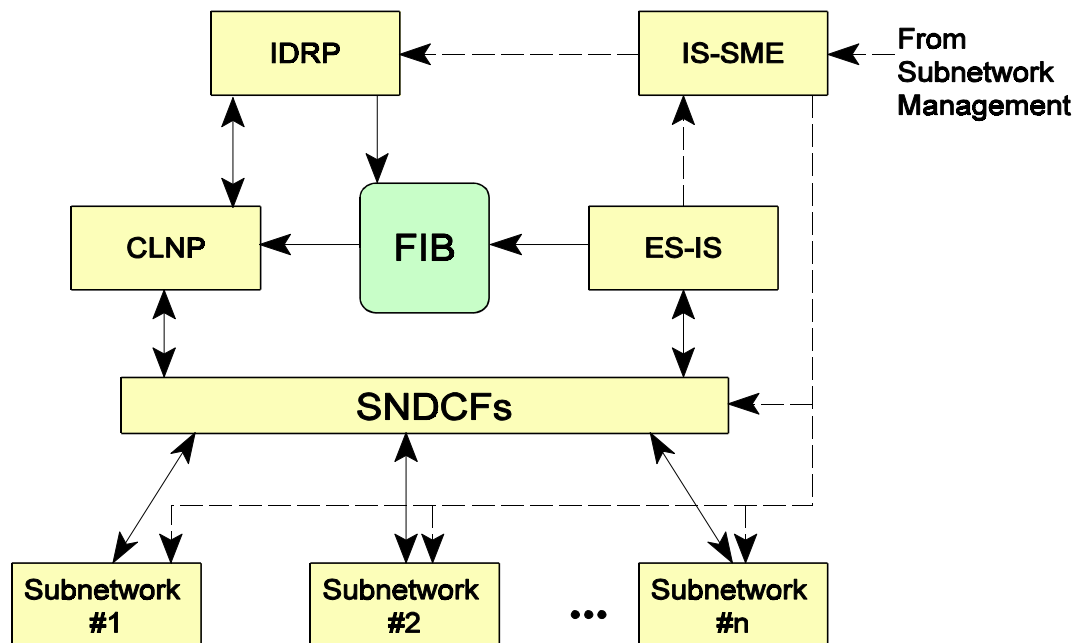


Figure 5.3-1 Assumed ATN Router Architecture for Air/Ground Route Initiation

5.3.5.2 Air/Ground Route Initiation

5.3.5.2.1 General

5.3.5.2.1.1 BIS-BIS communications over a Mobile Subnetwork shall be either air-initiated or ground-initiated, with one of these two modes of operation selected for all instances of a given subnetwork type.

Note 1.— Three classes of procedures are distinguished by this specification. These are: (a) Air-Initiated i.e. when the Airborne Router initiates the procedure, (b) Ground-Initiated i.e. when the Air/Ground Router initiates the procedure, and (c) Air or Ground-Initiated i.e. when either the Airborne or the Air/Ground Router may initiate the procedure.

Note 2.— Two types of Mobile Subnetworks are also recognised by this specification. These are: (a) those which provide information on the availability of specific Mobile Systems on the subnetwork through the Join Event defined in this section, and (b) those which do not. The latter type are only appropriate to Route Initiation Procedures which are Air-Initiated.

Note 3.— For a given Mobile Subnetwork type, the use of air-initiated or ground-initiated procedures, and the implementation of Join Events is outside of the scope of this specification, and is a matter for the SARPs specified by the relevant ICAO panel.

Note 4.— The interfaces to all Mobile Subnetworks are assumed to be compatible with ISO/IEC 8208. The ISO/IEC 8208 term Data Terminal Equipment (DTE) is also used in this specification to refer to a system attached to a Mobile Subnetwork.

5.3.5.2.1.2 For Air-Initiated Subnetworks that do not provide information on the availability of specific Mobile Systems, Airborne Routers shall comply with the procedures specified in 5.3.5.2.3.1, and Air/Ground Routers shall comply with the procedures specified in 5.3.5.2.2.

5.3.5.2.1.3 For Air-Initiated Subnetworks that do provide information on the availability of specific Mobile Systems, Airborne Routers shall comply with the procedures specified in 5.3.5.2.3.2, and Air/Ground Routers shall comply with the procedures specified in 5.3.5.2.2.

5.3.5.2.1.4 For Ground-Initiated Subnetworks, Air/Ground Routers shall comply with the procedures specified in 5.3.5.2.4, and Airborne Routers shall comply with the procedures specified in 5.3.5.2.2.

5.3.5.2.1.5 For Air or Ground-Initiated Subnetworks, Air/Ground and Airborne Routers shall comply with the procedures specified in 5.3.5.2.2 and 5.3.5.2.5.

5.3.5.2.2 Route Initiation Procedures for a Responding ATN Router

5.3.5.2.2.1 General

Note 1.— Route Initiation is always asymmetric with a clearly defined initiator and responder. In all cases, the ATN Router in the responder role, follows the same procedures, as specified below.

Note 2.— For Air-Initiated Route Initiation, the Air/Ground Router is the responding ATN Router. For Ground-Initiated Route Initiation, the Airborne Router is the responding ATN Router.

5.3.5.2.2.1.1 Each ATN Router that is specified to take the responder role for a given Mobile Subnetwork type, and when attached to a subnetwork of that subnetwork type, shall be configured into a state whereby it “listens” for Call Indications on that subnetwork.

5.3.5.2.2.1.2 For each Call Indication received, a responding ATN Router shall, based on local policy, either:

- a) Accept the incoming call immediately using a Call Accept Packet, or
- b) Validate the calling DTE address and either accept the call using a Call Accept Packet, or if the call is unacceptable then it shall be rejected using a Clear Request Packet.

Note 1.— The procedures used to validate the calling DTE address and to determine the acceptability of the call, are outside the scope of this specification.

Note 2.— The number of simultaneous virtual circuits that the ATN Router needs to support will be subject to an implementation limit, that needs to be sufficient for the role in which the ATN Router is deployed.

5.3.5.2.2.1.3 When a subnetwork connection is successfully established, then the procedures of 5.3.5.2.6 shall be applied to that subnetwork connection.

5.3.5.2.3 Air-Initiated Route Initiation

Note.— This section specifies the procedures to be used by an Airborne Router for Air-Initiated route initiation.

5.3.5.2.3.1 Airborne Router Procedures for use of an ISO/IEC 8208 Mobile Subnetwork that does not Provide Information on Subnetwork Connectivity

5.3.5.2.3.1.1 General

5.3.5.2.3.1.1.1 An Airborne Router's IS-SME shall be configured with a list of subnetwork addresses for each supported Mobile Subnetwork that does not provide information on subnetwork connectivity.

5.3.5.2.3.1.1.2 This list shall include the addresses which are necessary to meet the communication needs of the aircraft.

Note.— In the case of the AMSS, the Airborne Router's IS-SME will be configured with a list for each GES that the aircraft may use to communicate. Each such list will include the subnetwork addresses (e.g. DTE addresses) of the Air/Ground routers attached to the GES in question through which communications services may be required.

5.3.5.2.3.1.1.3 An Airborne Router's IS-SME shall continually issue a Call Request to each subnetwork address on each appropriate list with which it does not currently have a subnetwork connection and which is not subject to a back-off period (see 5.3.5.2.3.1.2), in turn.

5.3.5.2.3.1.1.4 The period between each successive Call Request shall be configurable to ensure that the Mobile Subnetwork is not rendered unavailable.

5.3.5.2.3.1.1.5 When a subnetwork connection is successfully established, then the procedures of 5.3.5.2.6 shall be applied to that subnetwork connection. The polling procedure shall continue for the remaining subnetwork addresses on the list, if any.

5.3.5.2.3.1.2 Call Request Failure

5.3.5.2.3.1.2.1 Whenever a Clear Indication is received in response to a Call Request that indicates rejection by the called DTE and includes a call clearing diagnostic code of 0, 133, 160..163, or 240, 241, 242, 244, 246, 248, then the Airborne Router shall implement a "back off" procedure.

5.3.5.2.3.1.2.2 The back off procedure shall comprise the effective quarantining of the called subnetwork address for a period configurable on a per subnetwork basis from 5 minutes to 20 minutes. During this period, a Call Request shall not be issued to the subnetwork address.

Note.— The purpose of the back off procedure is to avoid unnecessarily overloading of the Air/Ground subnetwork with Call Requests.

5.3.5.2.3.1.2.3 The “back off” procedure shall not be started on receipt of a Clear Indication which includes any other call clearing diagnostic code.

5.3.5.2.3.1.2.4 If a Clear Indication is received with a diagnostic code reporting an error that the SND CF is unable to correct, then the called DTE shall be removed from the polled DTEs list.

5.3.5.2.3.1.2.5 Otherwise, if required, the SND CF shall retry the call after having resolved the cause of the call rejection.

Note.— Certain call clearing diagnostic codes in the range 128..143 are used by the Mobile SND CF specified in 5.7. The semantics of these codes are described in Table 5.7-3.

5.3.5.2.3.1.2.6 However, during any period when an Airborne Router does not have any subnetwork connections over Mobile Subnetworks, then all back off procedures shall be suspended until connectivity is established with at least one Air/Ground Router.

5.3.5.2.3.2 Airborne Router Procedures for use of an ISO/IEC 8208 Mobile Subnetwork that does Provide Connectivity Information

Note 1.— The connectivity information is provided as a “Join Event”. This is an event generated by a Mobile Subnetwork when it is recognised that a system has attached to the subnetwork and is available for communication using the subnetwork. The Join Event provides the DTE Address of the newly available system. It may also include other subnetwork specific information needed to route a call to that DTE Address. For example, in the case of the VDL subnetwork, the call may need to be directed via a specific Ground Station and hence the Ground Station Address must be provided in addition to the DTE Address.

Note 2.— An actual implementation of a Join Event may concatenate several distinct Join Events as defined above into a single message.

Note 3.— For air-initiated subnetworks, the Join Event is received by the IS-SME in the Airborne Router. The mechanism by which it is received is both subnetwork and implementation dependent and is outside of the scope of this specification.

5.3.5.2.3.2.1 On receipt of a Join Event, the Airborne Router shall either:

- a) Issue an ISO/IEC 8208 Call Request with the DTE Address reported by the Join Event as the Called Address, or
- b) Validate the DTE Address reported by the Join Event as to whether or not a subnetwork connection with it is acceptable according to local Routing Policy. If such a connection is acceptable then an ISO/IEC 8208 Call Request shall be issued with the DTE Address reported by the Join Event as the Called Address. Otherwise, the Join Event shall be ignored.

Note.— The Airborne Router validates the DTE Address that is the subject of the Join Event and determines the acceptability of a subnetwork connection with the so identified ATN Router, using procedures outside of the scope of this specification.

5.3.5.2.3.2.2 On receipt of a Call Connected packet, and if the Called Line Address Modified Notification optional user facility is used in the received packet and indicates that the returned Called Address is different from that used in the Call Request packet, and the subnetwork also generates “Handoff” events (see 5.3.5.2.14), then the IS-SME shall store the relationship between the originally called DTE Address and the returned Called Address in the same local database. The knowledge of this relationship shall be retained as long as a subnetwork connection exists with the DTE.

5.3.5.2.3.2.3 When a subnetwork connection is successfully established, then the procedures of 5.3.5.2.6 shall be applied to that subnetwork connection.

5.3.5.2.4 Ground-Initiated Route Initiation

Note 1.— Ground-Initiated Route Initiation is only appropriate for Mobile subnetworks that originate a Join Event from their ground component.

Note 2.— For ground-initiated subnetworks, the Join Event is received by the IS-SME in the Air/Ground Router. The mechanism by which it is received is both subnetwork and implementation dependent and is outside of the scope of this specification.

5.3.5.2.4.1 On receipt of a Join Event, the Air/Ground Router shall either:

- a) Issue an ISO/IEC 8208 Call Request with the DTE Address reported by the Join Event as the Called Address, or
- b) Validate the DTE Address reported by the Join Event as to whether or not a subnetwork connection with it is acceptable according to local Routing Policy. If such a connection is acceptable then an ISO/IEC 8208 Call Request shall be issued with the DTE Address reported by the Join Event as the Called Address. Otherwise, the Join Event shall be ignored.

Note.— Option (b) above permits an administration or organisation operating a ground-initiated Mobile Subnetwork to implement procedures, according to its local policy, whereby an Air/Ground Router may validate the DTE that is the subject of the Join Event and hence determine the acceptability of a subnetwork connection with the so identified Airborne Router. The purpose of this facility is to enable efficient management of the available subnetwork resources in areas of overlapping coverage. This facility is not appropriate when its use may result in an aircraft being denied Air/Ground data communications.

5.3.5.2.4.2 On receipt of a Call Connected packet, and if the Called Line Address Modified Notification optional user facility is used in the received packet and indicates that the returned Called Address is different from that used in the Call Request, and the subnetwork also generates “Handoff” events (see 5.3.5.2.14), then the IS-SME shall store the relationship between the originally called DTE Address and the returned Called Address in the same local database. The knowledge of this relationship shall be retained as long as a subnetwork connection exists with the DTE.

5.3.5.2.4.3 When a subnetwork connection is successfully established, then the procedures of 5.3.5.2.6 shall be applied to that subnetwork connection.

5.3.5.2.5 Air or Ground-Initiated Route Initiation

Note 1.— Air or Ground-Initiated Route Initiation is only appropriate for mobile subnetworks that do provide connectivity information through a Join Event to the Airborne or Air/Ground Router, or both.

Note 2.— For Air or Ground-Initiated subnetworks, the Join Event is received by the IS-SME in the Airborne or Air/Ground Router, respectively. The mechanism by which it is received is both subnetwork and implementation dependent.

5.3.5.2.5.1 On receipt of a Join Event, the ATN Router shall either:

- a) Issue an ISO/IEC 8208 Call Request with the DTE Address reported by the Join Event as the Called Address, or
- b) Validate the DTE reported by the Join Event as to whether or not a subnetwork connection with it is acceptable according to local Routing Policy. If such a connection is acceptable then an ISO/IEC 8208 Call Request shall be issued with the DTE Address reported by the Join Event as the Called Address. Otherwise, the Join Event shall be ignored.

Note.— The ATN Router validates the DTE Address that is the subject of the Join Event and determines the acceptability of a subnetwork connection with the so identified ATN Router, using procedures outside of the scope of this specification.

5.3.5.2.5.2 On receipt of a Call Connected packet, and if the Called Line Address Modified Notification optional user facility is used in the received packet and indicates that the returned Called Address is different from that used in the Call Request, and the subnetwork also generates “Handoff” events (see 5.3.5.2.14), then the IS-SME shall store the relationship between the originally called DTE Address and the returned Called Address in the same local database. The knowledge of this relationship shall be retained as long as a subnetwork connection exists with the DTE.

5.3.5.2.5.3 When a subnetwork connection is successfully established, then the procedures of 5.3.5.2.6 shall be applied to that subnetwork connection.

Note.— When a call collision occurs, then the call collision resolution procedures are applied by the SNDCF in order to ensure that only a single virtual circuit is established and that connection initiator and responder are unambiguously identified.

5.3.5.2.6 Exchange of Configuration Information using the ISO/IEC 9542 ISH PDU

5.3.5.2.6.1 ATN Airborne and Air/Ground Routers shall implement the ISO/IEC 9542 “Configuration Information” Function for use over each Mobile Subnetwork that they support.

5.3.5.2.6.2 Whenever a subnetwork connection is established over a Mobile Subnetwork, the ISO/IEC 9542 “Report Configuration” Function shall be invoked in order to send an ISH PDU containing the NET of the Airborne or Air/Ground Router network entity over the subnetwork connection.

5.3.5.2.6.3 In the case of an Airborne Router, if it supports the use of IDRP for the exchange of routing information over this subnetwork, then the SEL field of the NET inserted into the ISH PDU shall always be set to 00h (i.e. a binary pattern of all zeroes).

5.3.5.2.6.4 Alternatively, if the Airborne Router implements the procedures for the optional non-use of IDRP over this subnetwork, then the SEL field of the NET inserted into the ISH PDU shall always be set to FEh (i.e. a binary pattern of 1111 1110).

5.3.5.2.6.5 An ATN Air/Ground Router shall include the Mobile Subnetwork Capabilities Parameter, as defined in 5.8.2.1.3, in the options part of the uplinked ISH PDU. The Mobile Subnetwork Capabilities Parameter shall indicate any restrictions on traffic types permitted to pass over the Mobile Subnetwork and the ATSC Class of the Mobile Subnetwork, if the ATN Operational Communications traffic type - Air Traffic Service Communications traffic category is among the permissible traffic types for this Mobile Subnetwork.

Note 1.— The ATSC Class assigned to a Mobile Subnetwork and the traffic type(s) allowed to pass over this Mobile Subnetwork are uplinked to the Airborne Router to enable this router to make the appropriate routing decision when downlinking packets over an air/ground adjacency which is made up of more than one Mobile Subnetwork.

Note 2.— The ISH PDU is only ever sent in the context of a single Mobile Subnetwork between the Air/Ground and Airborne Router. Thus the capability information carried in the Mobile Subnetwork Capabilities Parameter is unambiguously associated with this subnetwork.

5.3.5.2.6.6 **Recommendation.**— *When in the initiator role, an ATN Router should use the ISO/IEC 8208 “Fast Select” facility, if supported by the subnetwork, and encode the first ISH PDU in the Call Request user data, according to the procedures for the Mobile SND CF specified in Chapter 5.7.*

5.3.5.2.6.7 **Recommendation.**— *When in the responder role and the initiator has proposed use of the Fast Select Facility, the ATN Router should encode the first ISH PDU in the Call Accept user data, according to the procedures for the Mobile SND CF specified in Chapter 5.7.*

Note.— The purpose of encoding an ISH PDU in call request or call accept user data is to minimise the number of messages sent over limited bandwidth Mobile Subnetworks and the time taken to complete the route initiation procedures.

5.3.5.2.6.8 Whenever an ISO/IEC 9542 ISH PDU is received by an Airborne Router, this router shall evaluate the Mobile Subnetwork Capabilities Parameter contained in the options part of the received ISH PDU.

5.3.5.2.6.9 The Airborne Router shall use the received subnetwork capability information to update its local configuration data concerning the permissible traffic type(s) and the supported ATSC Class of the Mobile Subnetwork over which the ISH PDU was received.

5.3.5.2.6.10 Whenever an ISO/IEC 9542 ISH PDU is received, either as Call Request or Call Accept user data, or as data sent over the connection, the ISO/IEC 9542 Record Configuration function shall be invoked and the routing information necessary for NPDUs to be sent over the subnetwork connection shall be written into the Forwarding Information Base for use by ISO/IEC 8473.

5.3.5.2.6.11 A Systems Management notification shall be generated to report the arrival of the ISH PDU to the ATN Router's IS-SME.

5.3.5.2.7 Validation of the Received NET

5.3.5.2.7.1 The IS-SME shall, using the received NET to identify the remote ATN Router, validate the acceptability of a BIS-BIS connection with that remote ATN Router.

5.3.5.2.7.2 If a BIS-BIS connection is unacceptable, then a Clear Request shall be generated to terminate the subnetwork connection. Forwarding Information associated with the subnetwork connection shall be removed from the Forwarding Information Base.

Note.— The acceptability of a BIS-BIS connection with the ATN Router identified by the received NET is determined using procedures outside of the scope of this specification.

5.3.5.2.7.3 If a BIS-BIS Connection is acceptable then an Air/Ground Router shall apply the procedures of 5.3.5.2.8, and an Airborne Router shall apply the procedures of 5.3.5.2.9.

5.3.5.2.8 Determination of the Routing Information Exchange Procedure by an Air/Ground Router

5.3.5.2.8.1 When the arrival of the ISH PDU is reported to the IS-SME of an Air/Ground Router, then the SEL field of the NET contained in this ISH PDU shall be inspected:

- a) If the SEL field takes the value of 00h (i.e. a binary pattern of all zeroes), then this shall be taken to imply that the Airborne Router that sent this ISH PDU supports the use of IDRP for the exchange of routing information. The procedures of 5.3.5.2.10 shall be applied.
- b) If the SEL field takes the value of FEh (i.e. a binary pattern of 1111 1110), then this shall be taken to imply that the Airborne Router that sent this ISH PDU supports the procedures for the optional non-use of IDRP for the exchange of routing information. The procedures of 5.3.5.2.12 shall be applied.

5.3.5.2.9 Determination of the Routing Information Exchange Procedure by an Airborne Router

5.3.5.2.9.1 When the arrival of the ISH PDU is reported to the IS-SME of an Airborne Router, then if the Airborne Router supports the use of IDRP for the exchange of routing information, then the procedures of 5.3.5.2.10 shall be applied. If the Airborne Router supports the procedures for the optional non-use of IDRP for the exchange of routing information, then the procedures of 5.3.5.2.12 shall be applied.

5.3.5.2.10 Establishment of a BIS-BIS Connection

5.3.5.2.10.1 The IS-SME shall append the NET received on the ISH PDU to the externalBISNeighbor attribute of the BIS's idrpConfig Managed Object, if not already present, and create an adjacentBIS Managed Object for the remote ATN Router identified by this NET, if one does not already exist.

5.3.5.2.10.2 If the ISH PDU was received from a subnetwork connection which was established with the local ATN Router in the responder role, then an IDRP activate action shall be invoked to start the BIS-BIS connection according to ISO/IEC 10747, if such a BIS-BIS connection does not already exist.

5.3.5.2.10.3 If the ISH PDU was received from a subnetwork connection which was established with the local ATN Router in the initiator role, then no IDRPs activate action shall be invoked, but the ListenForOpen MO attribute shall be set to true if a BIS-BIS connection does not already exist.

Note.— This procedure minimises the route initiation exchanges over a bandwidth limited Mobile Subnetwork. The reversal of initiator and responder roles for the BIS-BIS connection compared with the subnetwork connection ensures the fastest route initiation procedure.

5.3.5.2.10.4 If a BIS-BIS connection was already established with the remote ATN Airborne Router, then the IS-SME of the Air/Ground Router shall cause

- a) the update of the Security path attribute's security information of all routes contained in the Adj-RIB-In associated with the remote ATN Airborne Router, and
- b) the IDRPs Routing Decision function to be invoked in order to rebuild the FIB, the Loc_RIB and relevant Adj-RIB-Out(s) taking into account the additional subnetwork connectivity.

5.3.5.2.10.5 If a BIS-BIS connection was already established with the remote ATN Air/Ground Router, then the IS-SME of the Airborne Router shall cause the IDRPs Routing Decision Function to be invoked in order to rebuild the FIB, the Loc-RIB and relevant Adj-RIB-Out(s) taking into account the additional subnetwork connectivity.

5.3.5.2.10.6 Furthermore, the Air/Ground Router shall re-advertise all routes affected by the change in subnetwork connectivity that are contained in the Adj-RIB-Out associated with the remote ATN Airborne Router subsequent to the update of the security path attribute's security information of these routes as specified in 5.8.

5.3.5.2.10.7 The IS-SME shall also ensure that the procedures for the optional non-use of IDRPs are not concurrently being applied to routing information exchange with an ATN Router with the same NET over a different subnetwork connection.

5.3.5.2.10.8 This is an error and shall be reported to Systems Management; a BIS-BIS connection shall not be established in this case.

5.3.5.2.10.9 **Recommendation.**— *When IDRPs is used to exchange routing information over an Air/Ground subnetwork, the value of the Holding Time field in the ISH PDU should be set to 65534, except when a watchdog timer is applied to the subnetwork connection (see 5.3.5.2.13).*

5.3.5.2.10.10 **Recommendation.**— *When IDRPs is used to exchange routing information over an Air/Ground subnetwork, the Configuration Timer should be set such that no further ISH PDUs are exchanged following the Route Initiation procedure.*

Note 1.— The purpose of the above is to effectively suppress the further generation of ISH PDUs.

Note 2.— Normally, the IDRPs KeepAlive mechanism is sufficient to maintain a check on the “liveness” of the remote ATN Router. However, when watchdog timers are necessary it is also necessary to ensure a “liveness” check on a per subnetwork connection basis. The ISH PDU fulfils this role.

5.3.5.2.11 Exchange of Routing Information using IDRP

5.3.5.2.11.1 Once a BIS-BIS connection has been established with a remote ATN Router, then:

- a) An Airborne Router shall advertise routes to the Air/Ground Router in accordance with the Routing Policy specified in 5.3.7.2.
- b) An Air/Ground Router shall advertise routes to the Airborne Router in accordance with the Routing Policy specified in 5.3.7.1.4 or 5.3.7.3.4 as appropriate for the role of the Air/Ground Router's RD.

5.3.5.2.12 Procedures for the Optional Non-Use of IDRP over an Air/Ground Data Link

5.3.5.2.12.1 General

Note.— In this case, there is no recommendation to suppress the periodic re-transmission of ISH PDUs according to the ISO/IEC 9542 Report Configuration Function. In the absence of IDRP, this re-transmission is necessary to maintain the “liveness” of the connection.

5.3.5.2.12.1.1 When the procedures for the optional non-use of IDRP are employed by an Airborne Router, then all ATN airborne systems on the same aircraft shall be located in the same Routing Domain.

Note.— This is because the procedures specified below make assumptions on the value and length of the NSAP Address Prefix that is common to all systems on board an aircraft, and these assumptions are invalidated if a single aircraft hosts multiple RDs.

5.3.5.2.12.2 Air/Ground Router

5.3.5.2.12.2.1 Through the actions of the IS-SME as specified below, an Air/Ground Router shall simulate the existence of a BIS-BIS connection with an Airborne Router that implements the procedures for the optional non-use of IDRP by implementing the following procedure:

- a) The NET of the remote ATN Router shall be appended to the **externalBISNeighbor** attribute of the BIS's **idrpConfig** Managed Object, if not already present, and an adjacentBIS Managed Object shall be created for the remote ATN Router identified by this NET, if one does not already exist. An Adj-RIB-In shall hence be created for this remote ATN Router and for the Security RIB-Att.

Note.— No activate action will be applied to this MO and the implementation will hence need to be able to process information in the Adj-RIB-In even though the MO is in the “idle” state. Implementations may choose to optimise the operation of these procedures with a special interface to IDRP.

- b) Truncating the NET received on the ISH PDU to the first eleven octets and using the resulting NSAP Address Prefix as the NLRI of a route which shall then be inserted into the Adj-RIB-In for the remote ATN Router and which shall be identified by the Security RIB-Att, as if it had been received over a BIS-BIS connection. This route shall include an RD_Path attribute with the received NET as the Routing Domain Identifier of the originating RD, and an empty security path attribute.

Note.— According to the rules for the update of path information specified in 5.8, the security path attribute will be updated by the Routing Decision process to include an Air/Ground Subnetwork type security tag and an ATSC Class security tag, if this is appropriate. This procedure is identical to the normal use of IDRP over a Mobile Subnetwork.

- c) The well-known mandatory path attribute RD_HOP_COUNT shall be set to 1 in the routes to be inserted into the Adj-RIB-In for the remote Router implementing the procedures for the optional non-use of IDRP. In addition, for routes to be inserted into the Adj-RIB-In for an adjacent Airborne Router implementing the procedures for the optional non-use of IDRP, the well-known mandatory path attribute CAPACITY shall be set according to the capacity of the Mobile Subnetwork(s) over which the Airborne Router is reachable.

5.3.5.2.12.2.2 If a subnetwork connection is concurrently established with the remote ATN Router over which the procedures for the optional non-use of IDRP have been applied, then the IS-SME shall not repeat the above procedures for the new subnetwork connection.

5.3.5.2.12.2.3 Instead, the IS-SME shall cause the IDRP Routing Decision function to be invoked in order to rebuild the FIB taking into account the additional subnetwork connectivity.

5.3.5.2.12.2.4 This shall include re-update of the security information contained in routes received from the remote ATN Router, according to 5.8.

5.3.5.2.12.2.5 The IS-SME shall also ensure that a normal BIS-BIS connection does not concurrently exist with an ATN Router with the same NET.

5.3.5.2.12.2.6 This is an error and shall be reported to Systems Management; the procedures for the optional non-use of IDRP shall not be applied in this case.

5.3.5.2.12.3 Airborne Router

5.3.5.2.12.3.1 An Airborne Router implementing the procedures for the optional non-use of IDRP over a Mobile Subnetwork shall simulate the operation of IDRP by maintaining a Loc-RIB for the Security RIB_Att, which is then used to generate FIB information.

5.3.5.2.12.3.2 Through the actions of its IS-SME, an Airborne Router shall derive entries for this Loc-RIB from the ISH PDU received from an Air/Ground Router as follows:

- a) The IS-SME shall insert into the Loc-RIB, a route derived by truncating the NET received on the ISH PDU to the first eleven octets and using the resulting NSAP Address Prefix as the NLRI of a route. This route shall include a security path attribute with the Air/Ground Subnetwork Type and ATSC Class security tags (if any) determined from the Mobile Subnetwork Capabilities Parameter contained in the options part of the received ISH PDU, or from locally known information if such a parameter is not present in the received ISH PDU.

Note.— This provides routing information for destinations in the Air/Ground Router's RD and assumes that the eleven octet prefix of the Air/Ground Router's NET is common to all destinations in that RD.

- b) The IS-SME shall insert into the Loc-RIB other routes available through the Air/Ground Router determined using locally known information. These routes shall include a security path attribute with the Air/Ground Subnetwork Type and ATSC Class security tags (if any) determined from the Mobile Subnetwork Capabilities Parameter contained in the options part of the received ISH PDU, or from locally known information if such a parameter is not present in the received ISH PDU.

Note.— As these routes are not subject to dynamic update, their availability must be ensured by the operator of the Air/Ground Router, within the limits specified for the applications that will use them.

5.3.5.2.13 Air/Ground Route Termination

Note 1.— The “Leave Event” is defined to signal when subnetwork connectivity with a remote ATN Router over a Mobile Subnetwork ceases to be available. This event may be generated by (a) the subnetwork itself using mechanisms outside of the scope of this specification, or (b) the SND CF when it receives a clear indication from the subnetwork reporting either a network or a user initiated call clearing. The Leave Event is always reported to the IS-SME.

Note 2.— When a Leave Event is generated by a subnetwork, it applies to all subnetwork connections to a given DTE. When it is generated locally by the SND CF, it typically applies to a single subnetwork connection.

5.3.5.2.13.1 Recommendation. — A “Leave Event” should not be generated by the Mobile SND CF when a subnetwork connection is closed due to the expiration of the X.25 Idle timer, except if this subnetwork connection fails to be re-established.

5.3.5.2.13.2 When a Mobile Subnetwork does not provide a network generated Clear Indication (e.g. to indicate that an aircraft has left the range of the Mobile Subnetwork, or when some other communication failure occurs, etc.), an ATN Router shall maintain a “watchdog” timer for each affected subnetwork connection and clear each such subnetwork connection once activity has ceased for a configurable period.

5.3.5.2.13.3 When such a “watchdog” timer expires, this shall be reported as a “Leave Event” for that subnetwork connection.

5.3.5.2.13.4 Recommendation.— The timer should be configurable according to the characteristics of the subnetwork.

Note.— An ATN Router maintains a “watch-dog” timer for each applicable subnetwork connection to detect the event of an aircraft leaving coverage (or other communication failure), if such an event detection is not provided by the subnetwork.

5.3.5.2.13.5 When an IS-SME receives a Leave Event for a subnetwork connection or a DTE on a subnetwork, then it shall ensure that respectively, either the affected subnetwork connection or all subnetwork connections on that subnetwork and with the identified DTE, are cleared.

5.3.5.2.13.6 If, as a result of this procedure, no other subnetwork connection exists anymore on that subnetwork and with the identified DTE, then the IS-SME shall remove the Configuration Information that was extracted from the ISH previously received from that DTE on that specified subnetwork, without waiting for the expiration of the Configuration Information Holding Timer.

5.3.5.2.13.7 If, as a result of this procedure or subsequent to the execution of the ISO/IEC 9542 “Flush Old Configuration” function, Configuration Information, that was extracted from an ISH previously received from that DTE still exists, then,

- a) In the case of an ATN Air-Ground Router having established a BIS-BIS connection with that ATN Router, or having simulated a BIS-BIS connection if that ATN Router implements the procedures for the optional non-use of IDRP, then,
 - 1) The IS-SME shall cause the update of the Security path attribute’s security information of all routes contained in the Adj-RIB-In associated with the remote ATN Airborne Router, and,
 - 2) The IS-SME shall cause the IDRP Routing Decision function to be invoked in order to rebuild the FIB, the Loc_RIB and relevant Adj-RIB-Out(s) taking into account the loss of subnetwork connectivity, and,
 - 3) The Air-Ground Router shall re-advertise all routes affected by the change in subnetwork connectivity that are contained in the Adj-RIB-Out(s) subsequent to the update of the security path attribute’s security information of these routes as specified in 5.8.
- b) In the case of an Airborne Router implementing the procedures for the optional non-use of IDRP, the IS-SME shall update the Security path attribute’s security information of all routes in the Loc-RIB that had been inserted according to the procedures of 5.3.5.2.12.3 as a result of an ISH PDU having been received from the Air/Ground Router for which loss of connectivity is reported.

5.3.5.2.13.8 If, as a result of the procedure 5.3.5.2.13.6 or subsequent to the execution of the ISO/IEC 9542 “Flush Old Configuration” function, no Configuration Information exists anymore for the ATN Router for which loss of connectivity is reported, then,

- a) In the case of an ATN Router having established a BIS-BIS connection with that ATN Router, an IDRP deactivate action shall be invoked to terminate that BIS-BIS connection.

Note.— As a consequence of the deactivate action and following normal IDRP operation, the IDRP Routing Decision process will be invoked, the local FIB updated and routes previously available via the remote ATN Router may be withdrawn if suitable alternatives are not available.

- b) In the case of an Air/Ground Router having simulated a BIS-BIS connection to an ATN Airborne Router, implementing the procedures for the optional non-use of IDRP, all routes shall be removed from the Loc-RIB that had been inserted into it according to the procedures of 5.3.5.2.12.2 as a result of an ISH PDU having been received from the Airborne Router for which a loss of connectivity is reported.

- c) In the case of an Airborne Router implementing the procedures for the optional non-use of IDRP, all routes shall be removed from the Loc-RIB that had been inserted into it according to the procedures of 5.3.5.2.12.3 as a result of an ISH PDU having been received from the Air/Ground Router for which a loss of connectivity is reported.

5.3.5.2.13.9 If the BIS-BIS connection is not re-established within a period configurable from 1 minute to 300 minutes, or when the resources are required for other use, then the adjacentBIS Managed Object associated with the initiating BIS shall be deleted, and the initiating BIS's NET removed from the externalBISNeighbor attribute of the BIS's idrpConfig Managed Object.

5.3.5.2.14 Subnetwork Handoff

Note 1.— Handoff is implemented by some subnetworks, for example, the VHF Digital Link (VDL), when an aircraft moves out of the coverage of a Ground Station it is currently using and into the coverage of another - typically operated by the same Service Provider. When the change of Ground Station also requires a change of ATN Air/Ground Router then the subnetwork may simply generate a Join Event for the new Air/Ground Router, followed by a Leave Event for the old Air/Ground Router. However, when the Air/Ground Router accessed through the old Ground Station is also accessible through the new Ground Station then a different procedure is required if the full overhead of Route Initiation is to be avoided.

Note 2.— A further event - the “Handoff Event” - and additional to the “Join” and “Leave” events is defined to initiate such a procedure. A Handoff Event may be received by an Airborne or an Air/Ground Router irrespective of whether the subnetwork is Air- or Ground-initiated, or both. The Handoff Event is also processed by the IS-SME.

Note 3.— The parameters of a Handoff Event include the DTE Address of the system for which Handoff is to take place, and may also include subnetwork specific information (e.g. to direct a Call Request via a specific Ground Station).

5.3.5.2.14.1 On receipt of a Handoff Event, the IS-SME shall check to see if a subnetwork connection already exists with the DTE identified by the Handoff Event. If it does not, then the Handoff Event shall be processed identically to a Join Event.

5.3.5.2.14.2 If a subnetwork connection already exists with the identified DTE, then the ATN Router shall issue an ISO/IEC 8208 Call Request to that DTE. If a different DTE Address to the originally called DTE Address was reported when the connection had previously been made to that DTE, then the returned Called DTE Address shall be used and not the originally called DTE Address.

5.3.5.2.14.3 If more than one subnetwork connection exists with the identified DTE, each with a distinct subnetwork connection priority, then a new subnetwork connection shall be initiated for each such subnetwork connection priority.

Note 1.— If the Maintenance/Initiation of the Local Reference Directory option is selected (see 5.7.6.2.1.5.12), then the subnetwork connection(s), once established, may become part of the same subnetwork connection group(s) as the one(s) of the old subnetwork connection(s). If this is the case, then the LREF Directory will be taken over by the new subnetwork connection(s).

Note 2.— No further action needs to be taken once the subnetwork connection(s) have been successfully established. This is because no change is implied to the Routing Information Base and the underlying subnetwork is responsible for timing out and disconnecting the old subnetwork connections, once all data in transit has been delivered.

Note 3.— In the case that a new (set of) connection(s) is established, existing old connections between the same pair of DTEs are likely to become unavailable shortly. Implementations are advised to use these new subnetwork connection(s) in preference to the old subnetwork connections(s).

5.3.5.2.15 Re-establishment of BIS-BIS connection

5.3.5.2.15.1 The IS-SME shall attempt to re-establish a BIS-BIS connection using the procedures in 5.3.5.2.10 irrespective of which side first initiated the adjacency when:

- a) a previously established BIS-BIS connection with the same remote ATN router is terminated for reasons other than the receipt of a Leave Event by the IS-SME, or
- b) a previous attempt to establish a BIS-BIS connection failed,

and at least one subnetwork connection between the local and remote ATN Router exists.

Note 1.— This procedure guarantees that whenever a subnetwork connectivity is available between an ATN Airborne and ATN Air/Ground Router routes are made available via IDRP and NPDUs can be exchanged via the air/ground adjacency.

Note 2.— This procedure will cause an OPEN BISPDU to be sent irrespective of which side was the initiator of the initial BIS-BIS connection in order to force the resynchronisation of the local and remote IDRP protocol machines which may be out of sync as a result of the failure causing the termination of a BIS-BIS connection.

5.3.5.2.16 APRL for Air/Ground Route Initiation

5.3.5.2.16.1 General

Item	Description	ATN SARPs Reference	ATN Support
njSubnet	Support of Subnetworks that do not provide a Join Event	5.3.5.2	O.1
jSubnet	Support of Subnetworks that do provide a Join Event	5.3.5.2	O.1
giSubnet	Support of Ground-Initiated Subnetworks	5.3.5.2	O.2
aiSubnet	Support of Air-Initiated Subnetworks	5.3.5.2	O.2
agSubnet	Support of Air or Ground-Initiated Subnetworks	5.3.5.2	O.2
fsSubnet	Support of Subnetworks that support Fast Select	-	O
noIDRP-a	Support of optional non-use of IDRP by Airborne BIS	5.3.5.2.12.3	O

Item	Description	ATN SARPs Reference	ATN Support
noIDRP-ag	Support of optional non-use of IDRP by Air/Ground BIS	5.3.5.2.12.2	M
lvSubnet	Support of Subnetworks that provide a Leave Event	5.3.5.2.13	M
sgClearInd	Provision of subnetwork generated Clear Indication	5.3.5.2.13	O
HoSubnet	Support of Subnetworks that provide a Handoff Event	5.3.5.2.14	O

5.3.5.2.16.2 Airborne Router - Subnetwork Connection Responder

Item	Description	ATN SARPs Reference	ATN Support
respAR-ar	Response to incoming Call Request	5.3.5.2.2	giOragSubnet: M
valCR-ar	Validation of incoming Call Request	5.3.5.2.2	giOragSubnet: O
RespISH-ar	Generation of ISH PDU	5.3.5.2.6	giOragSubnet: M
ISHinCC-ar	Encoding ISH PDU in Call Accept User Data	5.3.5.2.6	RespISH-ar and fsSubnet: O
negNoIDRP-ar	Transmission of ISH PDU with SEL field of NET set to FEh	5.3.5.2.6	giOragSubnet and noIDRP-a: M
negIDRP-ar	Transmission of ISH PDU with SEL field of NET set to zero	5.3.5.2.6	giOragSubnet and ^noIDRP-a: M
autoRoute-ar	Inference of available routes from received NET of A/G Router	5.3.5.2.12	giOragSubnet and noIDRP-a: M
initIDRP-ar	IDRP startup procedures - Invoke activate action	5.3.5.2.10	giOragSubnet and ^noIDRP-a: M
supISH-ar	Suppression of multiple ISH PDUs	5.3.5.2.10	giOragSubnet and ^noIDRP-a: O
valNET-ar	Validation of received NET	5.3.5.2.7	giOragSubnet and ^noIDRP-a: O
Handoff-ar	Processing of Handoff Event	5.3.5.2.14.1	HoSubnet: M

giOragSubnet: giSubnet or agSubnet

5.3.5.2.16.3 Airborne Router - Subnetwork Connection Initiator

Item	Description	ATN SARPs Reference	ATN Support
polling-ai	Procedures for polling a list of subnet addresses	5.3.5.2.3.1	pollReq: M
backoff-ai	Backoff Procedure	5.3.5.2.3.1.2	pollReq: M
connect-ai	Connect on receipt of Join Event	5.3.5.2.3.2	EventDrvn: M
ValJoin-ai	Validation of Join Event	5.3.5.2.3.2	EventDrvn: O
SendISH-ai	Generation of ISH PDU	5.3.5.2.6	EventDrvn or pollReq:M
ISHinCR-ai	Encoding of ISH PDU in Call Request	5.3.5.2.6	SendISH-ai and fsSubnet: O
negNoIDRP-ai	Transmission of ISH PDU with SEL field of NET set to FEh	5.3.5.2.8	(EventDrvn or pollReq) and noIDRP-a:M
negIDRP-ai	Transmission of ISH PDU with SEL field of NET set to zero	5.3.5.2.8	(EventDrvn or pollReq) and ^noIDRP-a:M
autoRoute-ai	Inference of available routes from received NET of A/G Router	5.3.5.2.12.3	(EventDrvn or pollReq) and noIDRP-a:M
initIDRP-ai	IDRP startup procedures - listenForOpen set to true	5.3.5.2.10	(EventDrvn or pollReq) and ^noIDRP-a:M
supISH-ai	Suppression of multiple ISH PDUs	5.3.5.2.10	(EventDrvn or pollReq) and ^noIDRP-a: O
valNET-ai	Validation of received NET	5.3.5.2.7	(EventDrvn or pollReq) and ^noIDRP-a: O
RelatedTE-ai	Maintain relationship between originally Called and Returned Called DTE Address	5.3.5.2.3.1.1.5	HoSubnet: M
Handoff-ai	Processing of Handoff Event	5.3.5.2.14.4	HoSubnet: M

pollReq: aiSubnet and njSubnet

EventDrvn: jSubnet and (aiSubnet or agSubnet)

5.3.5.2.16.4 Air/Ground Router - Subnetwork Connection Responder

Item	Description	ATN SARPs Reference	ATN Support
respAR-agr	Response to incoming Call Request	5.3.5.2.2	aiOragSubnet: M
valCR-agr	Validation of incoming Call Request	5.3.5.2.2	aiOragSubnet: O
RespISH-agr	Generation of ISH PDU	5.3.5.2.6	aiOragSubnet: M
ISHinCC-agr	Encoding ISH PDU in Call Accepted User Data	5.3.5.2.6	RespISH-agr and fsSubnet: O
negNoIDRP-agr	Receipt of ISH PDU with SEL field of NET set to FEh	5.3.5.2.8	aiOragSubnet: M
negIDRP-agr	Receipt of ISH PDU with SEL field of NET set to zero	5.3.5.2.8	aiOragSubnet: M
autoRoute-agr	Inference of available routes from received NET of Airborne Router	5.3.5.2.12.2	aiOragSubnet: M
initIDRP-agr	IDRP startup procedures - Invoke activate action	5.3.5.2.10	aiOragSubnet: M
supISH-agr	Suppression of multiple ISH PDUs	5.3.5.2.10	aiOragSubnet: O
valNET-agr	Validation of received NET	5.3.5.2.7	aiOragSubnet: O
Handoff-agr	Processing of Handoff Event	5.3.5.2.14.1	HoSubnet: M

aiOragSubnet: aiSubnet or agSubnet

5.3.5.2.16.5 Air/Ground Router - Subnetwork Connection Initiator

Item	Description	ATN SARPs Reference	ATN Support
connect-agi	Connect on receipt of Join Event	5.3.5.2.4	goOragSubnet: M
ValJoin-agi	Validation of Join Event	5.3.5.2.4	connect-agi: O

Item	Description	ATN SARPs Reference	ATN Support
SendISH-agi	Generation of ISH PDU	5.3.5.2.6	connect-agi: M
ISHinCR-agi	Encoding of ISH PDU in Call Request	5.3.5.2.6	Send-ISH-agi and fsSubnet: O
negNoIDRP-agi	Receipt of ISH PDU with SEL field of NET set to FEh	5.3.5.2.8	goOragSubnet:M
negIDRP-agi	Receipt of ISH PDU with SEL field of NET set to zero	5.3.5.2.8	goOragSubnet:M
autoRoute-agi	Inference of available routes from received NET of Airborne Router	5.3.5.2.12.2	goOragSubnet:M
initIDRP-agi	IDRP startup procedures - listenForOpen set to true	5.3.5.2.10	goOragSubnet:M
supISH-agi	Suppression of multiple ISH PDUs	5.3.5.2.10	goOragSubnet:O
valNET-agi	Validation of received NET	5.3.5.2.7	goOragSubnet:O
RelatedTE-agi	Maintain relationship between originally Called and Returned Called DTE Address	5.3.5.2.4.2	HoSubnet: M
Handoff-agi	Processing of Handoff Event	5.3.5.2.14.1	HoSubnet: M

goOragSubnet: giSubnet or agSubnet

5.3.5.2.16.6 Termination Procedures

Item	Description	ATN SARPs Reference	ATN Support
lvEvent	Processing of Leave Event	5.3.5.2.13	M
Watchdog	Watchdog Timer	5.3.5.2.13	^sgClearInd:M
ConfigWD	Configurability of Watchdog for Subnetwork Characteristics	5.3.5.2.13	^sgClearInd:O
conLeave	Processing of a per connection Leave Event	5.3.5.2.13	M
subnetLeave	Processing of a per subnetwork Leave Event	5.3.5.2.13	M

5.3.6 Handling Routing Information

5.3.6.1 All ATN Routers in the same RD shall implement the same routing policy.

Note 1.— As specified in 5.8, an ATN Router supports both the empty (default) RIB_Att, and the RIB_Att comprising the Security Path Attribute identifying the ATN Security Registration Identifier. An ATN Router therefore includes two RIBs known as the default RIB and the Security RIB, each of which comprises a Loc-RIB, and an Adj-RIB-In and an Adj-RIB-Out for each adjacent BIS.

Note 2.— Each ATN RD will necessarily have a distinct routing policy that depends on its nature, and the nature of the RDs to which it is interconnected. Section 5.3.7 specifies the baseline Routing Policy for each class of RD identified in 5.2.2.2 to 5.2.2.5 inclusive. ATN RDs may then extend the specified baseline to match their actual requirements.

Note 3.— Each Routing Policy is expressed as a set of policy statements or rules.

Note 4.— These baseline policy statements given below are always subject to the ISO/IEC 10747 requirement that routes are only advertised when the DIST_LIST_INCL and DIST_LIST_EXCL path attributes, if present, permit the route to be so advertised. Routes may never be advertised to an RD or RDC which the route has already traversed.

5.3.7 Policy Based Selection of Routes for Advertisement to Adjacent RDs

Note.— In general, the selection of routes for advertisement to adjacent Routing Domains is performed according to local routing policy rules. This specification mandates such routing policy rules for support of Air/Ground routing only. Routing Policy rules for support of Ground/Ground routing are a local matter.

5.3.7.1 Routing Policy Requirements for Members of an ATN Island Backbone RDC

5.3.7.1.1 General

5.3.7.1.1.1 An ATN RD that is a member of an ATN Island Backbone RDC shall implement a Routing Policy that is compatible with the policy statements given in this section and its subordinate sections.

5.3.7.1.2 Adjacent ATN RDs within the Backbone RDC

Note.— These policy statements apply to the routes advertised by an ATN Router in an RD that is a member of an ATN Island Backbone RDC, to an adjacent ATN Router in a different RD, which is also a member of the same ATN Island Backbone RDC.

5.3.7.1.2.1 Each ATN Router that is in an RD that is a member of an ATN Island Backbone RDC shall provide the following routes to each adjacent ATN RD within the same ATN Island Backbone RDC, and for the Security RIB-Att:

- a) A route to NSAPs and NETs contained within the RD; the route's destination shall be one or more NSAP Address prefixes common to all NSAP Addresses and NETs in the RD. If restrictions on distribution scope are applied by local routing policy, then they shall not prevent distribution of this route to any member of the same ATN Island Backbone RDC.

Note 1.— The well known discretionary attribute DIST_LIST_INCL may also be present. For example, to restrict the scope of the information to members of the ATN Island Backbone RDC only. The RDIs of other RDs and RDCs may also be present at the discretion of the local Administrative Domain, and by bilateral agreement.

Note 2.— The objective of this rule is to ensure that a member of an ATN Island Backbone RDC will tell all its neighbours within the backbone RDC about itself.

- b) The selected route to every Mobile RD for which a route is available.

Note.— The objective of this rule is to ensure that a member of an ATN Island Backbone RDC will inform all other Backbone RDC members within the island about all mobiles that it has available.

- c) The selected route to every Fixed ATN RD in the same ATN Island, for which a route is available.

Note.— The objective of this rule is to ensure that a member of an ATN Island Backbone RDC will tell other members of the same Backbone RDC about all fixed RDs that it knows about.

- d) Each selected route to a Mobile RD's "home".

Note 1.— The objective of this rule is to ensure that a member of an ATN Island Backbone RDC will tell all other members of the same Backbone RDC about all the "homes" that it knows about.

Note 2.— Such a route can be characterised by an NSAP Address Prefix which ends at the ADM field.

- e) A route to each "Home" that the ATN TRD itself provides for Mobile RDs. This route has as its destination, the common NSAP Address Prefix(es) for those mobile RDs. The Security Path attribute shall contain an ATSC Class Security Tag indicating support for both ATSC and non-ATSC traffic, and for all ATSC classes supported for Air/Ground data interchange, if any.

Note.— The objective of this item is to ensure that all RDs in the ATN Island Backbone RDC are aware that the identified "Homes" are located here.

5.3.7.1.3 All other ATN RDs within the ATN Island

Note.— These policy statements apply to the routes advertised by an ATN Router in an RD that is a member of an ATN Island Backbone RDC to an adjacent ATN Router in a different RD, which is also a member of the same ATN Island RDC, but which is not a member of that ATN Island Backbone RDC.

5.3.7.1.3.1 An ATN Router that is in an RD that is a member of an ATN Island Backbone RDC shall provide the following routes to each adjacent ATN RD within the ATN Island RDC, which is not a member of the ATN Island's Backbone RDC, and for the Security RIB-Att :

- a) A route to NSAPs and NETs contained within the RD; the route's destination shall be one or more NSAP Address prefixes common to all NSAP Addresses and NETs in the RD. If restrictions on distribution scope are applied by local routing policy, then they shall not prevent distribution of this route to any member of the same ATN Island RDC.

Note 1.— The well known discretionary attribute DIST_LIST_INCL may also be present. For example, to restrict the scope of the information to members of the ATN Island only. The RDIs of other RDs and RDCs may also be present at the discretion of the local Administrative Domain, and by bilateral agreement.

Note 2.— The objective of this rule is to ensure that a member of an ATN Island Backbone RDC will tell all RDs within the island about itself.

- b) The selected route to every Fixed ATN RD in the same ATN Island for which a route is available.

Note.— The objective of this rule is to ensure that an ATN Router located in an RD that is a member of an ATN Island Backbone RDC, will tell all RDs within the island about all the fixed RDs it knows about.

- c) A route to all AINSC Mobiles and all ATSC Mobiles. The well known discretionary attribute DIST_LIST_INCL shall be present, and shall contain the RDI of the ATN

Island RDC as its value. The Security Path attribute shall contain an ATSC Class Security Tag indicating support for both ATSC and non-ATSC traffic, and for all ATSC classes supported for Air/Ground data interchange, if any.

Note 1.— The objective of this rule is to tell the rest of the island that each RD in the ATN Island Backbone RDC provides a default route to all aircraft.

Note 2.— The distribution scope of the route is limited because the ATN Island defines the domain of the default route provider. This route is invalid outside of the local ATN Island.

Note 3.— This route is formally the result of aggregating all the routes to Mobile Systems and routes to “Home” RDs, known to the ATN Router.

- d) A route to each Mobile RD for which the adjacent RD is advertising a route to the Mobile RD's “home”.

Note.— The objective of this rule is to ensure that a member of an ATN Island Backbone RDC will tell all adjacent off Backbone RDs about all routes to Mobile RDs which have “home” routes advertised.

5.3.7.1.4 Mobile RDs

Note.— These policy statements apply to the routes advertised by an ATN Router in an RD that is a member of an ATN Island Backbone RDC to an adjacent ATN Router in a Mobile RD.

5.3.7.1.4.1 When IDRP is being used to exchange routing information with an Airborne Router, an ATN Router in an RD that is a member of an ATN Island Backbone RDC shall provide to each adjacent Mobile RD a route to NSAPs and NETs contained within the local RD for the Security RIB-Att; the route's destination shall be one or more NSAP Address prefixes common to all NSAP Addresses and NETs in the local RD.

Note.— The objective of this rule is to ensure that an RD that is a member of an ATN Island Backbone RDC will tell all adjacent Mobiles about itself.

5.3.7.1.4.2 **Recommendation.**— An ATN RD that is a member of an ATN Island Backbone RDC should also provide to each adjacent Mobile RD, and for the Security RIB-Att and for which a suitable route exists:

- a) An aggregated route to NSAPs and NETs contained within the local ATN Island RDC;

Note.— The objective of this rule is to ensure that an RD that is a member of an ATN Island Backbone RDC provides to each connected Mobile RD, a route to all fixed ATN RDs within the island.

- b) An aggregated route to NSAPs and NETs contained within all other ATN Islands for which a route is available.

Note.— The objective of this rule is to ensure that an RD that is a member of an ATN Island Backbone RDC will provide to each connected Mobile RD routing information to the backbone of other ATN islands.

5.3.7.1.5 ATN RDs in other ATN Islands

Note.— These policy statements apply to the routes advertised by an ATN Router in an RD that is a member of an ATN Island Backbone RDC to an adjacent ATN Router in a different RD, which is a member of a different ATN Island's ATN Island Backbone RDC.

5.3.7.1.5.1 An ATN Router in an RD that is a member of an ATN Island Backbone RDC shall provide the following routes to each adjacent ATN Router that is a member of a Backbone RDC in another ATN Island, and or the Security RIB-Att:

- a) An aggregated route to NSAPs and NETs contained within the ATN Island RDC.

Note.— The objective of this rule is to ensure that an RD that is a member of an ATN Island Backbone RDC will tell all adjacent RDs that are members of ATN Island Backbone RDCs in different ATN Islands about the local ATN Island.

- b) Each selected route to a Mobile RD's "home".
- c) A route to each "Home" that the ATN TRD itself provides for Mobile RDs. This route has as its destination, the common NSAP Address Prefix(es) for those Mobile RDs. The Security Path attribute shall contain an ATSC Class Security Tag indicating support for both ATSC and non-ATSC traffic, and for all ATSC classes supported for Air/Ground data interchange, if any.

Note 1.— The objective of this rule is to ensure that an ATN Island Backbone RD will tell all adjacent RDs that are member of an ATN Island Backbone RD in different ATN Islands about routes to Mobiles whose "home" is in the local island.

Note 2.— The "home" identified above needs not correspond to a geographical notion of a home.

Note 3.— The "home" is typically identified by an NSAP Address Prefix that identifies all the RD's belonging to the organisation responsible for the Mobile RD (i.e. aircraft), or all the Mobile RDs belonging to the organisation. The former is only possible if all such Fixed RDs are part of the same ATN Island RDC.

- d) A known route to each Mobile RD for which the adjacent RD is advertising a route to the Mobile RD's "home".

Note.— The objective of this rule is to ensure that a member of an ATN Island Backbone RDC will tell all adjacent RDs in different islands about all routes to Mobile RDs which have "home" routes advertised.

5.3.7.2 Routing Policy Requirements for a Mobile RD

5.3.7.2.1 When IDRP is being used to exchange routing information with an Airborne Router, a Mobile RD shall provide to each ATN RD to which it is currently connected, a route to NSAPs and NETs contained within the Mobile RD for the Security RIB-Att.

Note 1.— The objective of this rule is to ensure that a Mobile RD will tell adjacent RDs about itself.

Note 2.— This policy statement applies to the routes advertised by an ATN Router in a Mobile RD to an adjacent ATN Air/Ground Router in a Fixed ATN RD.

5.3.7.3 Routing Policy Requirements for an ATN TRD that is not a Member of the ATN Island Backbone RDC

5.3.7.3.1 General

5.3.7.3.1.1 An RD that is a member of an ATN Island RDC, and is a TRD, but which is not a member of that ATN Island's Backbone RDC shall implement a Routing Policy that is compatible with the policy statements given in this section and its subordinate sections.

Note.— An ATN RD that operates as a transit routing domain is referred to in this chapter as an ATN TRD.

5.3.7.3.2 Adjacent ATN RDs that are Members of the ATN Island's Backbone RDC

Note.— These policy statements apply to the routes advertised by an ATN Router in an ATN TRD to an adjacent ATN Router in an RD which is a member of the local ATN Island's Backbone RDC.

5.3.7.3.2.1 When an ATN TRD that is not itself a member of an ATN Island Backbone RDC is adjacent to an RD that is a member of an ATN Island Backbone RDC, then it shall provide the following routes to each such adjacent ATN RD, and for the Security RIB-Att:

- a) A route to NSAPs and NETs contained within the RD; the route's destination shall be one or more NSAP Address prefixes common to all NSAP Addresses and NETs in the RD.

Note 1.— The well known discretionary attribute DIST_LIST_INCL may also be present. For example, to restrict the scope of the information to members of the ATN Island only. The RDIs of other RDs and RDCs may also be present at the discretion of the local Administrative Domain, and by bilateral agreement.

Note 2.— The objective of this rule is to ensure that an ATN TRD that is not itself a member of an ATN Island Backbone RDC, will tell all adjacent ATN RDs which are members of an ATN Island Backbone RDC within the same ATN Island about itself.

- b) The selected route to every Mobile RD for which a route is available.

Note.— The objective of this rule is to ensure that an ATN TRD that is not itself a member of an ATN Island Backbone RDC, will tell all adjacent ATN RDs which are members of an ATN Island Backbone RDC within the same ATN Island about all Mobiles it knows about.

- c) The selected route to every Fixed ATN RD in the ATN Island for which a route is available.

Note.— The objective of this rule is to ensure that an ATN TRD that is not itself a member of an ATN Island Backbone RDC, will tell all adjacent ATN RDs which are members of an ATN Island Backbone RDC within the same ATN Island about all fixed RDs it knows about in the same ATN Island.

- d) A route to each “Home” that the ATN TRD itself provides for Mobile RDs. This route shall have as its destination, the common NSAP Address Prefix(es) for those Mobile RDs. The Security Path attribute shall contain an ATSC Class Security Tag indicating support for both ATSC and non-ATSC traffic, and for all ATSC classes supported for Air/Ground data interchange.

Note.— The objective of this rule is to support the operation of the Home Domain concept on any ATN TRD directly connected to an ATN Island Backbone RD.

5.3.7.3.3 Adjacent ATN RDs within the same ATN Island and which are not Members of the ATN Island’s Backbone RDC

Note.— These policy statements apply to the routes advertised by an ATN Router in an ATN TRD to an adjacent ATN Router in an ATN RD on the same ATN Island.

5.3.7.3.3.1 An ATN TRD shall provide the following routes to each adjacent ATN RD within the ATN Island RDC, other than ATN RDs which are members of the ATN Island Backbone RDC, and for the Security RIB-Att:

- a) A route to NSAPs and NETs contained within the RD for the Security RIB-Att; the route’s destination shall be one or more NSAP Address prefixes common to all NSAP Addresses and NETs in the RD.

Note 1.— The well known discretionary attribute DIST_LIST_INCL may also be present. For example, to restrict the scope of the information to members of the ATN Island only. The RDIs of other RDs and RDCs may also be present at the discretion of the local Administrative Domain, and by bilateral agreement, including the RDI of the ATN Island Backbone RD or RDC, when the adjacent RD is providing the local RD’s route to the ATN Island Backbone.

Note 2.— The objective of this rule is to ensure that an ATN TRD that is not itself a member of the ATN Island Backbone RDC, will tell all adjacent RDs within the island about itself.

- b) The selected route to every Fixed RD in the same ATN Island for which a route is available.

Note.— The objective of this rule is to ensure that an ATN TRD that is not itself a member of the ATN Island Backbone RDC, will tell all adjacent RDs within the island about all fixed ATN RDs in the same ATN Island that it knows about.

- c) If the RD is currently advertising the preferred route to all AINSC and ATSC Mobiles, then every route to an AINSC Mobile and an ATSC Mobile that is known to the local RD shall be advertised to this RD, subject only to constraints imposed by any DIST_LIST_INCL and DIST_LIST_EXCL path attributes.

Note.— The objective of this rule is to ensure that the provider of the default route to all aircraft (i.e. the Backbone) is kept informed of the actual location of every aircraft adjacent to the Island.

- d) The preferred route to all Mobiles, except when the RD is the source of this route.

Note.— The objective of this rule is to ensure propagation of the default route to all Mobiles throughout the ATN Island.

- e) A route to each Mobile RD for which the adjacent RD is advertising the preferred route to the Mobile RD's "home".

Note.— The objective of this rule is to ensure routes to Mobile RDs are propagated to off Backbone Homes.

- f) A route to each "Home" that the ATN TRD itself provides for Mobile RDs. This route has as its destination, the common NSAP Address Prefix(es) for those Mobile RDs. The Security Path attribute shall contain an ATSC Class Security Tag indicating support for both ATSC and non-ATSC traffic, and for all ATSC classes supported for Air/Ground data interchange, if any.

Note.— The objective of this item is to ensure that all RDs in the ATN Island are aware that the identified "Homes" are located here.

5.3.7.3.4 Mobile RDs

Note.— These policy statements apply to the routes advertised by an ATN Router in an ATN TRD to an adjacent ATN Router in a Mobile RD.

5.3.7.3.4.1 When IDRPs are being used to exchange routing information with the airborne router, an ATN TRD shall provide to each adjacent Mobile RD a route to NSAPs and NETs contained within the RD for the Security RIB-Att; the route's destination shall be one or more NSAP Address prefixes common to all NSAP Addresses and NETs in the RD.

Note.— The objective of this rule is to ensure that an ATN TRD will tell adjacent Mobile RDs about itself.

5.3.7.3.4.2 **Recommendation.**— An ATN TRD should also provide to each adjacent Mobile RD, and for the Security RIB-Att and for which a suitable route exists:

- a) an aggregated route to NSAPs and NETs contained within the local ATN Island RDC;
- b) an aggregated route to NSAPs and NETs contained within all other ATN Islands for which a route is available.

Note.— The objective of this rule is to encourage an RD to provide to each adjacent Mobile RD routing information about: a) all fixed RDs within the island, and b) other ATN islands.

5.3.7.4 The Routing Policy for a Fixed ATN ERD

5.3.7.4.1 A Fixed ATN ERD shall provide to each ATN RD to which it is currently connected, a route to NSAPs and NETs contained within the RD, for the Security RIB-Att.

Note 1.— The well known discretionary attribute DIST_LIST_INCL may be present, unless the RD permits routes to destinations within itself to be advertised by other ATN RDs without restriction to any other ATN RD, or non-ATN RD.

Note 2.— This policy statement applies to the routes advertised by an ATN Router in a fixed ATN ERD to an adjacent ATN Router in an ATN RD.

Note 3.— An ERD does not advertise routes to destinations in any other RD, to another RD.

5.4 NETWORK AND TRANSPORT ADDRESSING SPECIFICATION

5.4.1 Introduction

Note 1.— The ATN Internet Addressing Plan defines an OSI Network Service Access Point (NSAP) address structure which can support efficient internet routing procedures, and which conforms to common abstract syntax, semantic and encoding rules throughout the ATN OSI environment.

Note 2.— This addressing plan also defines the format and use of TSAP Selectors to enable the unambiguous identification of Multiple Transport Service users within a single End System.

5.4.1.1 Addressing Plan Scope

5.4.1.1.1 The ATN Internet Addressing Plan shall be used by ATN End Systems and Intermediate Systems.

Note.— The ATN Internet Addressing Plan serves the needs of a variety of aeronautical data communication user groups, including ATSC and AINSC users.

5.4.1.2 Addressing Plan Applicability

Note.— The ATN Internet Addressing Plan defines the Network and Transport Layer addressing information to be utilized by ATN End Systems, and by ATN Intermediate Systems.

5.4.1.3 Reserved Values in Address Fields

5.4.1.3.1 Address field values specified as “reserved” shall not be used until assigned by future versions of this specification.

5.4.1.4 Values of Character Format Fields

5.4.1.4.1 When the value of a field is defined as a character string, then the actual value of the field shall be derived from the IA-5 encoding of each character in the character string.

5.4.1.4.2 The IA-5 encoding of the first character in the string shall be taken as the value of the first octet of the field and so on until all octets in the field have been given a value.

5.4.1.4.3 If the length of the character string is smaller than the number of octets in the field, then the character string shall be right padded with the space character.

5.4.1.4.4 The most significant bit of each octet shall be set to zero.

Note.— For example, the character string ‘EUR’ would be encoded as 455552 hexadecimal, in a three octet field.

5.4.2 Transport Layer Addressing

5.4.2.1 General

Note 1.— This section provides requirements on the format of ATN TSAP addresses. An ATN TSAP address is an NSAP address and a TSAP selector.

Note 2.— The requirements in this section apply to the administration of transport addresses local to an ATN End System. They do not apply to all systems in a global OSI Environment. An ATN System may allow remote transport addresses to obey different standards, e.g. when interworking with a non-ATN system is required.

5.4.2.2 ATN TSAP Selector

5.4.2.2.1 An ATN TSAP selector shall be either one or two octets in length.

5.4.2.2.2 The TSAP Selector field shall be administered on a local basis.

5.4.2.2.3 Valid ATN TSAP Selector field values shall be in the range **0** to **65535**.

5.4.2.2.4 The TSAP Selector field shall be encoded as an unsigned binary number.

5.4.2.2.5 If the TSAP Selector needs to be encoded in more than one octet, then the number shall be encoded with the most significant octet first.

Note.— This follows the encoding rules specified in ISO/IEC 8073.

5.4.2.2.6 **Recommendation.**— *TSAP selector values in the range **0** to **255** should be encoded using one octet, higher values should be encoded using two octets.*

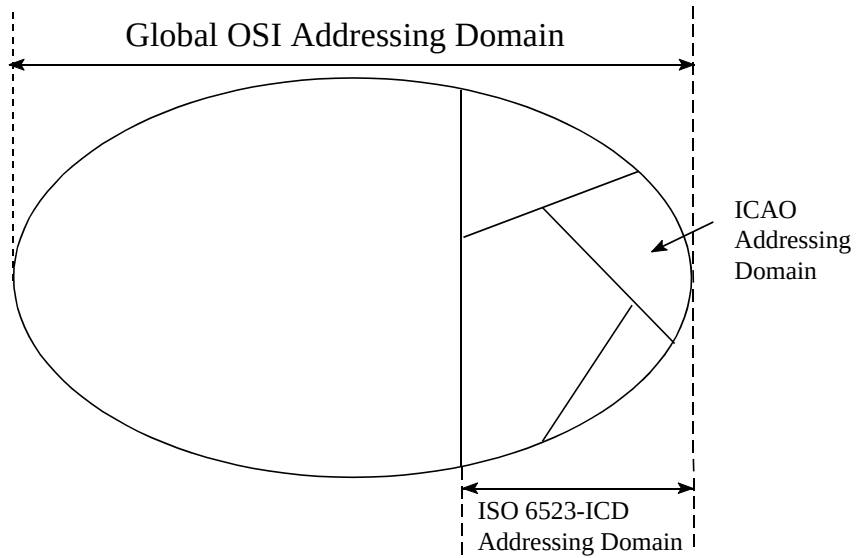


Figure 5.4-1 The Global OSI Network Addressing Domain

5.4.3 Network Layer Addressing

5.4.3.1 NSAP Addresses and Network Entity Titles (NETs)

Note 1.— The NSAP Address is formally defined in ISO/IEC 8348. It is the name of a Network Service Access Point (NSAP) located in an End System, and uniquely identifies that NSAP. It is also an address that may be used to find that NSAP.

Note 2. — The Network Entity Title (NET) is also formally defined in ISO/IEC 8348 and is the name of a Network Entity located within an End or Intermediate System. NETs are syntactically identical to NSAP Addresses and are allocated from the same address space. An NET is also an address that may be used to find the Network Entity.

Note 3.— An NSAP Address Prefix is a substring of an NSAP Address or NET that is comprised of the first ‘n’ characters of the NSAP Address or NET.

5.4.3.2 Network Addressing Domains

Note 1.— A Network Addressing Domain comprises all NSAP Addresses and NETs with a common NSAP Address Prefix, and is always a sub-domain of the Global NSAP Addressing Domain which contains all NSAP Addresses. This nesting of network addressing domains within the Global Network Addressing Domain is conceptually illustrated in Figure 5.4-1.

Note 2.— A Network Addressing Domain has a single Administrator responsible for the assignment of NSAP Addresses and NSAP Address Prefixes within the domain. A Network Addressing Domain is often sub-divided into a number of sub-ordinate domains each characterised by a unique NSAP Address Prefix. Management of such sub-ordinate Network Addressing Domains may then be devolved to another Administrator.

5.4.3.2.1 An ATSC Network Addressing Domain shall be a Network Addressing Domain administered by an ATSC authority.

5.4.3.2.2 An AINSC Network Addressing Domain shall be a Network Addressing Domain administered by a member of the Aeronautical Industry.

5.4.3.2.3 ATN End Systems or Intermediate Systems located on-board general aviation aircraft shall belong to an ATSC Network Addressing Domain, whereas ATN systems installed on-board commercial aircraft shall belong to an AINSC Network Addressing Domain.

5.4.3.3 The Syntax of an NSAP Address

Note 1.— Following ISO/IEC 10589, a Router interprets an NSAP Address as a three-fields bit string. This is illustrated in Figure 5.4-2 below.



Figure 5.4-2 ISO/IEC 10589 NSAP Address Syntax

Note 2.— An Area Address is typically common to all NSAP Addresses and NETs assigned to systems in a single Routing Area.

Note 3.— An Area Address is an example of an NSAP Address Prefix.

Note 4.— A System Identifier uniquely identifies an End or Intermediate System within a Routing Area.

Note 5.— A Selector (SEL) identifies a Network Service User or the Network Entity within an End or Intermediate System.

5.4.3.4 The ATN Addressing Plan

Note 1.— ISO/IEC 8348 has specified how the Global Network Addressing Domain is broken down into a number of sub-ordinate Network Addressing Domains, each of which is identified by a unique identifier that forms the initial part of all NSAP Addresses and NETs in those sub-ordinate domains. This initial part is known as the Initial Domain Part (IDP). The IDP itself is defined as comprising two parts: an Authority Format Identifier (AFI) and an Initial Domain Identifier (IDI). The AFI identifies the format and allocation procedures for the IDI and the format of the remainder of the NSAP Address.

Note 2.— The ATN Network Addressing Domain is such a sub-ordinate Network Addressing Domain and has an IDP that uses an ISO 6523-ICD IDI.

Note 3.— The IDP is always expressed as decimal digits. However, ISO/IEC 8348 permits NSAP Addresses in an ISO 6523-ICD domain to have either a binary or a decimal format for the remainder of the address - the Domain Specific Part (DSP). The format of the DSP is determined by the AFI.

5.4.3.4.1 All ATN NSAP Addresses shall have an AFI with the value 47 decimal.

Note.— This AFI value is defined by ISO/IEC 8348 to imply an ISO 6523-ICD IDI with a binary format DSP.

5.4.3.4.2 All ATN NSAP Addresses shall have an IDI value of 0027 decimal.

Note.— This value has been allocated by ISO to ICAO under the ISO 6523-ICD scheme. An IDP of 470027 therefore forms the common NSAP Address Prefix to all ATN NSAP Addresses and effectively defines the ATN Network Addressing Domain, as a sub-domain of the Global Network Addressing Domain.

5.4.3.5 The Reference Publication Format

Note.— The Reference Publication Format is defined by ISO/IEC 8348 for the publication of NSAP Addresses and NETs in a form suitable for text documents.

5.4.3.5.1 Recommendation.— *For the purposes of publication in a text format, ATN NSAP Addresses and NETs should be written as the character sequence “470027+”, identifying the common prefix for all ATN NSAP Addresses, followed by the DSP expressed as a sequence of hexadecimal characters.*

Note.— The “+” sign is used as a separator between the decimal syntax IDP and the hexadecimal syntax DSP.

5.4.3.5.2 Each successive pair of hexadecimal digits shall correspond to the next binary octet of the DSP.

5.4.3.6 The ATN NSAP Address Format

Note 1.— The derivation of the ATN NSAP Address Format is illustrated in Figure 5.4-3. This starts with the AFI and IDI fields required by ISO/IEC 8348. It ends with the System ID (SYS) and SEL fields required by ISO/IEC 10589. The remaining DSP fields are specified below and used to co-ordinate the allocation of ATN NSAP Addresses.

Note 2.— The VER field is used to partition the ATN Addressing Domain into a number of sub-ordinate addressing domains, each of which provides a different approach to address management.

Note 3.— The ADM field is then used to break down each such partition into a number of sub-ordinate addressing domains, each of which may then be managed by a different manager.

Note 4.— In Fixed Network Addressing Domains, the ARS field may then be used to identify a Network Addressing Domain that will correspond to each Routing Domain under the control of each such manager, and the LOC field may then be used to identify each Routing Area within each Routing Domain.

Note 5.— In Mobile Network Addressing Domains, the ARS field identifies an aircraft. Where all ATN systems onboard an aircraft form a single Routing Domain, the ARS field also identifies the Addressing Domain that will correspond to that Routing Domain, and the LOC field is used as above. However, when the ATN systems onboard a single aircraft form more than one Routing Domain, then part of the LOC field is also used to identify such an Addressing Domain.

Note 6.— The reason for the existence of the RDF field is historical.

5.4.3.7 NSAP Address Encoding

Note 1.— In ISO/IEC 8348 terms, the IDP has an abstract decimal syntax, and the DSP has an abstract binary syntax. The reason for the use of the word abstract is to emphasise the fact that the actual encoding is outside of the scope of ISO/IEC 8348, and instead is the responsibility of the standards that specify the encoding of network layer protocols.

Note 2.— ISO/IEC 8348 does, however, describe two possible encoding schemes, the “preferred binary encoding” and the “preferred decimal encoding”. ISO/IEC 8473 mandates the use of the preferred binary encoding for CLNP, while ISO/IEC 10747 mandates a modified version of the preferred binary encoding in order to cope with bit aligned NSAP Address Prefixes.

Note 3.— In consequence, this specification only specifies how each field of the DSP is allocated as an unsigned binary number. The actual encoding of the resulting bitstring in an NPDU is then according to the applicable protocol specification.

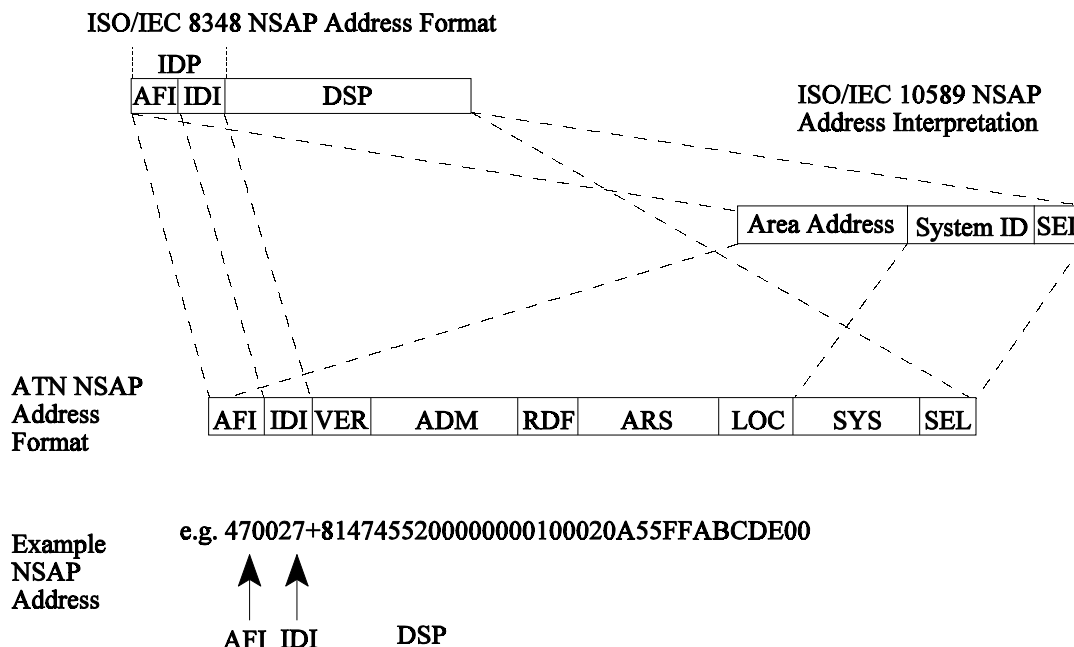


Figure 5.4-3 Derivation of the ATN NSAP Address Format

5.4.3.8 Allocation of the DSP

Note.— The DSP fields of an ATN NSAP Address are the VER, ADM, RDF, ARS, LOC, SYS and SEL fields. The size of each of these fields is given in Table 5.4-1.

5.4.3.8.1 The Version (VER) Field

Note 1.— The purpose of the VER field is to partition the ATN Network Addressing Domain into a number of sub-ordinate Addressing Domains.

Note 2.— The values currently specified for the VER Field and the Network Addressing Domains so defined, are summarised in Table 5.4-2.

5.4.3.8.1.1 The VER Field shall be one octet in length.

5.4.3.8.1.2 A VER field value of [0000 0001] shall be used for all NSAP Addresses and NETs in the Network Addressing Domain that comprises all Fixed AINSC NSAP Addresses and NETs.

Note.— The NSAP Address Prefix “470027+01” is therefore the common NSAP Address Prefix for the Fixed AINSC Network Addressing Domain.

5.4.3.8.1.3 A VER field value of [0100 0001] shall be used for all NSAP Addresses and NETs in the Network Addressing Domain that comprises all Mobile AINSC NSAP Addresses and NETs.

Note.— The NSAP Address Prefix “470027+41” is therefore the common NSAP Address Prefix for the Mobile AINSC Network Addressing Domain.

Address Field Name	Address Field Size
VER	1 Octet
ADM	3 Octets
RDF	1 Octet
ARS	3 Octets
LOC	2 Octets
SYS	6 Octets
SEL	1 Octet

Table 5.4-1 DSP NSAP Address Field Sizes

5.4.3.8.1.4 A VER field value of [1000 0001] shall be used for all NSAP Addresses and NETs in the Network Addressing Domain that comprises all Fixed ATSC NSAP Addresses and NETs.

Note.— The NSAP Address Prefix “470027+81” is therefore the common NSAP Address Prefix for the Fixed ATSC Network Addressing Domain.

5.4.3.8.1.5 A VER field value of [1100 0001] shall be used for all NSAP Addresses and NETs in the Network Addressing Domain that comprises all Mobile ATSC NSAP Addresses and NETs.

Note.— The NSAP Address Prefix “470027+C1” is therefore the common NSAP Address Prefix for the Mobile ATSC Network Addressing Domain.

5.4.3.8.1.6 All other VER field values shall be reserved.

VER Field Value	Network Addressing Domain	Common NSAP Address Prefix for Domain
[0000 0001]	Fixed AINSC	470027+01
[0100 0001]	Mobile AINSC	470027+41
[1000 0001]	Fixed ATSC	470027+81
[1100 0001]	Mobile ATSC	470027+C1

Table 5.4-2 VER Field Assigned Values

5.4.3.8.2 The Administration (ADM) Field

5.4.3.8.2.1 General

Note.— The purpose of the ADM field is to sub-divide each of the Network Addressing Domains introduced by the VER field into a further set of sub-ordinate Network Addressing Domains, and to permit devolved administration (i.e. address allocation) of each resulting domain to an individual State or Organisation.

5.4.3.8.2.1.1 The ADM field shall be three octets in length.

5.4.3.8.2.2 Fixed AINSC NSAP Addresses and NETs

Note.— In the Fixed AINSC Network Addressing Domain, the ADM field is used to sub-divide this Addressing Domain into a number of sub-ordinate Network Addressing Domains, each of which comprises NSAP Addresses and NETs for fixed systems operated by a single AINSC Organisation.

5.4.3.8.2.2.1 Allocation of NSAP Addresses and NETs in each such Network Addressing Domain subordinate to the Fixed AINSC Network Addressing Domain shall be the responsibility of the organisation identified by the value of the ADM field.

5.4.3.8.2.2.2 **Recommendation.**— *The field value should be derived from the set of three-character alphanumeric symbols representing an IATA Airline or Aeronautical Stakeholder Designator, according to 5.4.1.4.*

Note.— AINSC Organisations are intended to register their ADM values with IATA.

5.4.3.8.2.3 Fixed ATSC NSAP Addresses and NETs

Note.— In the Fixed ATSC Network Addressing Domain, the ADM field is used to sub-divide this Addressing Domain into a number of sub-ordinate Network Addressing Domains, each of which comprises NSAP Addresses and NETs for fixed systems operated by a single State or within an ICAO Region.

5.4.3.8.2.3.1 Allocation of NSAP Addresses and NETs in each such Network Addressing Domain subordinate to the Fixed ATSC Network Addressing Domain shall be the responsibility of the State or ICAO Region identified by the value of the ADM field.

5.4.3.8.2.3.2 When used for identifying a State, the ADM field shall be derived from the State's three-character alphanumeric ISO 3166 Country Code, represented as upper case characters.

5.4.3.8.2.3.3 In this case, the value of the field shall be determined according to 5.4.1.4.

Note.— For example, the encoding of 'GBR' is 474252 in hexadecimal. Therefore the NSAP Address Prefix 470027+81474252 is the common NSAP Address Prefix for all NSAP Addresses and NETs in the UK Fixed ATSC Network Addressing Domain.

5.4.3.8.2.3.4 When used to identify an ICAO Region, the first octet of the ADM field shall identify the ICAO Region, according to Table 5.4-3, while the values of the remaining two octets shall be assigned by the identified ICAO Region.

Note 1.— The ISO 3166 character codes are always represented as binary octets, each of which has a zero most significant bit. Therefore, it is possible to guarantee that the field values listed in Table 5.4-3 do not conflict with ISO 3166 derived State Identifiers.

ADM Field First Octet	ICAO Region
[1000 0000]	Africa
[1000 0001]	Asia
[1000 0010]	Caribbean
[1000 0011]	Europe
[1000 0100]	Middle East
[1000 0101]	North America
[1000 0110]	North Atlantic
[1000 0111]	Pacific
[1000 1000]	South America

Table 5.4-3 ICAO Region Identifiers

Note 2.— This Addressing Plan enables ICAO Regions to allocate ADM field values in the Fixed ATSC Network Addressing Domain to States and Organisations within the ICAO Region, in a structured manner. This is in order to permit the efficient advertisement of routing information. For example, in the advertisement of routes to ‘all RDs in the same ATN Island’ as recommended in 5.3.7.1.4.2.

5.4.3.8.2.3.5 All ADM field values in the Fixed ATSC Network Addressing Domain that do not correspond to valid ISO 3166 Country Codes or which are not assigned to ICAO Regions shall be reserved.

5.4.3.8.2.4 Mobile NSAP Addresses and NETs

Note.— In both the Mobile AINSC and the Mobile ATSC Network Addressing Domains, the ADM field is used to sub-divide this Addressing Domain into a number of sub-ordinate Network Addressing Domains, each of which comprises NSAP Addresses and NETs for mobile systems operated by a single Airline or onboard the General Aviation aircraft of a single State.

5.4.3.8.2.4.1 For Mobile AINSC NSAP Address and NETs, the ADM field value shall be set according to 5.4.3.8.2.2, and the corresponding sub-ordinate Network Addressing Domain administered by the organisation identified by the value of the ADM field.

5.4.3.8.2.4.2 For Mobile ATSC NSAP Address and NETs, the ADM field value shall be set according to 5.4.3.8.2.3, and the corresponding sub-ordinate Network Addressing Domain administered by the State identified by the value of the ADM field.

5.4.3.8.3 The Routing Domain Format (RDF) Field

Note 1.— There is no absolute requirement for the remainder of the DSP in each of the above defined Network Addressing Domains to be allocated according to a co-ordinated addressing plan, or for even the same fields to exist, or the NSAP Addresses to have the same length. However, in order to encourage common equipment development, this specification specifies the existence, size and use of the RDF, ARS and LOC fields.

Note 2.— The reason for the existence of the RDF field is historical.

5.4.3.8.3.1 The RDF field shall be one octet in length and its value shall be [0000 0000] in binary.

5.4.3.8.3.2 All other values shall be reserved.

5.4.3.8.4 The Administrative Region Selector (ARS) Field

Note 1.— In Fixed Network Addressing Domains, the purpose of the ARS field is to distinguish Routing Domains operated by the same State or Organisation.

Note 2.— In Mobile Network Addressing Domain, the purpose of the ARS field is to identify the aircraft on which the addressed system is located. When the systems onboard an aircraft form a single Routing Domain, then the ARS field also identifies the Routing Domain. When the systems onboard an aircraft form multiple RDs, then part of the LOC field is used to distinguish them.

5.4.3.8.4.1 The ARS field shall be three octets in length.

5.4.3.8.4.2 In the Fixed AINSC and ATSC Network Addressing Domains, the value of the ARS field shall be a 24-bit unsigned binary number that uniquely identifies the NSAP Addresses and NETs assigned to systems in a single Routing Domain.

5.4.3.8.4.3 In the Fixed AINSC and ATSC Network Addressing Domains, the State or Organisation identified by the value of the ADM field shall be responsible for assigning the ARS field.

Note 1.— For example, 470027+8147425200000000 and 470027+8147425200000001 are therefore NSAP Address Prefixes common to all NSAP Addresses and NETs assigned to fixed systems in two distinct Routing Domains operated by the UK ATSC authority.

Note 2.— Where necessary, the allocation of NSAP Addresses and NETs may thus readily be delegated to a Network Administrator responsible for each Network Addressing Domain that corresponds to each Routing Domain.

5.4.3.8.4.4 In Mobile AINSC and ATSC Network Addressing Domains, the value of the ARS field shall be the 24-bit ICAO Aircraft Address that uniquely identifies the NSAP Addresses and NETs in a single Routing Domain.

Note 1.— If the aircraft is operated by an IATA Airline then the NSAP Address or NET is in a Mobile AINSC Network Addressing Domain.

Note 2.— For General Aviation Aircraft, the NSAP Address or NET is in a Mobile ATSC Network Addressing Domain.

5.4.3.8.5 The Location (LOC) Field

Note 1.— In Fixed Network Addressing Domains, the purpose of the LOC field is to distinguish Routing Areas within the same Routing Domain.

Note 2.— In Mobile Network Addressing Domains, the LOC field is used

- a) *to distinguish Routing Areas within the same Mobile Routing Domain, or,*
- b) *when more than one Routing Domain is located on a single Aircraft, to distinguish each Routing Domain and the Routing Areas contained within them.*

Note 3.— For example, the first octet of the LOC field may be used to distinguish each Routing Domain on board a single aircraft, and the second octet to distinguish each Routing Area.

Note 4.— The combination of AFI, IDI, VER, ADM, RDF, ARS and LOC fields therefore forms an Area Address.

5.4.3.8.5.1 The LOC field shall be two octets in length and may be given any binary value.

5.4.3.8.5.2 The administrator of the Network Addressing Domain that co-incides with the Routing Domain in which a given Routing Area is located, shall be responsible for the allocation of a LOC field value that provides a unique Area Address for that Routing Area.

Note.— For example, 470027+81474252000000010045 is an Area Address in a Routing Domain operated by the UK ATSC Administration.

5.4.3.8.6 The System Identifier (SYS) Field

Note.— ISO/IEC 10589 defines the System Identifier as a variable length field which uniquely identifies an End or Intermediate System within a ISO/IEC 10589 Routing Area. Within a Routing Area, all System Identifiers are of the same length, although a Router is not able to make assumptions about the length of this field outside of its own Routing Area. However, the ATN Addressing Plan does specify this field to always be six octets in length in order to encourage a common equipment base.

5.4.3.8.6.1 In an ATN NSAP Address or NET, the System Identifier (SYS field) shall be six octets in length.

5.4.3.8.6.2 The value of the SYS field shall be a unique binary number assigned by the addressing authority responsible for the Network Addressing Domain that corresponds with the Routing Area in which the identified system is located.

Note.— If the System is attached to an IEEE 802 Local Area Network (e.g. an Ethernet), then a common approach is to use the 48-bit LAN address as the value of the SYS field.

5.4.3.8.7 The NSAP Selector (SEL) Field

Note.— The NSAP Selector (SEL) field identifies the End System or Intermediate System network entity or network service user process responsible for originating or receiving Network Service Data Units (NSDUs).

5.4.3.8.7.1 The SEL field shall be one octet in length.

5.4.3.8.7.2 The SEL field value for an Intermediate System network entity shall be [0000 0000], except for the case of an airborne Intermediate System implementing the procedures for the optional non-use of IDRP.

5.4.3.8.7.3 In the case of an airborne Intermediate System implementing the procedures for the optional non-use of IDRP, the SEL field value shall be [1111 1110].

5.4.3.8.7.4 The SEL field value [1111 1111] shall be reserved.

Note 1.— In an Intermediate System, any other SEL field value may be assigned to NSAPs. The actual value chosen is a local matter.

Note 2.— SEL field values in stand-alone End Systems (i.e. in End Systems not co-located with Intermediate Systems) are not constrained.

5.4.3.8.7.5 SEL field values other than those defined for Intermediate System Network Entities in 5.4.3.8.7.2 and 5.4.3.8.7.3 above or being reserved, shall be assigned by the addressing authority responsible for the identified End or Intermediate System.

5.4.3.9 Pre-Defined NSAP Address Prefixes

5.4.3.9.1 All AINSC Mobiles

5.4.3.9.1.1 The NSAP Address Prefix 470027+41 shall provide a common NSAP Address Prefix for all AINSC Mobiles.

5.4.3.9.2 All ATSC Mobiles

5.4.3.9.2.1 The NSAP Address Prefix 470027+C1 shall provide a common NSAP Address Prefix for all ATSC Mobiles.

Note.— The NLRI for the Default Route to all Mobiles comprises both the NSAP Address Prefixes defined above.

5.4.3.9.3 All Aircraft Belonging to an Airline

5.4.3.9.3.1 The NSAP Address Prefix 470027+41 plus the value of the ADM field assigned to the airline shall provide a common NSAP Address Prefix for all AINSC Mobiles operated by a single airline.

Note.— The NLRI for the Route to the “Home” for the aircraft belonging to a given airline contains this NSAP Address Prefix.

5.4.3.9.4 All General Aviation and Other Types of Aircraft Registered by a State

5.4.3.9.4.1 The NSAP Address Prefix 470027+C1 plus the value of the ADM field assigned to the State shall provide a common NSAP Address Prefix for all ATSC Mobiles registered by a single State.

Note.— The NLRI for the Route to the “Home” for the General Aviation and other types of aircraft registered by a single State contains this NSAP Address Prefix.

5.5 TRANSPORT SERVICE AND PROTOCOL SPECIFICATION

5.5.1 General

5.5.1.1 Overview

5.5.1.1.1 The COTP (Connection Oriented Transport Protocol) shall be used to provide an end-to-end reliable data transfer service between Transport Service users on two ATN End Systems.

5.5.1.1.2 In ATN End Systems, the implementation of the COTP shall conform to ISO/IEC 8073 and the mandatory requirements given in this Chapter.

5.5.1.1.3 The CLTP (Connectionless Mode Transport Protocol) shall be used to provide a Connectionless data transfer service between Transport Service users on two ATN End Systems.

5.5.1.1.4 In ATN End Systems, the implementation of the CLTP shall conform to ISO/IEC 8602 and the mandatory requirements given in this chapter.

Note.— The transport protocols specified for use in ATN End Systems provide both Connection Mode and Connectionless Mode communication services. The implementation and use of a particular mode of the Transport Layer service depends on the requirements of the application(s) supported by a given ATN End System.

5.5.1.2 Transport Service Description

Note 1.— When the TS-user requires use of the connection mode transport service the TS-user will provide the following information to the TS-provider on a per Transport Connection basis:

- a) *called and calling TSAP address;*
- b) *whether or not the expedited data option is required;*
- c) *the required residual error rate (RER) to determine whether use or non-use of the transport checksum is allowed;*
- d) *the Application Service Priority to be mapped into the resulting CLNP NPDUs according to Table 1-2;*
- e) *the ATN Security Label specifying the ATN Traffic Type, i.e.*
 - *ATN Operational Communications;*
 - *ATN Administrative Communications;*
 - *General Communications;*
 - *ATN Systems Management Communications.*

Note 2.— In the case where the Traffic Type specified is ATN Operational Communications the TS-user will additionally provide the traffic category, i.e. Air Traffic Services Communications (ATSC) or Aeronautical Operational Control (AOC).

Note 3.— In the case of the ATSC traffic category the TS-user will further specify the required ATSC Class as defined in Table 1-1, or no traffic type policy preference.

Note 4.— In the case of the AOC traffic category the TS-user will further specify the subnetwork preference (including no preference).

Note 5.— The ATN Traffic Types and their associated traffic categories are specified in Table 5.6-1. The encoding of the ATN Security Label is specified in Figure 5.6-1 and 5.6.2.2.2.2 bullet b).

Note 6.— The TS-user is not required to specify any other Transport Service Quality of Service parameters.

5.5.1.3 Transport Service Access Point Addresses

5.5.1.3.1 A TSAP address shall comprise two elements, a NSAP address and a TSAP selector.

5.5.1.3.2 The NSAP address and the TSAP selector shall conform to the provisions in 5.4.

5.5.1.4 Exchange of Transport-Selector parameters

Note.— TSAP Selectors are transmitted in Calling and Called Transport-Selector parameters in COTP, and in Source and Destination Transport-Selector parameters in CLTP.

5.5.1.4.1 The transport entity shall support Transport-Selector parameters to accommodate the ATN TSAP selector syntax and encoding requirements as specified in 5.4.

5.5.1.4.2 **Recommendation.**— *The transport entity should support remote Transport-Selector parameters of variable size from 0 up to 32 octets using any encoding and any value.*

Note.— The absence of a Calling and Called Transport-Selector assumes the Network Address alone unambiguously defines the Transport Address.

5.5.1.4.3 In COTP, on receipt of CR (Connection Request) TPDU, the absence of a Calling or Called Transport-Selector shall be treated as equivalent to a zero length Calling or Called Transport-Selector.

5.5.1.4.4 The absence of a Calling or Called Transport-Selector in a received CC (Connection Confirm) TPDU shall indicate that Calling or Called Transport-Selector is equivalent to the corresponding parameter specified in the sent CR TPDU.

5.5.1.4.5 When present in a received CC TPDU, Calling and Called Transport-Selector parameters shall be identical in length and value to the corresponding parameter specified in the sent CR TPDU.

5.5.1.4.6 In CLTP, on receipt of UD (User Data) TPDU, the absence of a Source or Destination Transport-Selector shall be treated as equivalent to a zero length Source or Destination Transport-Selector.

5.5.2 Connection Mode Transport Layer Operation

5.5.2.1 Connection Mode Transport Service Primitives

Note 1.— For the purpose of describing the notional interfaces between different OSI protocol layers, each protocol layer is assumed to provide a service to the next higher protocol layer. The assumed service provided by the ATN transport layer to its user is described in ISO/IEC 8072.

Note 2.— ATN Applications may specify their use of the COTP implemented in ATN End Systems using the Transport Service specified in ISO/IEC 8072, including use of ATN priority, and security parameters as defined in this specification.

Note 3.— There is no requirement to implement the service specified in ISO/IEC 8072 as a software interface.

5.5.2.2 ATN Specific Requirements

5.5.2.2.1 ATN End Systems shall implement the ISO/IEC 8073 Class 4 transport protocol in order to provide Connection Mode communications over the ATN Internet.

5.5.2.2.2 The COTP shall operate using the CLNS (Connectionless Network Service) as specified in 5.6.

Note.— TPDUs (Transport Protocol Data Units) are sent via the N-UNITDATA Request primitive.

5.5.2.2.3 The transport entity shall not concatenate TPDUs from TCs with different transport priorities or different Security Labels.

5.5.2.2.4 **Recommendation.**— *The Selective Acknowledgement mechanism should be used for conservation of bandwidth by preventing retransmission of correctly received out-of-sequence TPDUs.*

5.5.2.2.5 **Recommendation.**— *The Request of Acknowledgement mechanism should be used to reduce AK traffic.*

5.5.2.2.6 **Recommendation.** — *The maximum TPDU size should be at least 1024 octets.*

Note.— This is to support efficient transmission of anticipated application data exchanges.

5.5.2.2.7 **Recommendation.**— *The Transport Layer should propose a TPDU size of at least 1024 octets.*

5.5.2.2.8 **Recommendation.**— *The Transport Layer should use the TPDU size parameter rather than the preferred maximum TPDU size parameter.*

5.5.2.2.9 **Recommendation.**— *Implementations of the ATN Transport Layer should propose use of normal format in the CR TPDU.*

5.5.2.2.10 **Recommendation.**— *The Extended format should only be proposed when explicitly necessary to meet application Quality of Service requirements.*

Note.— Because the increased TPDU size resulting from use of extended data TPDU numbering may be more inefficient, this option is used on a TC only when absolutely required.

5.5.2.2.11 **Recommendation.**— *The Transport Layer should accept non-use of checksum when proposed in a CR TPDU.*

5.5.2.2.12 Implementations of the transport protocol shall support configurable values for all timers and protocol parameters, rather than having fixed values, in order to allow modification as operational experience is gained.

5.5.2.2.13 When intended for operation over Air/Ground subnetworks, transport protocol implementations shall support the minimum - maximum ranges for COTP timer values as presented in Table 5.5-1.

5.5.2.2.13.1 **Recommendation.**— *Nominal values indicated in Table 5.5-1 should be used.*

5.5.2.2.13.2 **Recommendation.**— *The assignment of optimized values for timers and parameters other than the nominal values indicated in Table 5.5-1 should be based on operational experience.*

5.5.2.2.14 **Recommendation.**— *When intended for operation exclusively over Ground/Ground subnetworks, implementations of transport protocol timer values should be optimized to ensure interoperability.*

Table 5.5-1 COTP Timer Value Ranges

Name	Description	Minimum Value	Nominal Value	Maximum Value
M_{RL}, M_{LR}	NSDU Lifetime, seconds	26	400	600
E_{RL}, E_{LR}	Maximum Transit Delay, seconds	1	100	150
A_L, A_R	Acknowledgement Time, seconds	1	20	400
T1	Local Retransmission Time, seconds	12	221	300
R	Persistence Time, seconds	1	443	2710
N	Maximum Number of Transmissions	1	3	10
L	Time bound on reference and/or sequence numbers, seconds	160	1263	3000
I	Inactivity Time, seconds	600	4500	6000
W	Window Time, seconds	160	4000	6000

Note 4.— In Table 5.5-1, the subscripts “R” and “L” refer to “remote” and “local” respectively. The variable E_{RL} , for example, refers to the maximum transit delay from the remote entity to the local entity. The variable E_{LR} is the maximum transit delay from the local entity to the remote entity. It is assumed that these values may be different.

Note 5.— Several of the timers and variables listed in Table 5.5-1 are not directly configurable, but may be determined based on the values of other timers and variables. These computed values are:

$$T1 = (E_{LR} + E_{RL} + A_R + x)$$

$$R = (T1 * (N-1) + x)$$

$$L = (M_{LR} + M_{RL} + R + A_R)$$

$$W = (I - E_{LR} - \text{offset})$$

x = Local processing time

offset = Unanticipated delay exceeding E_{LR} values

5.5.2.3 Connection Mode Transport Quality of Service

5.5.2.3.1 Connection Mode Transport Priority

5.5.2.3.1.1 The Transport Layer shall allow a TC (Transport Connection) priority in the range [0 - 14]

5.5.2.3.1.2 The Transport Layer shall not alter the proposed TC priority specified by the TS-user.

5.5.2.3.1.3 The Transport Layer shall treat all connections without expressed priority as being at the default TC priority.

5.5.2.3.1.4 The default TC priority shall be the lowest priority, i.e. priority [14].

5.5.2.3.1.5 When a TS-user specifies a TC priority, the relationship between this TC priority and the CLNP priority shall be as specified in Table 1-2.

5.5.2.3.2 Connection Mode Transport Security

Note.— The ATN security mechanism does not make use of the ISO/IEC 8073 Protection parameter. The support of the Protection parameter is therefore optional.

5.5.2.3.2.1 The Transport Layer shall allow a TS-user to specify a Security Label for a transport connection. The transport security procedure shall be implemented as specified in 5.2.7.3.1.

5.5.2.3.2.2 The Security Label format shall be according to 5.2.7.1. The Transport Layer shall not alter the Security Label specified by the TS-user.

Note.— When no Security Label is present, a « General Communications » traffic type is implied. In this case, CLNP NPDUs are generated without the Security parameter.

5.5.2.4 Encoding of Transport Protocol Data Units

5.5.2.4.1 General

5.5.2.4.1.1 The encoding of TPDU's shall conform to ISO/IEC 8073 for the COTP.

5.5.2.4.2 Encoding of the Acknowledgment Time Parameter

5.5.2.4.2.1 In ATN compliant systems, the acknowledgement time parameter of the CR and CC TPDU's shall be encoded as follows:

Parameter Code: 1000 0101

Note 1.— This is identical to the ISO/IEC 8073 standard parameter.

Parameter Length: 2 or 3 octets.

Parameter Value: Acknowledgment Timer (A_L) value expressed in milliseconds (per ISO/IEC 8073 standard)

Note 2.— This enhancement is in response to the unique requirements of the aeronautical environment which may require longer acknowledgment times than foreseen in ISO/IEC 8073.

Note 3.— Initial values of these timers may depend upon the subnetwork, traffic type and routing policy requirements expressed in the associated ATN Security Label.

Note 4.— In cases where the A_L value is expressed in 2 octets (less than 65536 milliseconds), the ATN implementation will behave in compliance with the ISO/IEC 8073 standard.

Note 5.— Implementors are advised to permit systems administrators to readily specify initial values.

5.5.2.5 Transport Layer Congestion Avoidance

5.5.2.5.1 General

Note 1.— The congestion avoidance mechanisms in the Transport Layer make use of the notification by the Network Layer of Congestion Experienced flags in received NPDUs. This mechanism allows transport entities to reduce the window, i.e. the number of DT TPDU's allowed to be sent without acknowledgement, when the proportion of NPDUs indicating congestion reaches a certain threshold.

Note 2.— This congestion information consists of the total length of the sequence of NPDUs forming the associated NSDU, and the number of NPDUs of that sequence that had their congestion experienced flag set upon reception.

Note 3.— Transport Congestion Avoidance measures are applicable to the Connection Mode transport service only.

5.5.2.5.1.1 The transport entity shall implement the congestion avoidance algorithm defined in this section.

5.5.2.5.1.2 This algorithm shall be applied for each transport connection individually.

5.5.2.5.2 Advertised Window

5.5.2.5.2.1 General

5.5.2.5.2.1.1 A receiving transport entity shall provide the sending transport entity with the lower window edge and the size of the advertised window (W) by using the explicit flow control mechanisms specified in ISO/IEC 8073.

*Note.— The **advertised window** is the window advertised by the receiver of the data to the sender of the data. It indicates the number of DT TPDUs that the receiver is willing to accept.*

5.5.2.5.2.2 Initialisation of the Advertised Window

5.5.2.5.2.2.1 The initial value of the window W_0 that is advertised to the sending transport entity shall have a locally configurable value.

5.5.2.5.2.2.2 This initial window shall be sent to the sending transport entity in the first CDT field transmitted.

5.5.2.5.3 Receiving Transport Entity Congestion Avoidance

5.5.2.5.3.1 General

5.5.2.5.3.1.1 Congestion avoidance shall be performed within repeated update phases.

5.5.2.5.3.1.2 Each update phase shall terminate with the possible advertisement of a new window size to the sending transport entity.

5.5.2.5.3.2 Start of Update Phase

5.5.2.5.3.2.1 An update phase of the advertised window shall start after the receiving transport entity has advertised a new value of the window W_{new} to the sending transport entity.

5.5.2.5.3.3 Ignoring Congestion Information

5.5.2.5.3.3.1 After having advertised a new window size, the receiving transport entity shall ignore congestion information coming from the Network Layer, until it has received W (i.e. the « old » advertised window size) further DT-TPDUs. It then shall enter the sampling sub-phase.

5.5.2.5.3.3.2 When the sending transport entity advertises the initial window size W_0 , it shall set W to 0.

5.5.2.5.3.4 Sampling Congestion Information

5.5.2.5.3.4.1 The receiving transport entity shall maintain a count N equal to the total number of NPDUs that convey DT-TPDUs, and a count NC equal to the number of such NPDUs that had their congestion experienced flag set upon reception.

5.5.2.5.3.4.2 Upon entering the sampling sub-phase, these counts shall be reset to zero.

5.5.2.5.3.4.3 These counts shall be updated upon receipt of a DT-TPDU using the congestion information supplied by the Network Layer.

5.5.2.5.3.4.4 The sampling sub-phase shall end as soon as the transport entity has received W_{new} DT-TPDUs within the sampling sub-phase. The end of the sampling sub-phase also terminates the update phase.

5.5.2.5.3.5 Action Upon the End of the Update Phase

5.5.2.5.3.5.1 The receiving transport entity shall take the following actions at the end of each update phase:

- a) If the count NC is less than λ % of the count N , the receiving transport entity shall increase the size of the advertised window by adding δ up to a maximum based on the local buffer management policy. Otherwise, it shall decrease the size of the advertised window by multiplying it by β . If the result of this multiplication has a decimal part, the new window size shall be truncated to its integer value. The size of the advertised window shall not go to a value smaller than 1.
- b) The counts N and NC shall be reset to 0.
- c) The new window size shall be transmitted to the sending transport entity in accordance with the explicit flow control mechanisms specified in ISO/IEC 8073.

Note.— This procedure does not explicitly require the reduction of the upper window edge, as it is possible to gradually reduce the credit window.

5.5.2.5.4 Recommended Algorithm Values

5.5.2.5.4.1 **Recommendation.**— *The value settings defined in the following table should be implemented and configurable by a System Manager:*

Table 5.5-2. Congestion Avoidance algorithm values

Name	Description	Recommended value/range
β	Window decrease factor	0.75 to 0.95
δ	Window increase amount	1
W_0	Initial window	1
λ	Congestion ratio	50%

5.5.2.6 Use of the ATN Network Service

Note.— This section specifies how the COTP operates over the CLNS provided by the ATN Network Layer.

5.5.2.6.1 Use of the N-UNITDATA Request

5.5.2.6.1.1 General

5.5.2.6.1.1.1 The Transport Layer shall use the N-UNITDATA Request primitive, as defined in ISO/IEC 8073, to transmit TPDU(s).

Note.— The way the parameters are exchanged between the transport entity and the Network Service is a local matter.

5.5.2.6.1.1.2 The length indication given to the network service shall be equal to the length of the TPDU(s).

Note.— The maximum size of each TPDU is restricted to the locally defined maximum NSDU size.

5.5.2.6.1.2 NS-user-data

Note.— Transport entities transmit TPDU(s) as NS-user-data of the N-UNITDATA Request primitive.

5.5.2.6.1.3 Network Service Access Point Addresses

Note.— The Transport Layer has knowledge of the source and destination address parameters only as octet strings.

5.5.2.6.1.4 Network Quality of Service

5.5.2.6.1.4.1 General

5.5.2.6.1.4.1.1 The COTP shall use the network QoS parameters as defined in the sections below.

5.5.2.6.1.4.2 Network Layer Priority

5.5.2.6.1.4.2.1 The COTP shall use the network priority parameter to indicate the relative priority of a NSDU.

5.5.2.6.1.4.2.2 When a transport priority has been specified, the value of network priority shall be determined based on the transport connection priority, as defined in Table 1-2.

5.5.2.6.1.4.2.3 If the Transport Layer supports levels of TC priority numerically greater than 14, TPDU(s) associated with the TC shall be transmitted using a network priority level of zero.

Note.— As specified in ISO/IEC 8073, the Transport Layer priority level zero is highest. ISO/IEC 8473 specifies zero as the lowest network priority and fourteen as the highest. Table 1-2 defines the required mapping between these two schemes for use by ATN systems.

5.5.2.6.1.4.3 Network Layer Security

Note.— The use of the Network Layer security is specified in 5.2.7.3.1.

5.5.2.6.2 Use of the N-UNITDATA Indication

5.5.2.6.2.1 General

5.5.2.6.2.1.1 The Transport Layer shall be capable of receiving TPDUs from the ATN network service using the N-UNITDATA indication primitive, as defined in ISO/IEC 8073.

Note.— The way the parameters are exchanged between the transport entity and the Network Service is a local matter.

5.5.2.6.2.2 NS-user-data

Note.— Transport entities receive TPDUs as NS-user-data of the N-UNITDATA Indication primitive.

5.5.2.7 Connection Mode Transport APRL

5.5.2.7.1 Mandatory and Optional Functions

5.5.2.7.1.1 General

Note.— The requirements for the COTP are provided in the form of an ATN Protocol Requirements List (APRL). The APRL has been prepared using the PICS (Protocol Implementation Conformance Statement) proforma provided with ISO/IEC 8073.

5.5.2.7.1.1.1 An implementation of the ISO/IEC 8073 Transport Protocol shall be used in an ATN End System if and only if its PICS is in compliance with the APRL provided with these SARPs.

5.5.2.7.1.2 Protocol Implementation

5.5.2.7.1.2.1 Classes Implemented

Index	Class	ISO/IEC 8073 References	ISO Status	ATN Support
C0	Class 0	14.2	O.1	O
C1	Class 1	14.4	C0:O	O
C2	Class 2	14.2	O.1	O
C3	Class 3	14.3	C2:O	O
C4	Class 4 operation over CONS	14.3	C2:O	O
C4L	Class 4 operation over CLNS	14.3	C2:O	M

5.5.2.7.1.2.2 Specific ATN Requirements

Index	Feature	SARPs Reference	ATN Support
ATN1	Support of Congestion Avoidance Procedures?	5.5.2.5	M
ATN2	Transport to Network Priority Mapping?	5.5.2.6.1.4.2	M
ATN3	Support of ATN Security Label?	5.5.2.6.1.4.3	M
ATN4	Configurable Transport Timers?	5.5.2.2.12	M
ATN5	Enhanced encoding of Acknowledgment Time Parameter?	5.5.2.4.2	M

5.5.2.7.1.3 Initiator/Responder Capability for Protocol Classes 0-4

Index		ISO/IEC 8073 References	ISO Status	ATN Support
IR1	Initiating CR TPDU	14.5 a)	O.2	M
IR2	Responding to CR TPDU	14.5 a)	O.2	M

5.5.2.7.1.4 Supported Functions

5.5.2.7.1.4.1 Supported Functions for Class 4 (C4 or C4L::)

5.5.2.7.1.4.1.1 Mandatory Functions for Class 4

Index	Function	ISO/IEC 8073 References	ISO Status	ATN Support
T4F1	TPDU transfer	6.2	M	M
T4F2	Segmenting	6.3	M	M
T4F3	Reassembling	6.3	M	M
T4F4	Separation	6.4	M	M
T4F5	Connection establishment	6.5	M	M
T4F6	Connection refusal	6.6	M	M
T4F7	Data TPDU numbering (normal)	6.10	M	M

Index	Function	ISO/IEC 8073 References	ISO Status	ATN Support
T4F8	Retention and acknowledgement of TPDUs (AK)	6.13.4.1	M	M
T4F9	Explicit flow control	6.16	M	M
T4F10	Checksum	6.17	M	M
T4F11	Frozen references	6.18	M	M
T4F12	Retransmission on time-out	6.19	M	M
T4F13	Resequencing	6.20	M	M
T4F14	Inactivity control	6.21	M	M

5.5.2.7.1.4.1.2 Mandatory Functions for Operation over Connectionless Network Service

Index	Function	ISO/IEC 8073 References	ISO Status	ATN Support
T4F23	Transmission over CLNS	6.1.2	M	M
T4F24	Normal release when operating over CLNS (explicit)	6.7.2	M	M
T4F25	Association of TPDUs with transport connections when operating over CLNS	6.9.2	M	M
T4F26	Expedited data transfer when operating over CLNS (Network normal)	6.11.2	M	M
T4F27	Treatment of protocol errors when operating over CLNS	6.22.2	M	M

5.5.2.7.1.4.1.3 ISO/IEC 8073 Optional Functions

Index	Feature	ISO/IEC 8073 References	ISO Status	ATN Support
T4F28	Data TPDU numbering (extended)	6.10	O	O
T4F29	Non-use of checksum	6.17	O	M
T4F30	Concatenation	6.4	O	O

Index	Feature	ISO/IEC 8073 References	ISO Status	ATN Support
T4F31	Retention and acknowledgement of TPDUs Use of selective acknowledgement	6.13.4.4	O	O
T4F32	Retention and acknowledgement of TPDUs Use of request acknowledgement	6.13.4.3	O	O

5.5.2.7.1.5 Supported TPDUs

Index	TPDUs		ISO/IEC 8073 References	ISO Status	ATN Support
ST1	CR	supported on transmission	13.1	IR1:M	M
ST2	CR	supported on receipt	13.1	IR2:M	M
ST3	CC	supported on transmission	13.1	IR2:M	M
ST4	CC	supported on receipt	13.1	IR1:M	M
ST5	DR	supported on transmission	13.1	IR2:M	M
ST6	DR	supported on receipt	13.1	IR1:M	M
ST7	DC	supported on transmission	13.1	C4L:M	M
ST8	DC	supported on receipt	13.1	C4L:M	M
ST9	DT	supported on transmission	13.1	M	M
ST10	DT	supported on receipt	13.1	M	M
ST11	ED	supported on transmission	13.1	C4L:M	MO
ST12	ED	supported on receipt	13.1	C4L:M	MO

Index	TPDUs		ISO/IEC 8073 References	ISO Status	ATN Support
ST13	AK	supported on transmission	13.1	C4L:M	M
ST14	AK	supported on receipt	13.1	C4L:M	M
ST15	EA	supported on transmission	13.1	C4L:M	MO
ST16	EA	supported on receipt	13.1	C4L:M	MO
ST19	ER	supported on receipt	13.1	M	M

Note.— The following table states for which classes, if any, ER TPDUs are supported on transmission.

Index	Class	ISO/IEC 8073 References	ISO Status	ATN Support
SER4L	ER support on transmission of Class 4 over CLNS	6.22.2	O	O

5.5.2.7.1.6 Supported Parameters of Issued TPDUs

5.5.2.7.1.6.1 Parameter Values for CR TPDUs (C4L::)

Index	feature	ISO/IEC 8073 Reference	ISO Status	ATN Support
ICR1	Bits 8 and 7 in the additional options selection parameter of a CR TPDUs set to zero?	13.3.4 g)	M	M

5.5.2.7.1.6.1.1 If the preferred class in the CR is 2,3 or 4:

Index	feature	ISO/IEC 8073 Reference	ISO Status	ATN Support
ICR2	Is class 0 always offered as an alternative class?	14.4	O	X

5.5.2.7.1.6.2 Supported parameters for Class 4 TPDUs (C4L::)

5.5.2.7.1.6.2.1 Optional Parameters for a Connection Request TPDU

Index	Supported parameters	ISO/IEC 8073 References	ISO Status	ATN Support
I4CR7	Called Transport-Selector	13.3.4 a)	O	M
I4CR8	Calling Transport-Selector	13.3.4 a)	O	M
I4CR9	TPDU size	13.3.4 b)	O	O
I4CR10	Version Number	13.3.4 d)	O	O
I4CR11	Protection parameters	13.3.4 e)	O	O
I4CR12	Additional option selection	13.3.4 g)	O	M
I4CR13	Throughput	13.3.4 k)	O	O
I4CR14	Residual error rate	13.3.4 m)	O	O
I4CR15	Priority	13.3.4 n)	O	M
I4CR16	Transit delay	13.3.4 p)	O	O
I4CR17	Acknowledgement time	13.3.4 j)	O	M
I4CR18	Preferred maximum TPDU size	13.3.4 c)	O	O
I4CR19	Inactivity timer	13.3.4 r)	O	M

5.5.2.7.1.6.2.2 Optional Parameters for a Connection Confirm TPDU

Note 1.— According to ISO, the following parameters are optional if a CC TPDU is issued in class 4:

Index	Supported parameters	ISO/IEC 8073 References	ISO Status	ATN Support
I4CC6	Called Transport-Selector	13.4.4	O	M
I4CC7	Calling Transport-Selector	13.4.4	O	M
I4CC8	TPDU size	13.4.4	O	O
I4CC9	Protection parameters	13.4.4	O	O
I4CC10	Additional option selection	13.4.4	O	M
I4CC11	Acknowledgement time	13.4.4	O	M

Index	Supported parameters	ISO/IEC 8073 References	ISO Status	ATN Support
I4CC12	Throughput	13.4.4	O	O
I4CC13	Residual error rate	13.4.4	O	O
I4CC14	Priority	13.4.4	O	M
I4CC15	Transit delay	13.4.4	O	O
I4CC16	Preferred maximum TPDU size	13.4.4	I4CR18:O	O
I4CC17	Inactivity timer	13.4.4	O	M

Note 2.— The support of T4F26 implies that the Additional Options Selection parameter is mandatory.

5.5.2.7.1.6.2.3 Optional Parameter for a Disconnect Request TPDU

Index	Supported parameter	ISO/IEC 8073 References	ISO Status	ATN Support
I4DR4	Additional information	13.5.4 a)	O	O

5.5.2.7.1.6.2.4 Mandatory Parameter for a Data TPDU

Note.— According to ISO, the following parameter is mandatory in a DT TPDU if request of acknowledgement has been selected.

Index	Supported parameter	ISO/IEC 8073 References	ISO Status	ATN Support
I4DT4	Request of acknowledgement	13.7.3 b)	T4F32:M	T4F32:M

5.5.2.7.1.6.2.5 Optional Parameter for an Acknowledgement TPDU

Note.— According to ISO, an AK TPDU containing flow control information will be transmitted if an AK TPDU is received under the conditions specified in ISO/IEC 8073 12.2.3.9. The following parameter is mandatory for ATN compliant systems if an AK TPDU is issued in Class 4.

Index	Supported parameter	ISO/IEC 8073 References	ISO Status	ATN Support
I4AK4	Flow control confirmation	13.9.4 c)	O	M

5.5.2.7.1.6.2.6 Use of the Subsequence Number Parameter in the Acknowledgement TPDU

Note.— According to ISO, if an implementation can reduce credit and does so in the manner outlined in ISO/IEC 8073 12.2.3.8.2 then the subsequence number in AK TPDU is mandatory.

Index	Supported parameters	ISO/IEC 8073 References	ISO Status	ATN Support
I4AK5	Subsequence number	13.9.4. b)	O	M

5.5.2.7.1.6.2.7 Use of the Selective Acknowledgement Parameter in the Acknowledgement TPDU

Note.— According to ISO, the following parameter is optional in an AK TPDU if selective acknowledgement has been negotiated.

Index	Supported parameter	ISO/IEC 8073 References	ISO Status	ATN Support
I4AK6	Selective acknowledgement parameters	13.9.4. d)	T4F31:O	T4F31:O

5.5.2.7.1.6.2.8 Optional Parameters for an Error TPDU

Index	Supported parameter	ISO/IEC 8073 References	ISO Status	ATN Support
I4ER3	Invalid TPDU	13.12.4 a)	O	O

5.5.2.7.1.7 Supported parameters for received TPDUs

Note.— ISO/IEC 8073 requires implementations to be capable of receiving and processing all possible parameters for all possible TPDUs, depending upon the class and optional functions implemented.

5.5.2.7.1.7.1 TPDUs in Class 4 (C4L::)

Note.— According to ISO, if use of checksum has been selected then it is mandatory to process a checksum parameter in the following TPDUs.

Index	TPDU	ISO/IEC 8073 References	ISO Status	ATN Support
R4CCch	CC TPDU	13.4.4	M	M
R4DRch	DR TPDU	13.5.4 b)	M	M
R4DCch	DC TPDU	13.6.4	M	M
R4DTch	DT TPDU	13.7.4	M	M

Index	TPDU	ISO/IEC 8073 References	ISO Status	ATN Support
R4EDch	ED TPDU	13.8.4	M	M
R4AKch	AK TPDU	13.9.4 a)	M	M
R4EAch	EA TPDU	13.10.4	M	M
R4ERch	ER TPDU	13.12.4 b)	M	M

5.5.2.7.1.8 User Data in Issued TPDUs

5.5.2.7.1.8.1 Class 4 (C4 or C4L::)

Index	User Data	ISO/IEC 8073 References	ISO Status	ATN Support
D4ICR	User data of up to 32 octets in a CR with preferred class 4	13.3.5	M	M
D4ICC	User data of up to 32 octets in a CC	13.4.5	M	M
D4IDR	User data of up to 64 octets in a DR	13.5.5	M	M

5.5.2.7.1.9 User Data in Received TPDUs

Index	User Data	ISO/IEC 8073 References	ISO Status	ATN Support
DRCC	32 octets of user data in a CC TPDU	13.4.5	IR1:M	IR1:M
DRDR	64 octets of user data in a DR TPDU	13.5.5	IR1:M	IR1:M
DRCR	32 octets of user data in a CR TPDU	13.3.5	IR2:M	IR2:M

5.5.2.7.1.10 Negotiation

Note.— If an option is not returned in the CC, it is considered to have been refused. This allows compatible negotiation between versions of the ISO/IEC 8073 transport protocol.

5.5.2.7.1.10.1 Class Negotiation - Initiator

Index	Feature	ISO/IEC 8073 References	ATN Supported Value
NC	The preferred class in the CR TPDU may contain any of the classes supported by the implementation	6.5.5 j)	Class 4

Note 1.— Negotiation of other protocol classes is out of scope. If this is the only profile supported then it is not possible to negotiate any other protocol class.

Note 2.— The table below specifies valid alternative classes.

Index	Preferred class	ISO/IEC 8073 References	ISO Allowed Values	ATN Supported Values
NAC5	Class 4 over CLNS	6.5.5 j)	None	None

Note 3.— The class cannot be negotiated since Class 4 is the only class allowed over CLNS.

5.5.2.7.1.10.2 Class negotiation - responder side

Index	Preferred class	ISO/IEC 8073 References	ISO Allowed Responses	ATN Supported Values
RC4	What classes can you respond with if CR proposes only class 4?	6.5.4 j) Table 3	2,4 or connection refused depending on classes supported	Class 4
RC4a	What classes can you respond with if CR proposes class 4 as preferred class and the alternative class parameter is present?	6.5.4 j) Table 3	0,1,2,3,4 or connection refused depending on classes supported and coding of alternative class	Class 4

Note.— This table does not preclude connection refusal for other reasons.

5.5.2.7.1.10.3 TPDU Size Negotiation

Index	TPDU size	ISO/IEC 8073 References	ISO Status	ATN Support
TS1	If maximum TPDU size is proposed in a CR TPDU then the initiator shall support all TPDU sizes from 128 octets to the maximum proposed	14.6 e)	I4CR9:M	I4CR9:M
TS2	If the preferred maximum TPDU size parameter is used in a CR TPDU then the initiator shall support all TPDU sizes, except 0, that are multiples of 128 octets up to the preferred maximum proposed	14.6 e)	I4CR18:M	I4CR18:M

Index	TPDU size	ISO/IEC 8073 References	ISO Allowed Values	ATN Supported Values
TS3	What is the largest value of the preferred maximum TPDU size parameter in a CR TPDU?	14.6 e)	any multiple of 128 octets	any multiple of 128 octets
TS4	What is the largest value of the preferred maximum TPDU size parameter in a CC TPDU?	14.6 e)	any multiple of 128 octets	any multiple of 128 octets

Note.— An implementation of the Transport Layer can support a preferred maximum TPDU size larger than 1024 octets.

Index	TPDU size	ISO/IEC 8073 References	ISO Allowed Values	ATN Supported Values
T4S1	What is the largest value of the maximum TPDU size parameter in a CR TPDU with preferred class 4?	14.6 e)	One of 128, 256, 512, 1024, 2048, 4096, 8192	One of 128, 256, 512, 1024, 2048, 4096, 8192
T4S2	What is the largest value of the maximum TPDU size parameter which may be sent in the CC TPDU when class 4 is selected?	14.6 e)	128, 256, 512, 1024, 2048, 4096, 8192	128, 256, 512, 1024, 2048, 4096, 8192

5.5.2.7.1.10.4 Use of Extended Format

Index	Extended format	ISO/IEC 8073 References	ISO Allowed Values	ATN Supported Value
NEF3	What formats can you propose in the CR TPDU in class 4?	6.5.5 n)	normal, extended	normal,extended
NEF6	What formats can you select in CC when extended has been proposed in CR in class 4?	6.5.5 n)	normal, extended	normal,extended

Note.— This table does not preclude proposal of the extended format.

5.5.2.7.1.10.5 Expedited data Transport service

Index	Expedited data	ISO/IEC 8073 References	ISO Status	ATN Supported
TED1	Is the expedited data indication supported in CR and CC TPDU?	6.5.5 r)	M	MO

Note.— Expedited data is proposed using the Additional Options Parameters in the CR and CC TPDUs.

5.5.2.7.1.10.6 Non-use of Checksum (C4L and T4F29::)

Index	Non-use of checksum	ISO/IEC 8073 References	ISO Allowed Values	ATN Supported Values
NUC1	What proposals can you make in the CR?	6.5.5 p)	non-use, use	non-use, use
NUC2	What proposals can you make in CC when non-use of checksum has been proposed in CR?	6.5.5 p)	non-use, use	non-use, use

Note 1.— A Transport Layer is able to propose either use or non-use of checksum in a CR TPDU.

Note 2.— The term “non-use” means that the Transport Layer may respond accepting non-use of checksum. A Transport Layer may also respond with use of checksum if non-use has been proposed.

5.5.2.7.1.10.7 Use of selective acknowledgement

Index	Selective Acknowledgement	ISO/IEC 8073 References	ISO Status	ATN Support
USA1	Is use of selective acknowledgement proposed in CR TPDUs ?	6.5.5 s)	O	O
USA2	Is use of selective acknowledgement selected in a CC when it has been proposed in a CR ?	6.5.5 s)	O	O

5.5.2.7.1.10.8 Use of Request Acknowledgement

Index	Request of Acknowledgement	ISO/IEC 8073 References	ISO Status	ATN Support
ROA1	Is use of request of acknowledgement proposed in CR TPDUs ?	6.5.5 t)	O	O
ROA2	Is use of request of acknowledgement selected in a CC when it has been proposed in a CR ?	6.5.5 t)	O	O

5.5.2.7.1.11 Error Handling

Note.— Using Class 4 over CLNS, a TPDU with an invalid checksum will be discarded.

5.5.2.7.1.11.1 Action on Detection of a Protocol Error

Index	Item	ISO/IEC 8073 References	ISO Allowed Values	ATN Supported Values
PE4L	Class 4 over CLNS	6.22.2.3	C4L: ER, DR, Discard	C4L: ER, DR, Discard

Note.— The choice of action (DR, Discard) is an implementation choice and may depend on the type of error encountered.

5.5.2.7.1.11.2 Actions on receipt of an invalid or undefined parameter in a CR TPDU

Index	Event	ISO/IEC 8073 References	ISO Status	ATN Support
RR1	A parameter not defined in ISO/IEC 8073 shall be ignored	13.2.3	M	M
RR2	An invalid value in the alternative protocol class parameter shall be treated as a protocol error	13.2.3	M	M
RR3	An invalid value in the class and option parameter shall be treated as a protocol error	13.2.3	M	M
RR4	On receipt of the additional option selection parameter bits 8 to 7, and bits 6 to 1 if not meaningful for the proposed class, shall be ignored	13.3.4 g)	M	M
RR6	On receipt of the class option parameter bits 4 to 1 if not meaningful for the proposed class shall be ignored	13.3.3	M	M

Index	Event	ISO/IEC 8073 Reference	ISO Allowed Value	ATN Supported Value
RR7	What action is supported on receipt of a parameter defined in ISO 8073 (other than those covered above) and having an invalid value ?	13.2.3	Ignore, Protocol Error	Ignore, Protocol Error

Note.— The choice of action (Ignore, Protocol error) is an implementation choice and may depend on the type of error encountered.

5.5.2.7.1.11.3 Actions on receipt of an invalid or undefined parameter in a TPDU other than a CR TPDU

Index	Event	ISO/IEC 8073 References	ISO Status	ATN Support
U11	A parameter not defined in ISO/IEC 8073 shall be treated as a protocol error	13.2.3	M	M

Index	Event	ISO/IEC 8073 References	ISO Status	ATN Support
U12	A parameter which has an invalid value as defined in ISO/IEC 8073 shall be treated as a protocol error	13.2.3	M	M
U13 (class 4 only)	A TPDU received with a checksum which does not satisfy the defined formula shall be discarded	6.17.3	M	M

5.5.2.7.1.12 Class 4 Timers and Protocol Parameters

Index		ISO/IEC 8073 References	ISO Status	ATN Support
TA1	T1 (Local Retransmission)	12.2.1.1.4	M	M
TA2	N (Maximum Transmission)	12.2.1	M	M
TA3	IL (Local Inactivity Time)	12.2.1.1.7	M	M
TA4	W (Window Update)	12.2.1	M	M
TA5	L (Frozen Reference Time)	12.2.1.1.6	M	M

Index		ISO/IEC 8073 References	ISO Status	ATN Support
ATN-TA1	R (Persistence)	12.2.1.1.5	O	O
ATN-TA2	M _{LR} (NSDU Lifetime)	12.2.1.1.1	O	O
ATN-TA3	M _{RL} (NSDU Lifetime)	12.2.1.1.1	O	O
ATN-TA4	E _{LR} (Maximum Transit Delay)	12.2.1.1.2	O	O
ATN-TA5	E _{RL} (Maximum Transit Delay)	12.2.1.1.2	O	O
ATN-TA6	A _L (Acknowledgement Time)	12.2.1.1.3	O	M
ATN-TA7	A _R (Acknowledgement Time)	12.2.1.1.3	O	M
ATN-TA8	I _R (Remote Inactivity Time)	12.2.1.1.7	O	M

Note.— According to ISO, the following applies to an implementation under test (IUT):

Index		ISO/IEC 8073 References	ISO Status	ATN Support
OT9	Does IUT support optional timer TS2 when operating in class 4?	6.22.2.3	O	O

5.5.3 Connectionless Mode Transport Protocol Operation

5.5.3.1 Connectionless Mode Transport Protocol Overview

5.5.3.1.1 ATN End Systems shall implement the ISO/IEC 8602 transport protocol in order to provide Connectionless Mode communications over the ATN Internet.

Note.— The ATN CLTS model conforms to the service model defined in ISO/IEC 8072.

5.5.3.1.2 The CLTP shall operate over the CLNS provided by the ATN Network Layer, according to the provisions in 5.5.3.5.

5.5.3.2 Connectionless Mode Transport Service Primitives

Note 1.— For the purposes of describing the notional interfaces between different OSI protocol layers, each protocol layer is assumed to provide a service to the next higher protocol layer. The assumed service provided by the ATN Transport Layer to its user is described in ISO/IEC 8072.

Note 2.— ISO/IEC 8072 limits CL user-data to a maximum of 63488 octets per TSDU.

Note 3.— There is no requirement to implement the service specified in ISO/IEC 8072 as a software interface.

5.5.3.2.1 T-UNITDATA Request

5.5.3.2.1.1 The source and destination Transport Addresses shall conform to the ATN Transport Layer Addressing provisions as specified in 5.4.

5.5.3.2.2 T-UNITDATA Indication

Note.— All of the associated parameter values are equal to the values passed to the TS provider via the T-UNITDATA Request primitive, except possibly the QoS parameter values.

5.5.3.3 ATN Connectionless Mode Transport Quality of Service Parameters

5.5.3.3.1 General

5.5.3.3.1.1 The Transport Layer shall support the use of checksums on a per TSDU (Transport Service Data Unit) basis.

Note.— The actual use of this feature will be dependant upon the application's requirements.

5.5.3.3.2 Priority

5.5.3.3.2.1 The transport entity providing the connectionless mode transport service shall allow a TS-user to specify TSDU priority in the range **[0-14]**.

Note.— The CLTP itself does not support a priority field in the TPDU.

5.5.3.3.3 Security

5.5.3.3.3.1 The transport entity providing the connectionless mode transport service shall allow a TS-user to specify the ATN Security Label in the T-UNITDATA request.

Note.— The CLTP itself does not support a security parameter field in the TPDU.

5.5.3.4 Encoding of Transport Protocol Data Units

5.5.3.4.1 The encoding of TPDU's shall conform to ISO/IEC 8602 for the CLTP.

5.5.3.5 Use of the ATN Network Service

Note.— This section specifies how the CLTP operates over the CLNS provided by the ATN Network Layer.

5.5.3.5.1 Use of the N-UNITDATA Request

5.5.3.5.1.1 General

5.5.3.5.1.1.1 The Transport Layer shall use the N-UNITDATA Request Primitive, as defined in ISO/IEC 8602, to transmit TPDU's.

5.5.3.5.1.2 NS-user-data

Note.— Transport Entities transmit TPDU's as NS-user-data of the N-UNITDATA Request primitive.

5.5.3.5.1.3 Network Service Access Point Addresses

Note.— The Transport Layer has knowledge of source and destination address parameters only as octet strings.

5.5.3.5.1.4 Network Quality of Service

5.5.3.5.1.4.1 General

5.5.3.5.1.4.1.1 The transport entity providing the connectionless mode transport service shall use the network QoS parameters as defined in the sections below.

5.5.3.5.1.4.2 Network Layer Priority

5.5.3.5.1.4.2.1 The transport entity providing the connectionless mode transport service shall use the network priority parameter to indicate the relative priority of an NSDU. The NSDU priority shall be determined from the TSDU priority, using the mapping given in Table 1-2.

5.5.3.5.1.4.3 Network Layer Security

5.5.3.5.1.4.3.1 The transport entity providing the connectionless mode transport service shall use the Security Label provided in the T-UNITDATA request as the value of the N-UNITDATA security parameter.

5.5.3.5.2 Use of the N-UNITDATA Indication

Note.— Following ISO/IEC 8602, the Transport Layer receives TPDUs from the Network Layer-provided N-UNITDATA Indication primitive.

5.5.3.5.2.1 Network Quality of Service

5.5.3.5.2.1.1 To meet the ISO/IEC 8072 service specification, the transport entity providing the connectionless mode transport service shall translate the received NSDU priority to the TSDU priority using the mapping shown in Table 1-2.

5.5.3.6 ATN Connectionless Mode Transport APRL

Note.— The requirements for the CLTP are provided in the form of an APRL. The APRL has been prepared using the PICS proforma provided with ISO/IEC 8602.

5.5.3.6.1 General

5.5.3.6.1.1 An implementation of the ISO/IEC 8602 Transport Protocol shall be used in an ATN End System if and only if its PICS is in compliance with the APRL provided with these SARPs.

5.5.3.6.2 Protocol Implementation

Item	Protocol Function Support	ISO/IEC 8602 Reference	ISO Status	ATN Support
NS	Network service selection	5.3.2.2	M	M
AM	Address mapping	5.3.2.3	M	M
	PDU Support			
UD1	Unitdata PDU supported on transmission	6.1.3	M	M
UD2AM	Unitdata PDU supported on reception	6.1.3	M	M
	Parameters of the Unitdata PDU on Transmission			
TpTc	<t> TPDU UD Checksum	6.2.4.1	O	M
TpTs	<t> TPDU UD Source Transport-Selector	6.2.4.1	M	M
TpTd	<t> TPDU UD Destination Transport-Selector	6.2.4.1	M	M
TpTu	<t> TPDU UD User Data	6.2.4.1	M	M
	Parameters of the Unitdata PDU on Reception			
TpRc	<r> TPDU UD Checksum	6.2.4.2	M	M

Item	Protocol Function Support	ISO/IEC 8602 Reference	ISO Status	ATN Support
TpRs	<r> TPDU UD Source Transport-Selector	6.2.4.2	M	M
TpRd	<r> TPDU UD Destination Transport-Selector	6.2.4.2	M	M
TpRu	<r> TPDU UD User Data	6.2.4.2	M	M
	Service Support			
CL	Connectionless Mode Network Service	6.2	M	M

5.6 INTERNETWORK SERVICE AND PROTOCOL SPECIFICATION

5.6.1 Introduction

Note 1.— The ATN Internet comprises a number of interconnected ATN routers and constituent subnetworks supporting data communication among host computers operating the ATN Internet protocols.

Note 2.— All ATN NPDUs (Network Protocol Data Units) are encapsulated within appropriate subnetwork protocol data units for transfer among ATN network entities using the connectionless ISO OSI Network Layer service provided by the ATN Internet. As the ATN Internet protocol is connectionless, any information required to process a particular NPDUs is carried within the header of that network protocol data unit for processing by ATN routers and host computers.

5.6.1.1 Scope

Note 1.— This chapter provides requirements and recommendations pertaining to the use of the ISO/IEC 8473 by ATN End System and Intermediate System Network entities. This Chapter is concerned with the use of ISO/IEC 8473 in the context of the internetworking protocol approach to the provision of CLNS as defined in ISO/IEC 8348. This Chapter contains ATN-specific protocol implementations and is concerned with the interoperability of protocol implementations. It therefore provides appropriate compliance statements and APRLs for this purpose.

Note 2.— The ATN Network Layer Connectionless-Mode Network Service supports the transfer of a connectionless network service data unit (NSDU) from a source NSAP to a destination NSAP within the ATN network. Each such NSDU transfer is the result of a single invocation of the connectionless-mode Network Service encompassed within the ATN.

5.6.1.2 Applicability of Requirements

5.6.1.2.1 All ATN Intermediate System and End System Network entities shall comply with the provisions contained in 5.6.2 and 5.6.3, in addition to all APRLs specified in 5.6.4.

5.6.2 ATN Specific Features

5.6.2.1 Purpose of ATN Specific Features

Note 1.— The ATN infrastructure, referred to as an Internet, comprises the interconnection of computers with gateways and routers via real subnetworks. This internetworking infrastructure, allows for the incorporation of differing Air/Ground and Ground/Ground subnetworks servicing differing user groups, i.e., Air Traffic Services (ATS), Aeronautical Operational Control Services (AOC), and others.

Note 2.— The CLNP (Connectionless Network Protocol) protocol used to operate this internetworking infrastructure is based on ISO/IEC 8473 with ATN-specific additions to reflect the unique communications environment of the ATN.

Note 3.— The ATN specific functions listed in this chapter reflect responses to the additional functional needs of ATN Network entities in order to support user requirements concerned with:

- a) Ensuring that information is conveyed about Traffic Type and Routing Policy requirements pertaining to user data in NPDUs;*
- b) Ensuring that a priority scheme can be applied for management of End Systems and Intermediate Systems output queues and buffers;*
- c) Ensuring that specific policies and procedures are available to handle congestion avoidance and congestion control requirements within the ATN.*

5.6.2.2 The Security Function

5.6.2.2.1 General

5.6.2.2.1.1 The SECURITY Function of ISO/IEC 8473, as defined in this specification, shall be supported by ATN End System or Intermediate System Network entities receiving or transmitting inter-domain traffic other than Traffic Type as General Communications.

5.6.2.2.1.2 ATN Network entities shall therefore provide the Globally Unique Security format for all created NPDUs.

5.6.2.2.1.3 The sole exception to this requirement is for General Communications traffic where no Security parameter information is required to be encoded in created NPDUs.

5.6.2.2.2 Encoding of the Security Parameter

5.6.2.2.2.1 The CLNP Options Security Parameter shall be used in the ATN to convey information about the Traffic Type and Routing Policy Requirements pertaining to the user data of the NPDU (other than General Communications).

Note.— The CLNP Options Security Parameter may also be used to convey a security classification.

5.6.2.2.2.2 The value component of the CLNP Options Security Parameter (in the NPDU header) shall be encoded as follows:

- a) The first octet shall always be encoded as **[1100 0000]** to indicate the Globally Unique Security Format;
- b) The remaining octets shall contain the ATN Security Label encoded as the four fields illustrated in Figure 5.6-1, and defined below.

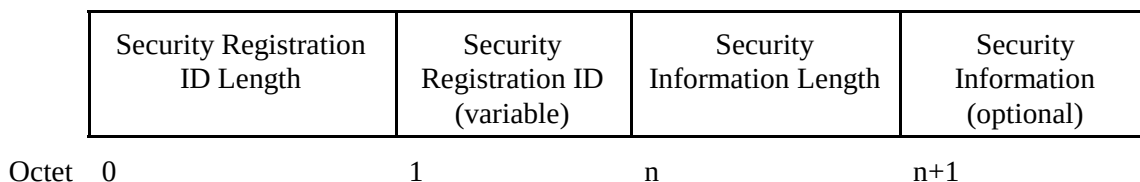


Figure 5.6-1: The ATN Security Label

5.6.2.2.3 Security Registration ID Length

5.6.2.2.3.1 This field shall be one octet long and contain the length in octets of the Security Authority's Security Registration Identifier.

Note.— The Security Registration ID identifies the authority that has specified the associated security policy.

5.6.2.2.4 Security Registration ID

5.6.2.2.4.1 This field shall contain the following hexadecimal string which identifies the ATN Security Registration ID:

[06 04 2b 1b 00 00]

Note.— The ATN Security Registration ID value defined above is the encoding using ASN.1 Basic Encoding Rules [ISO/IEC 8825-1] of the ATN Security Registration Identifier defined as {1 3 27 0 0}. This ATN Security Registration Identifier identifies the ATN Security Authority as an object in the ICAO object hierarchy. ICAO has been assigned an International Code Designator (ICD) decimal value [0027] in accordance with the dictates of ISO/IEC 6523. According to ISO/IEC 6523 and ISO/IEC 8824 this value identifies an arc of the identified organisation of ISO. ICAO object identifiers designate an ICAO defined hierarchy starting with {1 3 27}. Under this arc, {0} has been designated as ATN, and the flat address space under ATN starts with object identifiers {0,1,2,3,4, ...}. Value {0} has been assigned as the Traffic Type and Routing Policy identifier.

5.6.2.2.5 Security Information Length

5.6.2.2.5.1 This field shall be one octet in length and shall define the length in octets of the Security Information.

5.6.2.2.5.2 If there is no security information, this field shall be set to zero.

5.6.2.2.6 Security Information

5.6.2.2.6.1 General

5.6.2.2.6.1.1 When present, the Security Information field of the ATN Security Label shall be used to convey, as separate Tag Sets:

- a) The Traffic Type and Routing Policy Requirements, if any, applicable to the transfer of the user data through the ATN.
- b) The Security Classification

5.6.2.2.6.1.2 When no traffic type is identified then the General Communications traffic type shall be assumed, with a routing policy requirement of “no preference”. When no security classification is specified then “unclassified” shall be assumed.

5.6.2.2.6.2 Encoding of the Security Information Field

5.6.2.2.6.2.1 The Security Information Field shall comprise zero, one or more Security Tag Sets. A Security Tag with the same Tag Set Name shall not occur more than once in the options Security Parameter of the CLNP NPDU.

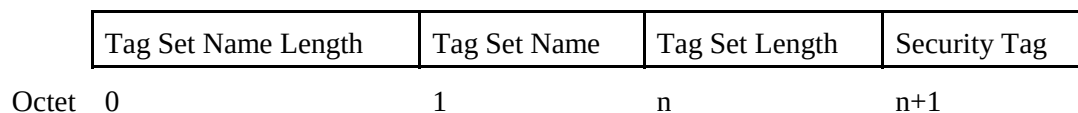


Figure 5.6-2: Security Tag Set Format

5.6.2.2.6.2.2 Each Security Tag Set shall consist of four fields, as illustrated in Figure 5.6-2, and shall be as defined in the following sections.

Note.— This format has been chosen to provide for an extensible type-length-value encoding method for security related information placed in the CLNP Header under rules specified by the ATN Security Authority.

5.6.2.2.6.3 Security Tag Set Name Length

5.6.2.2.6.3.1 The Security Tag Set Name Length shall contain the length in octets of the Tag Set Name field.

5.6.2.2.6.4 Security Tag Set Name

5.6.2.2.6.4.1 The Security Tag Set Name shall be used to uniquely identify the Tag Set.

5.6.2.2.6.5 Tag Set Length

5.6.2.2.6.5.1 The Tag Set Length Field shall contain the length in octets of the Security Tag field.

5.6.2.2.6.6 Security Tag

5.6.2.2.6.6.1 The Security Tag field shall be used to convey security related information for which the syntax and semantics are identified by the preceding Tag Set Name.

5.6.2.2.6.7 Encoding of the Tag Set for Traffic Type and Associated Routing Policies

5.6.2.2.6.7.1 The Tag Set Name shall be set to [0000 1111].

5.6.2.2.6.7.2 When present in the CLNP options Security Parameter, this Tag Set shall always be the first Tag Set to be encoded in the Security Information field of the ATN Security Label.

Note.— This Tag Set is used to identify the traffic type of the data, whether it is for ATC or Airline communications, and, for Operational Communications, any Routing Policy requirements that apply.

5.6.2.2.6.7.3 The Security Tag shall indicate the Routing Policy Requirements for the data contained in the same NPDU, according to Table 5.6-1.

Note.— See 5.2.7 for detailed information on the ATN Security Policy.

Table 5.6-1 Encoding of Traffic Type Security Tag

Traffic Type	Category	Security Tag Value	Semantics
ATN Operational Communications	Air Traffic Service Communications (ATSC)	000 00001	No Traffic Type Policy Preference.
		000 10000	Traffic preference for Class A ATSC route(s).
		000 10001	Traffic preference for Class B ATSC route(s).
		000 10010	Traffic preference for Class C ATSC route(s).
		000 10011	Traffic preference for Class D ATSC route(s).
		000 10100	Traffic preference for Class E ATSC route(s).
		000 10101	Traffic preference for Class F ATSC route(s).
		000 10110	Traffic preference for Class G ATSC route(s).
		000 10111	Traffic preference for Class H ATSC route(s).
	Aeronautical Operational Control (AOC)	001 00001	No Traffic Type Policy Preference.
		001 00010	Route Traffic only via Gatelink.
		001 00011	Route Traffic only via VHF Data Link.
		001 00100	Route Traffic only via Satellite Data Link.
		001 00101	Route Traffic only via HF Data Link.

Traffic Type	Category	Security Tag Value	Semantics
		001 00110	Route Traffic only via Mode S Data Link.
		001 00111	Route Traffic using an ordered preference of Gatelink first, then VHF Data Link.
		001 01000	Route Traffic using an ordered preference of Gatelink first, then VHF Data Link, then Satellite.
		001 01001	Route Traffic using an ordered preference of Gatelink first, then VHF Data Link, then HF Data Link, then Satellite Data Link.
ATN Administrative Communications	No category defined	001 10000	No Traffic Type Policy preference.
General Communications	No category defined	N/A	<i>Note.— General Communications traffic does not require encoding of security parameters within created NPDUs. Specification of a Security Tag Value for such General communications is therefore not applicable.</i>
ATN Systems Management Communications	No category defined	011 00000	No Traffic Type Policy preference.

5.6.2.2.6.8 Encoding of the Tag Set for Security Classification

5.6.2.2.6.8.1 The Tag Set Name shall be set to [0000 0011].

5.6.2.2.6.8.2 When present in the security parameter, this Tag Set shall always follow the Tag Set for Traffic Type and Associated Routing Policies (see 5.6.2.2.6) if present, but otherwise shall be the first Tag Set to be encoded in the field.

Note.— The purpose of this field is to permit the later extension of the ATN to handle classified data.

5.6.2.2.6.8.3 The Security Tag shall indicate the security classification of the NPDU according to the following table:

Table 5.6-2 Encoding of the Security Classification Tag

Value	Security Classification
0000 0001	unclassified
0000 0010	restricted

Value	Security Classification
0000 0011	confidential
0000 0100	secret
0000 0101	top secret
0000 0110 to 1111 1111	unassigned

5.6.2.3 Management of Network Priority

Note.— Network priority handling provisions are specified in 5.2.8.

5.6.2.4 Congestion Management

Note 1.— The congestion management provisions in the Network Layer are intended to guarantee the notification to the Transport Layer of potential risks of congestion via the CE bit conveyed in the QoS Maintenance parameter of an ISO/IEC 8473 NPDUs. 5.5.2.5 defines the measures that the Transport Layer implements upon receipt of NPDUs with the CE bit set.

Note 2.— The above requirement is applicable to all types of ISO/IEC 8473 NPDUs.

5.6.2.4.1 Setting of the congestion experienced flag

5.6.2.4.1.1 The congestion experienced flag (CE-flag) in the QoS maintenance parameter in the options part of an NPDUs header shall initially be set to zero by the originator of the NPDUs.

5.6.2.4.1.2 When a NPDUs is being forwarded by an ATN Intermediate System, the Intermediate System shall examine the depth of the output queue selected for that NPDUs.

5.6.2.4.1.3 If the depth of the selected output queue exceeds a threshold α (see Table 5.6-3), the ATN Intermediate System shall set the CE-flag in the NPDUs header.

Note.— The above assumes a single output queue per network (CLNP) priority. If mixed priority queues are implemented, which is valid provided that priority order is always maintained, then the CE-flag is set only when the number of NPDUs on the queue of the same or a higher priority exceeds alpha.

5.6.2.4.1.4 Once the CE-flag is set, it shall not be reset by any ATN Intermediate System traversed by the NPDUs further along to the path towards the destination.

5.6.2.4.2 Forwarding congestion information to the receiving NS-User

5.6.2.4.2.1 For each sequence of NPDUs that together form an NSDU, the destination network entity shall keep two counters:

- a) the first one, n-total, shall reflect the length of that sequence.

- b) the second one, n-CE, shall reflect the number of those NPDUs of this sequence, that had the CE-flag set on reception by the destination network entity.

Note.— Each NSDU is forwarded through the network as a sequence consisting of one or more NPDUs.

5.6.2.4.2.2 When the destination network entity passes an NSDU to the receiving NS-User, it shall convey the associated counters n-total and n-CE to the NS-User.

Note.— The conveyance of the congestion information associated with an NSDU to the NS-User is a local matter.

5.6.2.4.3 Required algorithm values

5.6.2.4.3.1 The value settings defined in the following table shall be implemented:

Table 5.6-3 Required Value for α

Name	Description	Required range
α	Output queue threshold	1 packet

5.6.3 ATN Specific Requirements for ISO/IEC 8473

Note.— This section defines ATN specific requirements on the ISO/IEC 8473 Protocol.

5.6.3.1 Segmentation Function

5.6.3.1.1 ATN Intermediate Systems (ISs) shall support both the segmenting and the non-segmenting subsets of ISO/IEC 8473.

5.6.3.1.2 ATN End Systems shall support the ISO/IEC 8473 segmenting subset.

Note.— Use of the non-segmenting subset of ISO/IEC 8473 for NPDUs whose packet size exceeds the maximum SNSDU size supported by an underlying subnetwork will result in the packet being discarded. Use of the non-segmenting ISO/IEC 8473 subset is most appropriate for situations where the SNSDU size of the subnetwork(s) used for NPDU transfer is well understood.

5.6.3.2 Security Function

Note.— The ATN Specific use of the ISO/IEC 8473 Security Function is specified in 5.6.2.2.

5.6.3.3 Echo Request Function

5.6.3.3.1 **Recommendation.** — ATN End System and Intermediate System Network Entities (NEs) should support the ECHO REQUEST Function as invoked by Network Layer management.

Note.— The Echo Request Function is invoked to obtain information on the reachability of specific network entities and the path characteristics between NEs through the operation of Network Layer routing functions.

5.6.3.4 Echo Response Function

5.6.3.4.1 ATN End Systems and Intermediate Systems shall support the Echo Response Function of ISO/IEC 8473.

Note.— The Echo Response function is performed by a Network Entity when it has received an Echo Request (ERQ) PDU that has reached its destination. When invoked, the Echo Response function causes an Echo Response (ERP) PDU to be created.

5.6.3.4.2 If the data part of the received ERQ PDU contains an ERP PDU header, then the options part of the ERP PDU to be sent shall be identical to (copied from) the options part of the ERP PDU header contained in the data part of the received ERQ PDU.

5.6.3.4.3 If the data part of the received ERQ PDU does not contain an ERP PDU header, then the security, priority, and QoS maintenance options of the ERP PDU shall be identical to the corresponding options in the header of the ERQ PDU, if present.

5.6.3.4.4 If the data part of the received ERQ PDU does not contain an ERP PDU header, and if the security option (respectively the priority or QoS maintenance option) is not present in the received ERQ PDU

header, then the security option (respectively the priority or QoS maintenance option) shall not be included in the ERP PDU.

5.6.3.4.5 If the data part of the received ERQ PDU does not contain an ERP PDU header, and if the partial recording of route option is present in the received ERQ PDU header, then the partial recording of route option shall be specified in the ERP PDU, with the second octet of the parameter value field set to the value 3.

5.6.3.5 Network Priority

Note.— The ATN Specific use of the ISO/IEC 8473 Priority is specified in 5.2.8.4.

5.6.4 APRLs

5.6.4.1 General

5.6.4.1.1 An implementation of the ISO/IEC 8473 Protocol shall be used in an ATN System if and only if its PICS is in compliance with the APRL provided in these SARPs.

Note.— The CLNP requirements list is a statement of which capabilities and options of the protocol at minimum are required to be implemented for the ATN environment. The requirements list may be used by the protocol implementor as a check list to conform to this standard; by the supplier and procurer to provide a detailed indication of the capabilities of an implementation; by the user to check the possibility of interworking between two different implementations; and by the protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance to the protocol.

5.6.4.2 Support of ATN-Specific Network Layer features

Index	Item	ATN SARPs Reference	ATN Support
ATN CLNP1	Encoding and use of the Security Parameter	5.6.2.2	M
ATN CLNP2	Management of Network Priority	5.6.2.3, 5.2.8.4	M
ATN CLNP4	Echo Request Function	5.6.3.3	O
ATN CLNP5	Congestion Management	5.6.2.4	M
ATN CLNP6	Echo Response Function	5.6.3.4.1	M
ATN CLNP7	Echo Response parameter setting	5.6.3.4.2, 5.6.3.4.3, 5.6.3.4.4	M

5.6.4.3 Major Capabilities

Item	Capability	ISO/IEC 8473 Reference	Status	ATN Support
ES	End System		O.1	O.1
IS	Intermediate System		O.1	O.1
FL-r	<r> Full protocol	8473-1: 6	M	M
FL-s	<s> Full protocol	8473-1: 6	M	M
NSS-r	<r> Non-segmenting subset	8473-1: 5.2	M	M
NSS-s	<s> Non segmenting subset	8473-1: 5.2	IS:M ^IS:O	IS:M ^IS:O
IAS-r	<r> Inactive subset	8473-1: 5.2	ES:O	ES:O
IAS-s	<s> Inactive subset	8473-1: 5.2	IAS-r:M ^IAS-r:X	IAS-r:M ^IAS-r:X
S802	SNDCF for ISO 8802	8473-2: 5.4	O.2	O
SCLL	SNDCF for CL Link Service	8473-4: 5.3.1	O.2	O
SCOL	SNDCF for CO Link Service	8473-4: 5.3.2	O.2	O
SX25	SNDCF for ISO 8208	8473-3: 5.4	O.2	O
ATN SNDCF	SNDCF for Mobile Subnetworks	ATN SARPs Ref: 5.7	N/A	ISMOB:M ISGRD:O ^IS:O

ISMOB: If ISO/IEC 8473 is used over Mobile Subnetworks, then ISMOB is true, else ISMOB is false.

ISGRD: If ISO/IEC 8473 is used over Ground Subnetworks, then ISGRD is true, else ISGRD is false.

O.1: The supported functions, NPDUs, associated parameters and timers required for End Systems are provided in APRLs 5.6.4.4 through 5.6.4.11. The supported functions, NPDUs, associated parameters and timers required for Intermediate Systems are provided in APRLs 5.6.4.12 through 5.6.4.18.

O.2: APRLs for the SNDCF for use with ISO/IEC 8802-2 subnetworks are provided in 5.7.7.2 and 5.7.7.3. APRLs for the SNDCF for use with ISO/IEC 8208 subnetworks are provided in 5.7.7.4 through 5.7.7.7.

5.6.4.4 End Systems - Supported Functions

Item	Function	ISO/IEC 8473-1 Reference	Status	ATN Support
ePDUC	PDU Composition	6.1	M	M
ePDUD	PDU Decomposition	6.2	M	M
eHFA	Header Format Analysis	6.3	M	M
ePDUL-s	<s> PDU Lifetime Control	6.4	M	M
ePDUL-r	<r> PDU Lifetime Control	6.4	O	M
eRout	Route PDU	6.5	M	M
eForw	Forward PDU	6.6	M	M
eSegm	Segment PDU	6.7	M	M
eReas	Reassemble PDU	6.8	M	M
eDisc	Discard PDU	6.9	M	M
eErep	Error Reporting	6.1	M	M
eEdec-s	<s> Header Error Detection	6.11	M	M
eEdec-r	<r> Header Error Detection	6.11	M	M
eSecu-s	<s> Security	6.13, ATN SARPs Ref: 5.6.2.2	O	M
eSecu-r	<r> Security	6.13, ATN SARPs Ref. 5.6.2.2	O	M
eCRR-s	<s> Complete Route Recording	6.15	O	OX
eCRR-r	<r> Complete Route Recording	6.15	O	O
ePRR-s	<s> Partial Route Recording	6.15	O	M
ePRR-r	<r> Partial Route Recording	6.15	O	M
eCSR	Complete Source Routing	6.14	O	OX
ePSR	Partial Source Routing	6.14	O	OX

Item	Function	ISO/IEC 8473-1 Reference	Status	ATN Support
ePri-s	<s> Priority	6.17, ATN SARPs Ref. 5.6.3.5	O	M
ePri-r	<r> Priority	6.17, ATN SARPs Ref. 5.6.3.5	O	M
eQOSM-s	<s> QOS Maintenance	6.16	O	M
eQOSM-r	<r> QOS Maintenance	6.16	O	M
eCong-s	<s> Congestion Notification	6.18	eQOSM-s:M	eQOSM-s:M
eCong-r	<r> Congestion Notification	6.18	O	M
ePadd-s	<s> Padding	6.12	O	OX
ePadd-r	<r> Padding	6.12	M	M
eEreq	Echo request	6.19, ATN SARPs Ref. 5.6.3.3	O	O
eErsp	Echo response	6.20, ATN SARPs Ref. 5.6.3.4	O	M
eSegS	Create segments smaller than necessary	6.8	O	O

5.6.4.5 End Systems - Supported NPDUs

Item	NPDU	ISO/IEC 8473-1 Reference	Status	ATN Support
eDT-t	DT (full protocol) transmit	7.7	M	M
eDT-r	DT (full protocol) receive	7.7	M	M
eDTNS-t	DT (non-segment) transmit	7.7	NSS-s:M	NSS-s:M
eDTNS-r	DT (non-segment) receive	7.7	M	M
eER-t	ER transmit	7.9	M	M
eER-r	ER receive	7.9	M	M
eIN-t	Inactive PDU transmit	7.8	IAS-s:M	IAS-s:M
eIN-r	Inactive PDU receive	7.8	IAS-r:M	IAS-r:M
eERQ-t	ERQ transmit	7.1	eEreq:M	eEreq:M
eERQ-r	ERQ receive	7.1	M	M
eERP-t	ERP transmit	7.11	eErsp:M	eErsp:M
eERP-r	ERP receive	7.11	M	M

5.6.4.6 End Systems - Supported DT Parameters

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
edFxFt-s	<s> Fixed Part	7.2	M	M
edFxFt-r	<r> Fixed Part	7.2	M	M
edAddr-s	<s> Address	7.3	M	M
edAddr-r	<r> Address	7.3	M	M
edSeg-s	<s> Segmentation Part	7.4	M	M
edSeg-r	<r> Segmentation Part	7.4	M	M
edPadd-s	<s> Padding	7.5.2	ePadd-s:M	-
edPadd-r	<r> Padding	7.5.2	M	M
edSecu-s	<s> Security	7.5.3	eSecu-s:M	eSecu-s:M
edSecu-r	<r> Security	7.5.3	eSecu-r:M	eSecu-r:M

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
edCRR-s	<s> Complete Route Recording	7.5.5	eCRR-s:M	-
edCRR-r	<r> Complete Route Recording	7.5.5	eCRR-r:M	eCRR-r:M
edPRR-s	<s> Partial Route Recording	7.5.5	ePRR-s:M	ePRR-s:M
edPRR-r	<r> Partial Route Recording	7.5.5	ePRR-r:M	ePRR-r:M
edCSR-s	<s> Complete Source Routing	7.5.4	eCSR:M	-
edPSR-s	<s> Partial Source Routing	7.5.4	ePSR:M	-
edQOSM-s	<s> QOS Maintenance	7.5.6	eQOSM-s or eCong-s:M	eQOSM-s:M
edQOSM-r	<r> QOS Maintenance	7.5.6	eQOSM-r or eCong-r :M	eQOSM-r or eCong-r:M
edPri-s	<s> Priority	7.5.7	ePri-s:M	ePri-s:M
edPri-r	<r> Priority	7.5.7	ePri-r:M	ePri-r:M
edData-s	<s> Data	7.6	M	M
edData-r	<r> Data	7.6	M	M
edUnSup2	Are received PDUs containing parameters selecting unsupported type 2 functions discarded and where appropriate an Error Report PDU generated ?	6.21	M	M
edUnSup3	Are parameters selecting unsupported Type 3 functions ignored ?	6.21	M	M

5.6.4.7 End Systems - Supported ER Parameters

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
eeFxPt-s	<s> Fixed Part	7.2	M	M
eeFxPt-r	<r> Fixed Part	7.2	M	M
eeAddr-s	<s> Address	7.3	M	M
eeAddr-r	<r> Address	7.3	M	M
eePadd-s	<s> Padding	7.5.2	ePadd-s:M	-
eePadd-r	<r> Padding	7.5.2	M	M
eeSecu-s	<s> Security	7.5.3	eSecu-s:M	eSecu-s:M
eeSecu-r	<r> Security	7.5.3	eSecu-r:M	eSecu-r:M
eeCRR-s	<s> Complete Route Recording	7.5.5	eCRR-s:M	-
eeCRR-r	<r> Complete Route Recording	7.5.5	eCRR-r:M	eCRR-r:M
eePRR-s	<s> Partial Route Recording	7.5.5	ePRR-s:M	ePRR-s:M
eePRR-r	<r> Partial Route Recording	7.5.5	ePRR-r:M	ePRR-r:M
eeCSR-s	<s> Complete Source Routing	7.5.4	eCSR:M	-
eePSR-s	<s> Partial Source Routing	7.5.4	ePSR:M	-
eeQOSM-s	<s> QOS Maintenance	7.5.6	eQOSM-s or eCong-s:M	eQOSM-s or eCong-s:M
eeQOSM-r	<r> QOS Maintenance	7.5.6	eQOSM-r or eCong-r:M	eQOSM-r or eCong-r:M
eePri-s	<s> Priority	7.5.7	ePri-s:M	ePri-s:M
eePri-r	<r> Priority	7.5.7	ePri-r:M	ePri-r:M
eeDisc-s	<s> Reason for discard	7.9.5	M	M
eeDisc-r	<r> Reason for discard	7.9.5	M	M
eeData-s	<s> Data	7.9.6	M	M
eeData-r	<r> Data	7.9.6	M	M

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
eeUnSup2	Are received PDUs containing parameters selecting unsupported type 2 functions discarded and where appropriate an Error Report PDU generated ?	6.21	M	M
eeUnSup3	Are parameters selecting unsupported Type 3 functions ignored ?	6.21	M	M

5.6.4.8 End Systems - Inactive DT Parameters

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
eiNLPI-s	<s> Inactive Network Layer Protocol identifier	7.8.2	IAS-s:M	IAS-s:M
eiNLPI-r	<r> Inactive Network Layer Protocol Identifier	7.8.2	IAS-r:M	IAS-r:M
eiData-s	<s> Data	7.8.3	IAS-s:M	IAS-s:M
eiData-r	<r> Data	7.8.3	IAS-r:M	IAS-r:M

5.6.4.9 End Systems - Supported ERQ Parameters

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
eqFxFt-s	<s> Fixed Part	7.2	M	M
eqFxFt-r	<r> Fixed Part	7.2	M	M
eqAddr-s	<s> Address	7.3	M	M
eqAddr-r	<r> Address	7.3	M	M
eqSeg-s	<s> Segmentation Part	7.4	M	M
eqSeg-r	<r> Segmentation Part	7.4	M	M
eqPadd-s	<s> Padding	7.5.2	ePadd-s:M	-
eqPadd-r	<r> Padding	7.5.2	M	M
eqSecu-s	<s> Security	7.5.3	eSecu-s:M	eSecu-s:M

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
eqSecu-r	<r> Security	7.5.3	eSecu-r:M	eSecu-r:M
eqCRR-s	<s> Complete Route Recording	7.5.5	eCRR-s:M	-
eqCRR-r	<r> Complete Route Recording	7.5.5	eCRR-r:M	eCRR-r:M
eqPRR-s	<s> Partial Route Recording	7.5.5	ePRR-s:M	ePRR-s:M
eqPRR-r	<r> Partial Route Recording	7.5.5	ePRR-r:M	ePRR-r:M
eqCSR-s	<s> Complete Source Routing	7.5.4	eCSR:M	-
eqPSR-s	<s> Partial Source Routing	7.5.4	ePSR:M	-
eqQOSM-s	<s> QOS Maintenance	7.5.6	eQOSM-s or eCong-s:M	eQOSM-s:M
eqQOSM-r	<r> QOS Maintenance	7.5.6	eQOSM-r or eCong-r :M	eQOSM-r or eCong-r:M
eqPri-s	<s> Priority	7.5.7	ePri-s:M	ePri-s:M
eqPri-r	<r> Priority	7.5.7	ePri-r:M	ePri-r:M
eqData-s	<s> Data	7.6	M	M
eqData-r	<r> Data	7.6	M	M
eqUnSup2	Are received PDUs containing parameters selecting unsupported type 2 functions discarded and where appropriate an Error Report PDU generated ?	6.21	M	M
eqUnSup3	Are parameters selecting unsupported Type 3 functions ignored ?	6.21	M	M

5.6.4.10

End Systems - Supported ERP Parameters

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
epFxPt-s	<s> Fixed Part	7.2	M	M

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
epFxFt-r	<r> Fixed Part	7.2	M	M
epAddr-s	<s> Address	7.3	M	M
epAddr-r	<r> Address	7.3	M	M
epSeg-s	<s> Segmentation Part	7.4	M	M
epSeg-r	<r> Segmentation Part	7.4	M	M
epPadd-s	<s> Padding	7.5.2	ePadd-s:M	-
epPadd-r	<r> Padding	7.5.2	M	M
epSecu-s	<s> Security	7.5.3, ATN SARPs Ref: 5.6.3.4.3, 5.6.3.4.4	eSecu-s:M	eSecu-s:M
epSecu-r	<r> Security	7.5.3, ATN SARPs Ref: 5.6.3.4.3, 5.6.3.4.4	eSecu-r:M	eSecu-r:M
epCRR-s	<s> Complete Route Recording	7.5.5	eCRR-s:M	-
epCRR-r	<r> Complete Route Recording	7.5.5	eCRR-r:M	eCRR-r:M
epPRR-s	<s> Partial Route Recording	7.5.5, ATN SARPs Ref: 5.6.3.4.5	ePRR-s:M	ePRR-s:M
epPRR-r	<r> Partial Route Recording	7.5.5, ATN SARPs Ref: 5.6.3.4.5	ePRR-r:M	ePRR-r:M
epCSR-s	<s> Complete Source Routing	7.5.4	eCSR:M	-
epPSR-s	<s> Partial Source Routing	7.5.4	ePSR:M	-
epQOSM-s	<s> QOS Maintenance	7.5.6, ATN SARPs Ref: 5.6.3.4.3, 5.6.3.4.4	eQOSM-s or eCong-s:M	eQOSM-s:M

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
epQOSM-r	<r> QOS Maintenance	7.5.6, ATN SARPs Ref: 5.6.3.4.3, 5.6.3.4.4	eQOSM-r or eCong-r :M	eQOSM-r or eCong-r:M
epPri-s	<s> Priority	7.5.7, ATN SARPs Ref: 5.6.3.4.3, 5.6.3.4.4	ePri-s:M	ePri-s:M
epPri-r	<r> Priority	7.5.7, ATN SARPs Ref: 5.6.3.4.3, 5.6.3.4.4	ePri-r:M	ePri-r:M
epData-s	<s> Data	7.6	M	M
epData-r	<r> Data	7.6	M	M
epUnSup2	Are received PDUs containing parameters selecting unsupported type 2 functions discarded and where appropriate an Error Report PDU generated ?	6.21	M	M
epUnSup3	Are parameters selecting unsupported Type 3 functions ignored ?	6.21	M	M

5.6.4.11 End Systems - Timers

Item	Timer	Ref	ISO Status	ISO Range	ATN Support	Values Supported
ELifReas	Is reassembly timer <= received derived PDU lifetime?	6.8	M		M	
eReasTim	Reassembly Timer	6.8	M	500ms to 127.5s	M	<= lifetime

5.6.4.12

Intermediate Systems - Supported Functions

Item	Function	ISO/IEC 8473-1 Reference	Status	ATN Support
iPDUC	PDU Composition	6.1	M	M
iPDUD	PDU Decomposition	6.2	M	M
iHFA	Header Format Analysis	6.3	M	M
iPDUL	<s> PDU Lifetime Control	6.4	M	M
iRout	Route PDU	6.5	M	M
iForw	Forward PDU	6.6	M	M
iSegm	Segment PDU	6.7	iDSNS:M	iDSNS:M
iReas	Reassemble PDU	6.8	O	O
iDisc	Discard PDU	6.9	M	M
iErep	Error Reporting	6.10	M	M
iEdec	<s> Header Error Detection	6.11	M	M
iSecu	<s>Security	6.13 ATN SARPs Ref: 5.6.2.2	O	M
iCRR	<s> Complete Route Recording	6.15	O	OX
iPRR	<s> Partial Route Recording	6.15	O	M
iCSR	Complete Source Routing	6.14	O	OX
iPSR	Partial Source Routing	6.14	O	OX
iPri	<s> Priority	6.17, ATN SARPs Ref: 5.6.3.5	O	M
iQOSM	<s> QOS Maintenance	6.16	O	M
iCong	<s> Congestion Notification	6.18, ATN SARPs Ref: 5.6.2.4	O	M
iPadd	<s> Padding	6.12	M	M
iEreq	Echo request	6.19, ATN SARPs Ref: 5.6.3.3	O	O

Item	Function	ISO/IEC 8473-1 Reference	Status	ATN Support
iErsP	Echo response	6.20, ATN SARPs Ref: 5.6.3.4	O	M
iSegS	Create segments smaller than necessary	6.8	O	O
iDSNS	Simultaneous support of subnetworks with different SN-User data sizes	6.7	O	O

5.6.4.12.1 Supported Security Parameters

Item	Function	ISO/IEC 8473-1 Reference	Status	ATN Support
iSADSSEC	Source Address Specific Security	7.5.3.1	iSecu:O.5	iSecu:O
iDADSSEC	Destination Address Specific Security	7.5.3.2	iSecu:O.5	iSecu:O
iGUNSEC	Globally Unique Security	ATN SARP Ref. 5.6.2.2	iSecu:O.5	iSecu:M

O.5: The Security parameter within a single NPDU specifies a security format code indicating Source Address Specific, Destination Address Specific or Globally Unique Security.

5.6.4.12.2 Quality of Service Maintenance Function

Item	Function	ISO/IEC 8473-1 Reference	Status	ATN Support
iQOSNAVAIL	If requested QOS not available, deliver at different QOS	6.16	iQOSM:M	iQOSM:M
iQOSNOT	Notification of failure to meet requested QOS	6.16	iQOSM:O	iQOSM:O
	Which of the following formats of QOS are implemented ?			

Item	Function	ISO/IEC 8473-1 Reference	Status	ATN Support
iSADDQOS	Source Address Specific QOS	7.5.6.1	iQOSM:O.3	iQOSM:O
iDADDQOS	Destination Address Specific QOS	7.5.6.2	iQOSM:O.3	iQOSM:O
iGUNQOS	Globally Unique QOS	7.5.6.3	iQOSM:O.3	iQOSM:M
iSvTD	Sequencing versus Transit Delay	7.5.6.3	iGUNQOS:O.4	iGUNQOS:O.4
iCongE	Congestion Experienced	7.5.6.3	iGUNQOS:O.4	iGUNQOS:M
iTDvCst	Transit Delay versus Cost	7.5.6.3	iGUNQOS:O.4	iGUNQOS:O.4
iREpVTD	Residual Error Probability versus Transit Delay	7.5.6.3	iGUNQOS:O.4	iGUNQOS:O.4
iREpVcst	Residual Error Probability versus Cost	7.5.6.3	iGUNQOS:O.4	iGUNQOS:O.4

O.3: The Quality of Service Maintenance parameter within a single NPDU specifies a QOS format code indicating Source Address Specific, Destination Address Specific or Globally Unique QOS.

O.4: If the QOS format code indicates that the Globally Unique QOS maintenance function is employed, then each bit in the associated parameter value may be set to indicate the order of intra and inter domain routing decisions based on QOS. The parameter values which apply to inter-domain routing are provided in Table 4 of ISO/IEC 10747.

5.6.4.13

Intermediate Systems - Supported NPDUs

Item	Function	ISO/IEC 8473-1 Reference	Status	ATN Support
iDT-t	DT (full protocol) transmit	7.7	M	M
iDT-r	DT (full protocol) receive	7.7	M	M
iDTNS-t	DT (non-segment) transmit	7.7	M	M
iDTNS-r	DT (non-segment) receive	7.7	M	M

Item	Function	ISO/IEC 8473-1 Reference	Status	ATN Support
iER-t	ER transmit	7.9	M	M
iER-r	ER receive	7.9	M	M
iERQ-t	ERQ transmit	7.10	iEreq:M	O
iERQ-r	ERQ receive	7.10	M	M
iERP-t	ERP transmit	7.11	iErs:M	O
iERP-r	ERP receive	7.11	M	M

5.6.4.14 Intermediate Systems - Supported DT Parameters

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
idFxFt-s	<s> Fixed Part	7.2	M	M
idFxFt-r	<r> Fixed Part	7.2	M	M
idAddr-s	<s> Addresses	7.3	M	M
idAddr-r	<r> Addresses	7.3	M	M
idSeg-s	<s> Segmentation Part	7.4	M	M
idSeg-r	<r> Segmentation Part	7.4	M	M
idPadd-s	<s> Padding	7.5.2	M	M
idPadd-r	<r> Padding	7.5.2	M	M
idSecu-s	<s> Security	7.5.3	iSecu:M	iSecu:M
idSecu-r	<r> Security	7.5.3	iSecu:M	iSecu:M
idCRR-s	<s> Complete Route Recording	7.5.5	iCRR:M	-
idCRR-r	<r> Complete Route Recording	7.5.5	iCRR:M	-
idPRR-s	<s> Partial Route Recording	7.5.5	M	M
idPRR-r	<r> Partial Route Recording	7.5.5	iPRR:M	iPRR:M
idCSR-s	<s> Complete Source Routing	7.5.4	iCSR:M	-

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
idCSR-r	<r> Complete Source Routing	7.5.4	iCSR:M	-
idPSR-s	<s> Partial Source Routing	7.5.4	M	M
idPSR-r	<r> Partial Source Routing	7.5.4	iPSR:M	-
idQOSM-s	<s> QOS Maintenance	7.5.6	M	M
idQOSM-r	<r> QOS Maintenance	7.5.6	iQOSM or iCong:M	iQOSM or iCong:M
idPri-s	<s> Priority	7.5.7	M	M
idPri-r	<r> Priority	7.5.7	iPri:M	iPri:M
idData-s	<s> Data	7.6	M	M
idData-r	<r> Data	7.6	M	M
idUnSup2	Are received PDUs containing parameters selecting unsupported type 2 functions discarded and where appropriate an Error Report PDU generated ?	6.19	M	M
idUnSup3	Are parameters selecting unsupported Type 3 functions ignored ?	6.19	M	M

5.6.4.15

Intermediate Systems - Supported ER Parameters

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
ieFxPt-s	<s> Fixed Part	7.2	M	M
ieFxPt-r	<r> Fixed Part	7.2	M	M
ieAddr-s	<s> Address	7.3	M	M
ieAddr-r	<r> Address	7.3	M	M
iePadd-s	<s> Padding	7.5.2	M	M
iePadd-r	<r> Padding	7.5.2	M	M
ieSecu-s	<s> Security	7.5.3	iSecu:M	iSecu:M

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
ieSecu-r	<r> Security	7.5.3	iSecu:M	iSecu:M
ieCRR-s	<s> Complete Route Recording	7.5.5	iCRR:M	iCRR:M
ieCRR-r	<r> Complete Route Recording	7.5.5	iCRR:M	-
iePRR-s	<s> Partial Route Recording	7.5.5	M	M
iePRR-r	<r> Partial Route Recording	7.5.5	iPRR:M	iPRR:M
ieCSR-s	<s> Complete Source Routing	7.5.4	iCSR:M	-
ieCSR-r	<r> Complete Source Routing	7.5.4	iCSR:M	-
iePSR-s	<s> Partial Source Routing	7.5.4	M	M
iePSR-r	<r> Partial Source Routing	7.5.4	iPSR:M	-
ieQOSM-s	<s> QOS Maintenance	7.5.6	M	M
ieQOSM-r	<r> QOS Maintenance	7.5.6	iQOSM or iCong:M	iQOSM or iCong:M
iePri-s	<s> Priority	7.5.7	M	M
iePri-r	<r> Priority	7.5.7	iPri:M	iPri:M
ieDisc-s	<s> Reason for Discard	7.9.5	M	M
ieDisc-r	<r> Reason for Discard	7.9.5	M	M
ieData-s	<s> Data	7.6	M	M
ieData-r	<r> Data	7.6	M	M
ieUnsup2	Are received PDUs containing parameters selecting unsupported type 2 functions discarded ?	6.21	M	M
ieUnsup3	Are parameters selecting unsupported Type 3 functions ignored ?	6.21	M	M

5.6.4.16

Intermediate Systems - Supported ERQ Parameters

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
iqFxFt-s	<s> Fixed Part	7.2	M	M
iqFxFt-r	<r> Fixed Part	7.2	M	M
iqAddr-s	<s> Addresses	7.3	M	M
iqAddr-r	<r> Addresses	7.3	M	M
iqSeg-s	<s> Segmentation Part	7.4	M	M
iqSeg-r	<r> Segmentation Part	7.4	M	M
iqPadd-s	<s> Padding	7.5.2	M	M
iqPadd-r	<r> Padding	7.5.2	M	M
iqSecu-s	<s> Security	7.5.3	iSecu:M	iSecu:M
iqSecu-r	<r> Security	7.5.3	iSecu:M	iSecu:M
iqCRR-s	<s> Complete Route Recording	7.5.5	iCRR:M	M
iqCRR-r	<r> Complete Route Recording	7.5.5	iCRR:M	-
iqPRR-s	<s> Partial Route Recording	7.5.5	M	M
iqPRR-r	<r> Partial Route Recording	7.5.5	iPRR:M	iPRR:M
iqCSR-s	<s> Complete Source Routing	7.5.4	iCSR:M	-
iqCSR-r	<r> Complete Source Routing	7.5.4	iCSR:M	-
iqPSR-s	<s> Partial Source Routing	7.5.4	M	M
iqPSR-r	<r> Partial Source Routing	7.5.4	iPSR:M	-
iqQOSM-s	<s> QOS Maintenance	7.5.6	M	M
iqQOSM-r	<r> QOS Maintenance	7.5.6	iQOSM or iCong:M	iQOSM or iCong:M
iqPri-s	<s> Priority	7.5.7	M	M
iqPri-r	<r> Priority	7.5.7	iPri:M	iPri:M

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
iqData-s	<s> Data	7.6	M	M
iqData-r	<r> Data	7.6	M	M
iqUnSup2	Are received PDUs containing parameters selecting unsupported type 2 functions discarded and where appropriate an Error Report PDU generated ?	6.19	M	M
iqUnSup3	Are parameters selecting unsupported Type 3 functions ignored ?	6.19	M	M

5.6.4.17 Intermediate Systems - Supported ERP Parameters

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
ipFxFt-s	<s> Fixed Part	7.2	M	M
ipFxFt-r	<r> Fixed Part	7.2	M	M
ipAddr-s	<s> Addresses	7.3	M	M
ipAddr-r	<r> Addresses	7.3	M	M
ipSeg-s	<s> Segmentation Part	7.4	M	M
ipSeg-r	<r> Segmentation Part	7.4	M	M
ipPadd-s	<s> Padding	7.5.2	M	M
ipPadd-r	<r> Padding	7.5.2	M	M
ipSecu-s	<s> Security	7.5.3, ATN SARP Ref: 5.6.3.4.3, 5.6.3.4.4	iSecu:M	iSecu:M
ipSecu-r	<r> Security	7.5.3, ATN SARP Ref: 5.6.3.4.3, 5.6.3.4.4	iSecu:M	iSecu:M
ipCRR-s	<s> Complete Route Recording	7.5.5	iCRR:M	M

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
ipCRR-r	<r> Complete Route Recording	7.5.5	iCRR:M	-
ipPRR-s	<s> Partial Route Recording	7.5.5, ATN SARPs Ref: 5.6.3.4.5	M	M
ipPRR-r	<r> Partial Route Recording	7.5.5, ATN SARPs Ref: 5.6.3.4.5	iPRR:M	iPRR:M
ipCSR-s	<s> Complete Source Routing	7.5.4	iCSR:M	-
ipCSR-r	<r> Complete Source Routing	7.5.4	iCSR:M	-
ipPSR-s	<s> Partial Source Routing	7.5.4	M	M
ipPSR-r	<r> Partial Source Routing	7.5.4	iPSR:M	-
ipQOSM-s	<s> QOS Maintenance	7.5.6, ATN SARPs Ref: 5.6.3.4.3, 5.6.3.4.4	M	M
ipQOSM-r	<r> QOS Maintenance	7.5.6, ATN SARPs Ref: 5.6.3.4.3, 5.6.3.4.4	iQOSM or iCong:M	iQOSM or iCong:M
ipPri-s	<s> Priority	7.5.7, ATN SARPs Ref: 5.6.3.4.3, 5.6.3.4.4	M	M
ipPri-r	<r> Priority	7.5.7, ATN SARPs Ref: 5.6.3.4.3, 5.6.3.4.4	iPri:M	iPri:M
ipData-s	<s> Data	7.6	M	M
ipData-r	<r> Data	7.6	M	M

Item	Parameter	ISO/IEC 8473-1 Reference	Status	ATN Support
ipUnsup2	Are received PDUs containing parameters selecting unsupported type 2 functions discarded and where appropriate an Error Report PDU generated ?	6.19	M	M
ipUnsup3	Are parameters selecting unsupported Type 3 functions ignored ?	6.19	M	M

5.6.4.18

Intermediate Systems - Timer and Parameter Values

Item	Timer	ISO/IEC 8473-1 Reference	Status	ATN Support
iReasTim	Reassembly Timer	6.8	iReas:M	M

5.7 SPECIFICATION OF SUBNETWORK DEPENDENT CONVERGENCE FUNCTIONS

5.7.1 Introduction

Note 1.— The purpose of a Subnetwork Dependent Convergence Function (SND CF) is to provide the connectionless SN-Service assumed by the ATN Internet Protocols over real subnetworks.

Note 2.— The Subnetwork Service (SN-Service) provided by an SND CF and as specified in this Chapter is provided to the ISO/IEC 8473 Internetwork Protocol and the ISO/IEC 9542 End System to Intermediate System Protocol entities.

Note 3.— The ATN Internetwork Layer, including CLNP and the routing protocols that support it, assumes this common connectionless service to be provided by all subnetworks providing communications between ATN systems.

Note 4.— Figure 5.7-1 illustrates the relationships between the SND CFs defined in this chapter, the SN-Service that they provide to CLNP and ES-IS, and the underlying subnetworks.

Note 5.— There is no requirement to implement this service as a software interface.

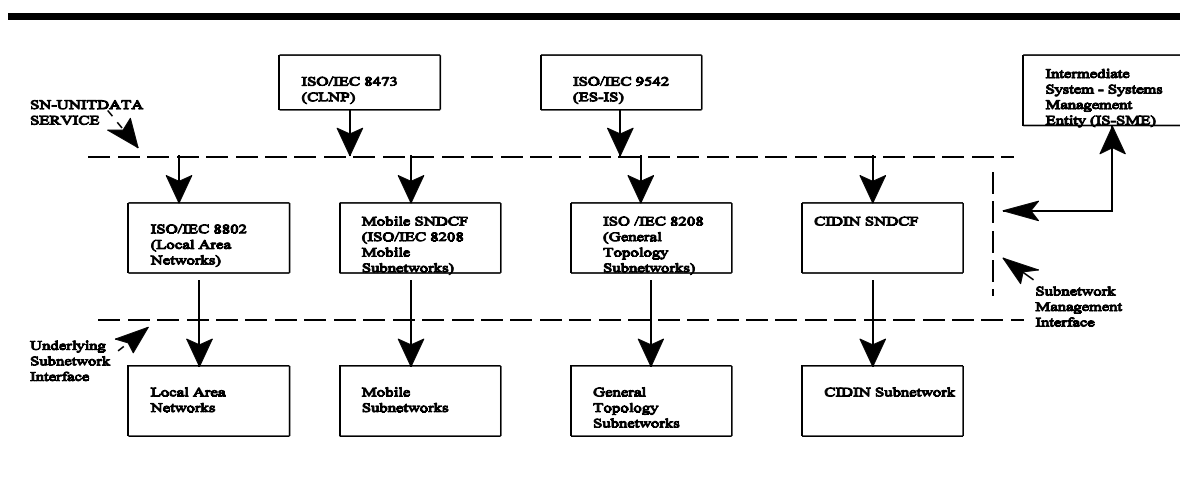


Figure 5.7-1 Relationship of SND CFs to SN-Service and underlying Subnetworks

5.7.2 Service Provided by the SND CF

Note 1.— This section specifies the assumed service provided internally by the SND CF for the purpose of conveying Network Data PDUs between Network Entities.

Note 2.— The service to support SN-Service-Users is defined by the primitives in Table 5.7-1.

Table 5.7-1 SN-Services and Associated Parameters

Parameter	SN-UNITDATA Request	SN-UNITDATA Indication
SN-Source-Address	Mandatory	Mandatory
SN-Destination-Address	Mandatory	Mandatory
SN-Priority	Optional	Optional
SN-Quality-of-Service	Optional	Optional
SNS-Userdata	Mandatory	Mandatory

5.7.2.1 Subnetwork Service Primitive Parameters

Note.— The following sections specify the Subnetwork Service primitive parameters.

5.7.2.1.1 Subnetwork Point of Attachment (SNPA) Addresses

Note.— The SN-Source-Address and SN-Destination-Address parameters specify the points of attachment to a public or private subnetwork(s). The SN-Source-Address and SN-Destination-Address addresses include information denoting a particular underlying subnetwork, as well as addressing information for systems attached directly to that subnetwork. SNPA values for a particular subnetwork are those specified and administered by the authority responsible for administration of that subnetwork.

5.7.2.1.2 SN-Priority

Note 1.— The SN-Priority parameter indicates the relative importance of the associated SNS-Userdata parameter, and may influence the order in which the SNS-Userdata are transferred via the real underlying subnetwork service.

Note 2.— SN-Priority values are in the range zero to fourteen, with higher values indicating higher priorities.

Note 3.— If no SN-Priority is indicated, the value zero is assumed to be the default.

Note 4.— Further requirements related to subnetwork priority are specified in 5.2.8.5.

5.7.2.1.3 Subnetwork Quality of Service (SNQoS)

Note 1.— The use of the SN-Quality-of-Service parameter is optional, and depends on the needs of the SN-Service-User.

Note 2.— Associated with each connectionless-mode transmission, certain measures of quality of service are requested when the SN-UNITDATA primitive action is initiated. These requested measures (or parameter values and options) are based on a priori knowledge of the service available from the subnetwork. Knowledge of the nature and type of service available is typically obtained prior to an invocation of the underlying connectionless-mode service and the information passed is a local matter.

5.7.2.1.4 Subnetwork Service Userdata

Note 1.— The SNS-Userdata contains the ISO/IEC 8473 or ISO/IEC 9542 NPDU that has to be conveyed between adjacent network entities.

Note 2.— The SNS-Userdata is an ordered multiple of octets, and is transferred transparently between the subnetwork points of attachment specified in the SNS primitive.

5.7.3 SNDCF for ISO/IEC 8802-2 Subnetworks

Note.— ISO/IEC 8802-2 subnetworks are subnetworks that provide the logical link control sublayer service defined by ISO/IEC 8802-2.

5.7.3.1 The SNDCF for use with ISO/IEC 8802-2 Subnetworks shall be implemented according to ISO/IEC 8473-2.

5.7.4 SNDCF for the Common ICAO Data Interchange Network (CIDIN)

5.7.4.1 General Considerations

Note 1.— The ISO/IEC 8473 Protocol assumes a Connectionless underlying subnetwork service. CIDIN provides a Connectionless Mode Service which is already very close to what is required by this protocol in the ATN.

Note 2.— This SNDCF maps to level 4 of CIDIN as specified in Annex 10, Volume III.

5.7.4.1.1 The SNDCF for CIDIN shall be as specified in the following sections.

5.7.4.2 SN-UNITDATA Request and Indication Primitives

5.7.4.2.1 These primitives shall correspond to the request to send a CIDIN message at a CIDIN entry centre and the reception of a CIDIN message at a CIDIN exit centre respectively.

5.7.4.2.2 CIDIN messages shall be sent with the “no acknowledgement” option.

Note.— CIDIN messages requested to be transported to exit addresses which are not reachable are discarded in the entry centre.

5.7.4.3 SN Source Address

5.7.4.3.1 This address shall correspond to a CIDIN entry address in the Entry Address item.

5.7.4.4 SN Destination Address

5.7.4.4.1 This address shall correspond to a CIDIN exit address in an Exit Address item.

5.7.4.5 SN Quality of Service

5.7.4.5.1 *A priori* values for transit delay, protection against unauthorized access, cost determinants and residual error probability shall be entered as management data in the ATN system.

5.7.4.6 SN Priority

5.7.4.6.1 The mapping between SN Priority and the CIDIN Subnetwork Priority shall be entered as management data in the ATN system.

5.7.4.7 SNS-Userdata

5.7.4.7.1 SNS-Userdata shall be conveyed as the contents of the CIDIN message which is transported transparently by CIDIN.

Note.— The coding of the CIDIN message is code and byte independent.

5.7.5 SNDCF for ISO/IEC 8208 General Topology Subnetworks

5.7.5.1 Over ISO/IEC 8208 General Topology Subnetworks, the subnetwork service described in 5.7.2 shall be provided using either the SNDCF for ISO/IEC 8208 General Topology Subnetworks as specified in ISO/IEC 8473-3, or the Mobile SNDCF specified below in 5.7.6.

Note.— Although most ATN Ground Systems are generally expected to use the ISO/IEC 8473-3 specified SNDCF over ISO/IEC 8208 General Topology Subnetworks, Ground Systems may specify the use of the Mobile SNDCF, in order to improve the bandwidth utilisation over fixed ISO/IEC 8208 subnetworks.

5.7.5.2 **Recommendation.**— Implementations using the Mobile SNDCF as specified in 5.7.6 and the LREF Compression Procedure for Ground/Ground communications, should also use the LREF optional local reference cancellation mechanism.

5.7.6 SNDCF for ISO/IEC 8208 Mobile Subnetworks

5.7.6.1 General

5.7.6.1.1 Over ISO/IEC 8208 Mobile Subnetworks, the subnetwork service described in 5.7.2 shall be provided using the SNDCF for ISO/IEC 8208 Mobile Subnetworks as specified below.

Note 1.— The SNDCF specified below is only applicable when providing the SN-UNITDATA service to ISO/IEC 8473, ISO/IEC 9542, ISO/IEC 11577 and ISO/IEC 10589 Network Layer protocols. Unpredictable behavior may result if used to support other Network Layer Entities.

Note 2.— This SNDCF supports the following Data Compression Procedures:

- *Local Reference (LREF) Compression as specified in 5.7.6.3;*
- *The ICAO Address Compression Algorithm (ACA) as specified in 5.7.6.4;*
- *Data Stream Mode Compression as specified in 5.7.6.5.*

Note 3.— The Data Stream Mode Compression uses the Deflate algorithm which was originally specified in IETF RFC 1951.

Note 4.— Optional features of LREF Compression provide for “Local Reference Cancellation” and for “maintenance of the Local Reference Directory”. The mechanism for maintaining the Local Reference Directory requires the support of the ISO/IEC 8208 Fast Select Facility.

Note 5.— A Subnetwork Connection Group is the set of virtual circuits simultaneously active between the same pair of DTEs, and which use the same subnetwork priority level, the same Data Compression Mechanisms and options, and share the same Local Reference Directory as defined in 5.7.6.3.1.

Note 6.— If a Subnetwork Connection Group already exists with the same remote DTE and the same compression mechanisms but with a different priority than the one used by the newly established virtual circuit, this circuit may not use the Local Reference Directory of this group, as packets will not travel at the same speed on two circuits which have different priorities.

Note 7.— The supported Data Compression Mechanisms and their options are negotiated when each virtual circuit used by the SNDCF is established. As a result of this negotiation, the virtual circuit is placed into a new Subnetwork Connection Group or is inserted in an existing Subnetwork Connection Group. Negotiated Data Compression mechanisms and their options are applied on a Subnetwork Connection Group basis.

5.7.6.1.2 All ATN Intermediate Systems using Mobile ISO/IEC 8208 subnetworks for communication with other Intermediate Systems shall implement the LREF compression procedure.

5.7.6.1.3 **Recommendation.**— *Implementations using this SNDCF for Air/Ground communications should only implement the LREF optional facility for local reference cancellation when the lifetime of the virtual circuits is of the same order as the flight time.*

5.7.6.2 Call Setup

5.7.6.2.1 Calling DTE Procedures

5.7.6.2.1.1 General

5.7.6.2.1.1.1 When it has been determined that a virtual circuit is to be made available, the calling SNDCF shall establish the virtual circuit using the procedures specified in ISO/IEC 8208, either

- a) dynamically, on receipt of a SN-UNITDATA request and when the SNDCF lacks a suitable virtual channel to the NPDU's destination supporting the required priority and QoS, or
- b) by the explicit intervention of Systems Management, identifying the destination SNDCF's SNPA address, priority and QoS.

5.7.6.2.1.1.2 An ISO/IEC 8208 CALL REQUEST packet shall be sent to the DTE Address specified as the SN-Destination-Address, with the following optional user facilities and CCITT-specified DTE facilities.

Note 1.— Normally, this is achieved by encoding the destination DTE Address as the called address of the ISO/IEC 8208 Call Request packet. This is appropriate when the ATN Router is directly connected to the air/ground subnetwork, or when it is connected to the air/ground subnetwork via another subnetwork and an interworking facility (ISO TR 10029). However, when the ATN Router is connected to the air/ground subnetwork via another subnetwork and an interworking facility is not available, one possible alternative approach is to address the ISO/IEC 8208 Call Request packet to the access point of the air/ground subnetwork (e.g. a GDLP) and to convey the destination DTE Address in the Called Address Extension facility of the ISO/IEC 8208 Call Request packet whereas the DTE addresses configured for the local access point of the air/ground subnetwork is encoded in the called address field of the ISO/IEC 8208 Call Request packet. It is then the responsibility of the air/ground subnetwork access facility to reformat the received ISO/IEC 8208 Call Request packet into a Call Request packet appropriate for transmission to the destination DTE address over the air/ground subnetwork.

Note 2.— Other optional user facilities and CCITT-specified DTE facilities may be required by subnetworks. The use of these facilities is a local matter.

5.7.6.2.1.1.3 The Call Request user data shall be formatted as specified in 5.7.6.2.1.5.

5.7.6.2.1.2 The Priority Facility

5.7.6.2.1.2.1 The mapping of ATN network layer priorities to ATN mobile subnetwork priorities shall be as defined in 1.3.8 for those mobile subnetworks subject to ICAO standards.

5.7.6.2.1.2.2 For mobile subnetworks not subject to ICAO standards, the Priority Facility shall be used if the subnetwork provider supports prioritisation of Virtual Circuits and specifies the mapping of Network Service to Subnetwork Service priorities.

5.7.6.2.1.2.3 The priority value passed in the SN-UNITDATA request or indicated by the System Manager shall be mapped to priority of data on a connection, as specified by the Subnetwork Provider.

5.7.6.2.1.2.4 If the priority to gain a connection and/or priority to keep a connection is conveyed within the ISO/IEC 8208 Facility Parameter Field, these priorities shall be consistent with the priority of data on a connection, and set according to the Subnetwork Provider's guidelines.

Note 3.— The SNDCF is assumed to know, a priori, if a given subnetwork supports prioritisation of virtual circuits, the number of discrete priority levels supported and the relationship between the subnetwork priority and SNSDU priority.

Note 4.— The mapping between SNSDU priority and subnetwork priority is specified separately for each subnetwork type.

5.7.6.2.1.3 The Non-Standard default packet size Facility

5.7.6.2.1.3.1 Non-standard default packet size Facility shall be used and the value requested set to the maximum supported by the subnetwork.

5.7.6.2.1.4 The Fast Select Facility

5.7.6.2.1.4.1 The Fast Select Facility shall be used if supported by all Subnetwork Provider(s) in the DTE-DTE virtual path.

Note.— Airborne routers are assumed to have a priori knowledge of Fast Select support (or lack thereof) along the DTE-DTE virtual path based on each individual destination air/ground router's DTE address.

5.7.6.2.1.4.2 No restriction on response shall be indicated.

Note 1.— This permits the responding DTE accept the call and to return up to 128 octets of user data.

Note 2.— If Fast Select is not supported, the Compression Procedures can only be negotiated by successive attempts to establish the virtual circuit requesting different combinations of Compression Procedures.

5.7.6.2.1.5 Call Request User Data

Note.— Call Request User Data is used to indicate which Compression Procedures are offered by the calling DTE. When the Fast Select Facility is used, Call Accept User Data is then used to indicate which Compression Procedures are accepted by the Called DTE.

5.7.6.2.1.5.1 The Call Request User Data format shall be as illustrated in Figure 5.7-2.

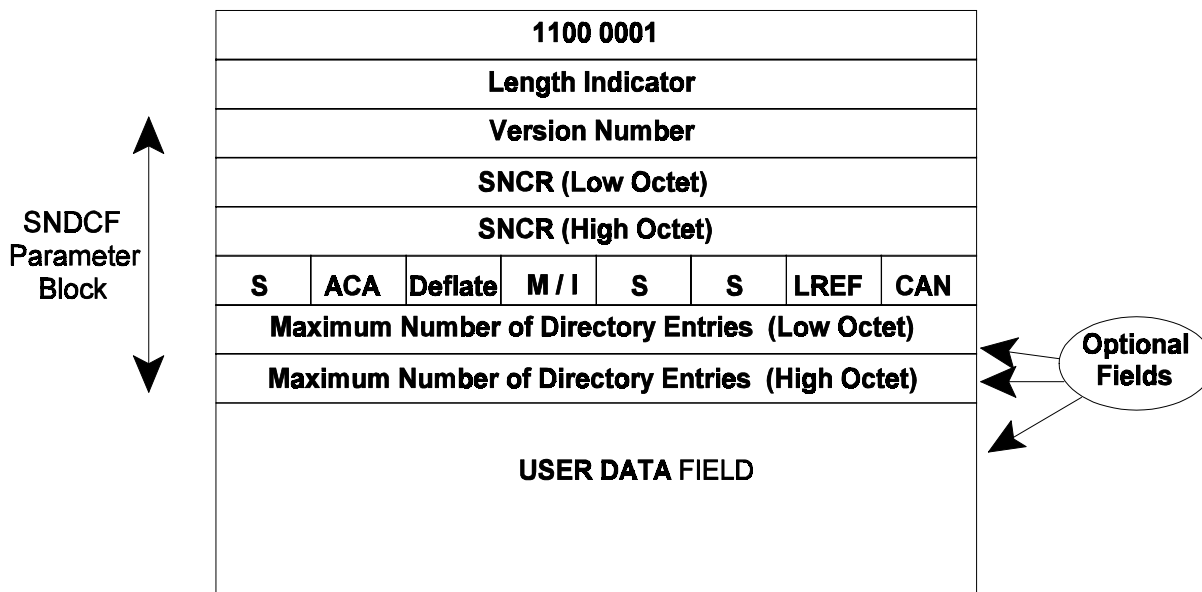


Figure 5.7-2 Format for Call Request User Data

5.7.6.2.1.5.2 The first octet of the call user data (the Subsequent Protocol Identifier (SPI)) shall be set to binary [1100 0001] to indicate that the virtual circuit is to be used to provide the underlying service by this SND CF.

Note.— ISO TR 9577 provides the international register for SPI values. The value binary [1100 0001] has not been assigned by the ISO Technical Report and it is unlikely that it will be.

5.7.6.2.1.5.3 The value of the second octet (length indicator) shall be an unsigned binary number giving the number of octets in the SND CF parameter block (from version number field up to and including (if present) the maximum number of directory entries field).

5.7.6.2.1.5.4 The third octet is the SND CF version indicator and shall be set to [0000 0001] to indicate this version of the SND CF protocol.

5.7.6.2.1.5.5 The fourth and fifth octets shall provide the Subnetwork Connection Reference (SNCR).

5.7.6.2.1.5.6 The value encoded in this field shall be the lowest available SNCR value in the range from 0 up to one less than the number of virtual circuits needed at this call priority.

Note.— The use of the SNCR is specified in ISO/IEC 8473 for use in call collision resolution over ISO/IEC 8208 subnetworks.

5.7.6.2.1.5.7 The sixth octet shall indicate the compression techniques offered by the calling DTE, according to Table 5.7-2.

5.7.6.2.1.5.8 LREF Compression shall always be offered.

Note 1.— This specification mandates the use of the LREF Compression algorithm. This may not be true in future versions of this specification. Hence procedures are specified to negotiate the use of the LREF Compression on a per virtual circuit basis.

Note 2.— The decision as regards which options to offer out of those supported is otherwise a local matter.

Note 3.— Multiple compression procedures may be offered.

Table 5.7-2 Compression Options Offered Parameter

bit number	option
bit 8	Spare (S)
bit 7	ICAO Address Compression Algorithm (ACA)
bit 6	Deflate
bit 5	Maintenance/Initiation of Local Reference Directory option (M/I)
bit 4	Spare (S)
bit 3	Spare (S)
bit 2	Local Reference (LREF) option
bit 1	Local Reference Cancellation Option (CAN) supported

5.7.6.2.1.5.9 Bit 1 of octet 6 shall only be set if bit 2 is also set.

Note.— At most, one of the the ACA or Deflate compression algorithms can be used. However, both can be offered in the Call Request Packet when Fast Select is in use, but only one can be accepted in the Call Accept Packet.

5.7.6.2.1.5.10 Both ACA and Deflate shall not be simultaneously offered if the Fast Select Facility is not in use.

5.7.6.2.1.5.11 When the LREF Compression algorithm is offered, i.e if bit 2 in octet six is set, then the seventh and eight octets (Maximum Directory Entries) shall indicate the maximum number of directory entries supported for the local reference (minimum size 128), as an unsigned even number.

5.7.6.2.1.5.12 The M/I bit shall be set to one by the calling SND CF in the Call Request Packet when the calling SND CF has identified a Subnetwork Connection Group with the called DTE, with the requested

subnetwork priority and Data Compression mechanisms and options, to request that the newly established circuit shares the Local Reference Directory associated with this group.

5.7.6.2.1.5.13 The request for Local Reference directory maintenance shall only be used when the Call Request uses the Fast Select Facility and when bit 2 of the Compression Options Parameter (Local Reference compression) is set to one.

5.7.6.2.1.5.14 When the request for Local Reference directory maintenance is used, then the Subnetwork Connection Reference (SNCR) of the Call Request packet shall be set to the lowest available SNCR value in the range from 0 up to one less than the number of virtual circuits needed at this call priority.

5.7.6.2.1.5.15 When the LREF compression algorithm is not offered, the seventh octet shall be the first octet of the User Data field.

5.7.6.2.1.5.16 When the LREF compression algorithm is offered, the ninth octet shall be the first octet of the User Data field.

Note.— When the fast select facility is available, the User Data field may be used to convey the ISO/IEC 9542 ISH PDU as part of the routing initiation sequence.

5.7.6.2.1.6 Receipt of “Call Confirm Packet”

5.7.6.2.1.6.1 Fast Select Facility In Use

5.7.6.2.1.6.1.1 When an ISO/IEC 8208 Call Confirm Packet is received from the Called DTE and the Fast Select Facility is in use, then the Calling DTE shall inspect the Call Confirm User Data (see 5.7.6.2.2.4) in order to determine which of the offered Compression Procedures have been accepted.

5.7.6.2.1.6.1.2 If the called SNDCEF has accepted the call indicating that an offered Compression Procedure is not supported, then the Calling SNDCEF shall maintain the virtual circuit and shall not apply this compression procedure.

5.7.6.2.1.6.1.3 If the M/I bit is set to zero in the Call Confirm User Data, then a new Subnetwork Connection Group shall be created and the newly established virtual circuit becomes the first member of that Group.

5.7.6.2.1.6.1.4 If the M/I bit is set to one in the Call Confirm User Data and the M/I bit in the preceding Call Request had also been set to one, then the newly established virtual circuit shall be inserted into the Subnetwork Connection Group identified when the Call Request was issued.

5.7.6.2.1.6.1.5 If the M/I bit is set to one in the Call Confirm User Data, and M/I bit had been set to zero in the preceding Call Request, then this is an error condition, and the call shall be cleared with an ISO/IEC 8208 Cause Code of zero, and a diagnostic code of 242 (Disconnection - incompatible information in user data).

5.7.6.2.1.6.1.6 If the length of the User Data of the received Call Confirm Packet is greater than one, then the remaining part of the received Call Confirm User Data contains an NPDU, and the calling SNDCEF shall pass this NPDU in an SN-UNITDATA indication to the appropriate SN-Service User.

5.7.6.2.1.6.1.7 The first octet of this NPDU (i.e. the SPI) shall be used by the calling SNDCEF in order to identify the network layer protocol, and hence which SN-Service User is responsible for handling this NPDU.

5.7.6.2.1.6.1.8 If no such SN-Service user exists, then the NPDU shall be discarded.

5.7.6.2.1.6.2 Fast Select Facility not in Use

5.7.6.2.1.6.2.1 When an ISO/IEC 8208 Call Confirm Packet is received from the Called DTE and the Fast Select Facility is not in use, then the Calling DTE shall assume that all of the offered Compression Procedures have been accepted.

5.7.6.2.1.7 Call Rejection by the DCE or Called DTE

5.7.6.2.1.7.1 General

5.7.6.2.1.7.1.1 **Recommendation.**— *When a DTE originated ISO/IEC 8208 Call Clearing Packet is received with a diagnostic value indicating that the proposed LREF directory is too big (see Table 5.7-3), then the call should be re-attempted with the minimum directory size (see 5.7.6.3.1.3).*

Note.— *This is to ensure that the call is not rejected again due to the requested directory size being too big.*

5.7.6.2.1.7.1.2 If the diagnostic indicates Call Collision resolution then no further attempt shall be made to re-establish the call.

5.7.6.2.1.7.1.3 **Recommendation.**— *In all other cases, the problem should be reported to a System Manager.*

Note.— *Any further attempts to establish the virtual circuit are a local matter.*

5.7.6.2.1.7.2 Fast Select Facility Requested

5.7.6.2.1.7.2.1 When a DCE or DTE originated ISO/IEC 8208 Call Clearing Packet is received with a diagnostic indicating Fast Select not Subscribed or Fast Select Acceptance Not Subscribed, then the call shall be re-attempted but without requesting the Fast Select Facility.

Note.— *Some Network Service Providers may indicate the non availability of the Fast Select Facility via other diagnostic codes.*

5.7.6.2.1.7.3 Fast Select Facility not in Use

Note.— *In this case, when rejection by the called DTE indicates that the reject reason is due to an offered compression procedure not being supported, then the call is re-attempted without offering the rejected procedure. This is the only negotiation procedure possible when Fast Select is not available.*

5.7.6.2.1.7.3.1 When a DTE originated ISO/IEC 8208 Call Clearing Packet is received with a diagnostic indicating *LREF Compression not Supported* (see Table 5.7-3), the call shall be re-attempted without offering LREF Compression.

5.7.6.2.1.7.3.2 When a DTE originated ISO/IEC 8208 Call Clearing Packet is received with a diagnostic indicating *Local Reference Cancellation not Supported* (see Table 5.7-3), the call shall be re-attempted without offering Local Reference Cancellation.

5.7.6.2.1.7.3.3 When a DTE originated ISO/IEC 8208 Call Clearing Packet is received with a diagnostic indicating *ACA compression not Supported* (see Table 5.7-3), the call shall be re-attempted without offering the ACA.

5.7.6.2.1.7.3.4 When a DTE originated ISO/IEC 8208 Call Clearing Packet is received with a diagnostic indicating *Deflate compression not Supported* (see Table 5.7-3), the call shall be re-attempted without offering Deflate compression.

5.7.6.2.2 Called DTE Procedures

5.7.6.2.2.1 Incoming Call Processing

5.7.6.2.2.1.1 When an ISO/IEC 8208 Incoming Call Packet is received, the called SNDCF first shall check for a call collision.

5.7.6.2.2.1.2 If the SNDCF has an outstanding Call Request to the same DTE Address, specified as the calling DTE in this Incoming Call Packet, and the call priority and SNCR are identical, then a call collision has occurred, and the call collision resolution procedures specified in ISO/IEC 8473-3 shall be invoked to resolve the call collision.

5.7.6.2.2.1.3 The called SNDCF shall then determine whether to accept the call.

5.7.6.2.2.1.4 The call shall be rejected if any of the following conditions are true:

- a) The proposed ISO/IEC 8208 facility is not available;
- b) The proposed priority is not supported;
- c) The Fast Select Facility was not selected in the Incoming Call Packet and an offered compression algorithm is not supported;
- d) The format of the call user data is invalid;
- e) The version number is not supported;
- f) The Local Reference compression is offered and the called SNDCF does not support the proposed directory size;
- g) Local Policy does not permit communication with the calling DTE.

5.7.6.2.2.1.5 The call shall then be rejected using a Call Clearing Packet, with the appropriate diagnostic code, as listed in Table 5.7-3.

5.7.6.2.2.1.6 If the call is to be accepted then the Called SNDCF shall perform the ISO/IEC 8208 procedures associated with accepting a call.

5.7.6.2.2.2 Call Acceptance with the Fast Select Facility in Use

5.7.6.2.2.2.1 The combination of compression techniques acceptable to the SND CF, out of those offered by the Calling SND CF, shall be indicated by setting the appropriate bits in the first octet of the ISO/IEC 8208 Call Accept User Data as shown in Figure 5.7-3.

5.7.6.2.2.2.2 The Called SND CF shall not accept both the ACA and Deflate compression algorithms simultaneously.

5.7.6.2.2.2.3 If the M/I bit is set to one in the Call Request User Data and,

- a) there is one and only one existing Subnetwork Connection Group with the calling DTE with the same Data Compression Procedures and options as indicated in the Call Request User Data, and the requested priority, and
- b) it is acceptable to share the Local Reference Directory associated with this Subnetwork Connection Group with this virtual circuit,

then the virtual circuit shall be inserted in this Subnetwork Connection Group and the M/I bit set to one in the Call Accept User Data.

5.7.6.2.2.2.4 Otherwise, a new Subnetwork Connection Group shall be created, with this virtual circuit as the first member of the group and the M/I bit set to zero in the Call Accept User Data.

Note.— By setting the M/I bit to zero, the responding SND CF can refuse to maintain the Local Reference directory from the old virtual circuit to the new virtual circuit. This will result in an additional Subnetwork Connection Group and , as long as one or more exists, in all further Local Reference directory maintenance requests to be rejected.

5.7.6.2.2.2.5 If there is additional User Data beyond the SND CF Parameter Block (see Figure 5-7.2) in the received Incoming Call Packet and the first octet of this additional user data is a recognized NPDU SPI, then the remaining part of the received Incoming Call User Data contains an NPDU, and the called SND CF shall pass this NPDU in an SN-UNITDATA indication to the appropriate SN-Service User.

5.7.6.2.2.2.6 The first octet of this NPDU (i.e. the SPI) shall be used by the called SND CF in order to identify the network layer protocol, and hence which SN-Service User is responsible for handling this NPDU.

5.7.6.2.2.2.7 If no such SN-Service User exists, then the NPDU shall be discarded.

5.7.6.2.2.3 Call Acceptance without the Fast Select Facility in Use

5.7.6.2.2.3.1 If Fast Select is not in use then a call shall only be accepted if all offered compression procedures and facilities are acceptable, and the proposed LREF directory size can be supported.

Note.— Call rejection is specified above in 5.7.6.2.2.1.4.

5.7.6.2.2.4 Call Accept User Data

Note.— User Data can only be present in the Call Accept packet if the Fast Select Facility is available and has been selected in the Call Request.

5.7.6.2.2.4.1 When Fast Select is available and has been selected in the Call Request, then a Call Accept



Figure 5.7-3 Format for Call Accept User Data

User Data shall be present in the Call Accept packet.

5.7.6.2.2.4.2 The Call Accept User Data format shall be as illustrated in Figure 5.7-3.

5.7.6.2.2.4.3 The first octet of the Call Accept User Data shall identify the compression procedure(s) accepted by the called DTE.

Note.— The bit fields have the same semantics as the ones used for the sixth octet of the Call Request User Data.

5.7.6.2.2.4.4 In case that the Call Accept Packet will be used to convey an NPDU, the second octet of the Call Accept User Data shall be the first octet of this NPDU.

Note 1.— An ISO/IEC 9542 ISH PDU may be conveyed as part of the routing initiation procedure.

Note 2.— Since the negotiated compression procedures apply to the data transfer phase (see 5.7.6.2.3.1), the optional NPDU in the Call Accept User Data, if present, is sent uncompressed.

5.7.6.2.3 Data Transfer Phase

5.7.6.2.3.1 During the data transfer phase of a virtual circuit established by this SNDCF, the Compression Procedures accepted by the called DTE shall be applied to each NPDU transferred over the virtual circuit.

Note.— NPDUs are queued for transfer as a result of an SN-UNITDATA.request. Received NPDUs are passed to the SN-Service user by an SN-UNITDATA.indication.

5.7.6.2.3.2 The order in which concurrently applied Compression Procedures and ISO/IEC 8208 segmentation are applied shall be as follows:

- a) If the LREF compression algorithm is used, it shall be applied to the ISO/IEC 8473 PDU first;

- b) If either of the ACA or Deflate compression algorithms is used, it shall be applied after LREF compression and before M-bit segmentation;
- c) Finally, if the ISO/IEC 8208 M-bit sequencing procedures are required due to the size of the PDU, then these shall be applied.

5.7.6.2.3.3 This sequence shall be inverted on the receiving end as follows:

- a) If M-bit -segmentation has been applied, then reassembly of the NPDU from the received ISO/IEC 8208 Data Packets shall be done first;
- b) If either of the ACA or Deflate compression algorithms is used the corresponding decompression algorithm shall be applied after M-bit segmentation and before LREF compression;
- c) Finally if the LREF compression is used, the LREF decompression algorithm shall then be applied.

5.7.6.2.4 Call Clearing

5.7.6.2.4.1 The SNDCF shall clear a virtual circuit when:

- a) System Management requests call clearing, or
- b) On the expiration of a timeout period following the transmission or receipt of SN-UNITDATA, or
- c) If the resources are required by another virtual circuit with a higher priority.

5.7.6.2.4.2 Items b) or c) above shall only apply to those virtual circuits that have been established following an SN-UNITDATA.request.

5.7.6.2.4.3 When it has been determined that a virtual circuit is to be cleared, the SNDCF shall invoke the ISO/IEC 8208 functions associated with call clearing.

5.7.6.2.4.4 All packets subsequently received other than a Clear Confirm or a Clear Indication shall be ignored.

5.7.6.2.4.5 The same actions shall apply to the receipt of a Clear Indication.

5.7.6.2.4.6 The Clearing Cause octet in the ISO/IEC-8208 Cause/Diagnostic field shall be set to [1000 0000].

5.7.6.2.4.7 The reason for clearing the call shall be placed in the Diagnostic field using the appropriate diagnostic values according to Table 5.7-3.

Note.— If a virtual connection is cleared due to a network problem, the SNDCF may attempt to re-establish the connection before the associated forwarding information is removed from Network Layer

routing tables. The selective re-establishment of X.25 connections may be based on the originating Clearing Cause and Diagnostic Codes.

5.7.6.3 Local Reference Compression Procedures

5.7.6.3.1 Local Directory Initialization

5.7.6.3.1.1 Both calling and called SNDCFs shall create a local directory to be associated with each newly established Subnetwork Connection Group.

5.7.6.3.1.2 This directory shall consist of entries numbered from zero to a maximum of 32767, each entry consisting of:

- a) A pair of NSAP Addresses, known as the inward and outward NSAP Addresses respectively;
- b) The ISO/IEC 8473 protocol version number;
- c) The value of the security options parameter which may be empty.

5.7.6.3.1.3 The directory shall be initially empty. The Mobile SNDCF shall support a minimum directory size of 128 entries.

Table 5.7-3 Diagnostic values for ATN call clearing

	Hexadecimal value	Decimal value	Clearing Cause
1	1111 1001	249	Connection Rejection - unrecognized protocol identifier in user data
2	1000 0000	128	Version number not supported
3	1000 0001	129	Length field invalid
4	1000 0010	130	Call Collision Resolution
5	1000 0011	131	Proposed Directory Size too large
6	1000 0100	132	Local Reference Cancellation Not Supported
7	1000 0101	133	Received DTE refused, received NET refused or invalid NET selector
8	1000 0110	134	Invalid SNCR field
9	1000 0111	135	ACA compression not supported
10	1000 1000	136	LREF compression not supported
11	1000 1111	143	Deflate compression not supported
12	1111 0000	240	System lack of resources
13	0000 0000	0	Cleared by System Management

	Hexadecimal value	Decimal value	Clearing Cause
14	1001 0000	144	Idle Timer expiration
15	1001 0001	145	Need to re-use the circuit
16	1001 0010	146	By local means (to be used for system local error)
17	1001 0011	147	Invalid SEL field value in received NET

5.7.6.3.2 Action following an SN-UNITDATA Request

5.7.6.3.2.1 General

5.7.6.3.2.1.1 On receipt of a SN-UNITDATA request the SNDCF shall identify an appropriate virtual circuit to the subnetwork user associated with the SN-Destination-Address, and which satisfies the PDU Priority and Security requirements, and queue the accompanying PDU (i.e. the user data associated with the SN-UNITDATA request) for transfer over that virtual circuit.

5.7.6.3.2.1.2 If there is no virtual circuit which satisfies the PDU Priority and Security requirement, then the SNDCF shall try to establish a virtual circuit with the requested PDU Security and priority.

5.7.6.3.2.1.3 If a suitable virtual circuit can be established, then the PDU shall be queued for transfer over the newly established virtual circuit. If no such virtual circuit can be established, then if an existing virtual circuit associated with the SN-Destination-Address provides an adequate level of security and priority, the PDU shall be queued for transfer over the existing virtual circuit.

5.7.6.3.2.1.4 Otherwise, the PDU shall be discarded.

Note 1.— The opening of an additional virtual circuit for this purpose may be inappropriate in certain cases. For example, opening an additional virtual circuit via a single frequency VDL subnetwork or via the Mode S subnetwork will not necessarily result in increased capacity.

Note 2.— The maintenance of the minimum QoS level includes ensuring that the number of local references that are required to support the number of data streams multiplexed over a given virtual circuit does not exceed the number available.

5.7.6.3.2.1.5 If no virtual circuit exists to the SN-Destination-Address, and the circuit is not classified as dynamically assigned by the ISO/IEC 10589 (IS-IS) routing protocol or under a static routing regime, then the SN-UNITDATA shall be discarded, with an error report sent to a System Manager.

Note.— Virtual Circuits between Intermediate Systems and between Intermediate Systems and End Systems are initially established by procedures associated with the specific routing procedures employed. If no such virtual circuit has been established, or may be established under the routing procedures, then no route exists and hence it is an error if an attempt is made to send a PDU over such a route.

5.7.6.3.2.2 Identification of Network Layer Protocol

5.7.6.3.2.2.1 Prior to transmission of an SN-UNITDATA SN-Userdata parameter over a virtual circuit, the SNDCEF shall inspect the initial octet of the SN-Userdata parameter (Initial Protocol Identifier (IPI)) to identify the Network Layer protocol contained within the SN-UNITDATA request.

5.7.6.3.2.2.2 If the IPI contains binary [1000 0001] indicating ISO/IEC 8473, then the procedures in 5.7.6.3.2.3 shall be performed.

5.7.6.3.2.2.3 If the IPI contains binary [1000 0010] indicating ISO/IEC 9542 (ES-IS), binary [1000 0011] indicating ISO/IEC 10589 (IS-IS), or binary [0100 0101] indicating ISO/IEC 11577 (NLSP), then the packet shall be sent unchanged over the virtual circuit, using the M-bit segmentation mechanism, if the packet is larger than the maximum length of user data permitted for the virtual circuit.

5.7.6.3.2.2.4 If the IPI contains any other value, the SN-UNITDATA request shall be discarded, and an error sent to a System Manager.

Note.— The IPI designating the ISO/IEC 11577 has been included in the set of allowed IPIs in order to preserve the possibility for use of this protocol in the future. However, at the time of publication of this specification, no ATN Security Protocol Architecture has been defined. Thus, this inclusion of the NLSP IPI in the allowed IPI set does not indicate that NLSP will be incorporated into the future ATN security architecture.

5.7.6.3.2.3 Identification of Option Parameter and Local Directory Look-up

5.7.6.3.2.3.1 The options part of the ISO/IEC 8473 NPDU header contained in the SN-Userdata shall then be inspected. If one of the following is true:

- a) Source Routing option is present,
- b) Recording of Route option is present,
- c) QoS Maintenance option is anything other than the globally unique format,
- d) padding option is present,
- e) priority option is present with a value greater than 14,
- f) an unknown parameter is present,

then the SN-Userdata shall be sent unchanged over the virtual circuit using M-bit segmentation procedures as appropriate.

5.7.6.3.2.3.2 Otherwise, the local directory associated with the virtual circuit shall then be interrogated to determine if an entry exists such that:

- a) the inward NSAP Address is equal to the PDU's source NSAP Address;
- b) the outward NSAP Address is equal to the PDU's destination NSAP Address;
- c) a security parameter is present with the same value as that contained in the PDU header, if present, and otherwise absent;

- d) the same ISO/IEC 8473 version number as is present in the PDU header.

5.7.6.3.2.3.3 If an entry is found, then the NPDU shall be sent in the compressed form constructed according to 5.7.6.3.3, using the local directory entry number as the local reference.

5.7.6.3.2.3.4 If no entry is found, then a new directory entry shall be created and the SN-Userdata shall be modified as specified in 5.7.6.3.2.4.

5.7.6.3.2.4 Establishing a New Local Reference

5.7.6.3.2.4.1 A new directory entry shall be created containing the NPDU source NSAP Address as the inward NSAP Address, and the NPDU destination NSAP Address as the outward NSAP Address.

5.7.6.3.2.4.2 The value of the protocol version number, and the security parameter, if present, shall also be placed in this entry.

5.7.6.3.2.4.3 The entry number shall have the lowest possible entry number that has not previously been used for the local directory associated with this virtual circuit, and shall be in the range [0..63] or [128..16447] if the SNDCF is the initiator of the first virtual circuit in a Subnetwork Connection Group, or [64..127] or [16448..32767], if the SNDCF is the responder for such a virtual circuit.

5.7.6.3.2.4.4 When a directory size greater than 128 but less than 32767 has been negotiated, then the highest local reference that the initiator may allocate shall be

$$127 + (n - 128) / 2$$

and the highest local reference that the responder may allocate shall be

$$16447 + (n - 128) / 2$$

where 'n' is the agreed maximum directory size.

5.7.6.3.2.4.5 If a directory full condition occurs then, as a local matter, either the PDU shall be sent unmodified over the virtual circuit or the virtual circuit shall be reset.

Note.— A user generated Network Reset results in the total clearing of the directory which then permits the assignment of an unused local reference.

5.7.6.3.2.4.6 **Recommendation.**— *When this SNDCF is used for Air/Ground communication or when the local reference cancellation option is available for use, then the PDU should be sent unmodified over the virtual circuit.*

5.7.6.3.2.4.7 The PDU, which may be either a DT PDU or an ER PDU, shall have an additional options field added to the PDU header.

5.7.6.3.2.4.8 This option parameter shall have local significance only (i.e. is only of interest to the sending and receiving SNDCEs), and is called the Local Reference.

5.7.6.3.2.4.9 This Local Reference option parameter shall be included as the first parameter in the Option Part of the DT or ER PDU header.

5.7.6.3.2.4.10 This option shall be specified as follows:

Parameter Code:	[0000 0101]
Parameter Length:	variable
Parameter Value:	the entry number of the local directory entry created above and expressed as an unsigned integer.

Note.— The entry number is therefore assigned as a so called Local Reference.

5.7.6.3.2.4.11 The Checksum, Length Indicator, and Segment Length fields of the PDU header shall be modified to reflect the insertion of the new options field, and any changes to the length of the source and destination address.

5.7.6.3.2.4.12 The Total Length, if present, shall be left unmodified.

5.7.6.3.2.5 Reference Cancellation Option

5.7.6.3.2.5.1 When the optional Local Reference Cancellation facility is implemented, and both SNDCFs using a virtual circuit have indicated that they support this facility, then the SNDCF shall monitor the number of local references on each virtual circuit which it has both assigned and are in use.

5.7.6.3.2.5.2 When the number of such local references on a given virtual circuit exceeds a System Manager specified threshold, then the local reference cancellation procedures specified in 5.7.6.3.6 shall be invoked, in order to ensure that the number of unused local references in the range in which the SNDCF is permitted to assign local references, is at least equal to a System Manager specified target.

5.7.6.3.2.6 Transfer of the Modified ISO/IEC 8473 PDU

5.7.6.3.2.6.1 The modified ISO/IEC 8473 NPDU (i.e. the NPDU with the added Local Reference Option) shall be inserted in the User Data field of an ISO/IEC 8208 Data packet and shall be sent over the virtual circuit, using the ISO/IEC 8208 M-bit segmentation procedure if appropriate.

5.7.6.3.3 Compression of SN-Userdata

5.7.6.3.3.1 General

5.7.6.3.3.1.1 An Initial DT NPDU shall be compressed according to the procedures specified in 5.7.6.3.3.2.

5.7.6.3.3.1.2 A Derived DT NPDU shall be compressed according to the procedures specified in 5.7.6.3.3.3.

5.7.6.3.3.1.3 An ER NPDU shall be compressed according to the procedures specified in 5.7.6.3.3.4.

Compressed Initial Data PDU Compressed Derived Data PDU

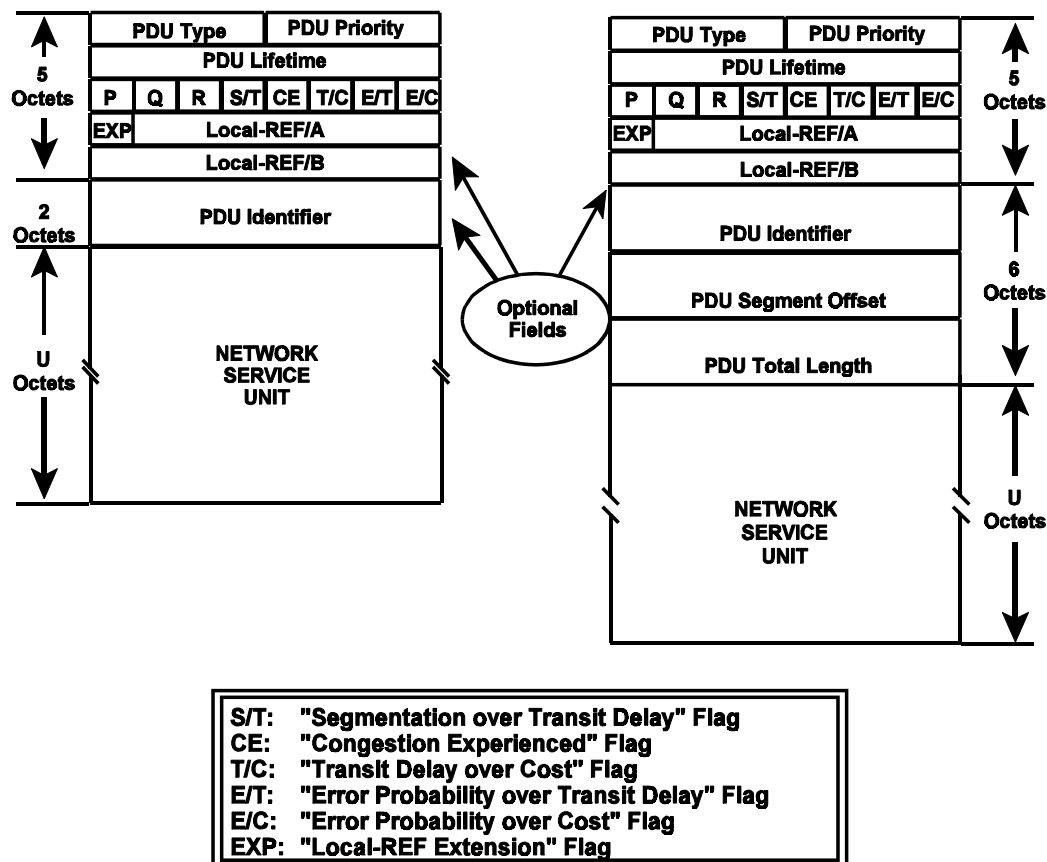


Figure 5.7-4 Compressed Initial and Derived PDU Formats

5.7.6.3.3.2 Initial DT PDU Compression

5.7.6.3.3.2.1 General

Note.— An Initial DT PDU is an ISO/IEC 8473 DT PDU that either contains no Segmentation Part in its PDU header or contains a Segmentation Part with a Segment Offset value that equals zero and the Segment Length is equal to the Total Length.

5.7.6.3.3.2.1.1 The original Initial DT PDU shall be compressed into the Compressed Initial Data PDU as shown in Figure 5.7-4.

5.7.6.3.3.2.1.2 The fields of the Compressed Initial Data PDU shall be set as follows.

5.7.6.3.3.2.2 Type Field

5.7.6.3.3.2.2.1 The PDU Type field value shall be set according to the values of the original Initial DT PDU ER, SP and More Segments (MS) flags as defined in Table 5.7-4.

Table 5.7-4 Initial DT PDU Type Codes

PDU Type Values	CLNP NPDU ER Value	CLNP NPDU SP Value	CLNP NPDU MS Value
0 0 0 0	0	0	0
0 0 0 1	0	1	0
0 0 1 0	1	0	0
0 0 1 1	1	1	0

5.7.6.3.3.2.3 PDU Priority Field

5.7.6.3.3.2.3.1 The PDU Priority field value shall be set to the lowest four bits of the original PDU Priority parameter value field, if the Priority option is present, and set to zero otherwise.

5.7.6.3.3.2.4 PDU Lifetime Field

5.7.6.3.3.2.4.1 The PDU Lifetime field value shall be set to the eight bits of the original NPDU lifetime field.

5.7.6.3.3.2.5 P bit Field

5.7.6.3.3.2.5.1 The P field value shall be set to one if the original uncompressed PDU contained the priority option. This field shall be set to zero otherwise.

5.7.6.3.3.2.6 Q bit Field

5.7.6.3.3.2.6.1 The Q field value shall be set to one if the original uncompressed PDU contained the QoS Maintenance option.

5.7.6.3.3.2.6.2 This field shall be set to zero otherwise.

5.7.6.3.3.2.7 R bit Field

5.7.6.3.3.2.7.1 The R field value shall be set to one if the original uncompressed PDU contains a non-zero checksum.

5.7.6.3.3.2.7.2 This field shall be set to zero otherwise.

5.7.6.3.3.2.8 S/T, CE , T/C, E/T, and E/C Fields

5.7.6.3.3.2.8.1 The values of these fields shall be set to bits 5 through 1 of the QoS parameter value option field of the original PDU, if the Quality of Service maintenance option is present.

5.7.6.3.3.2.8.2 The S/T field shall be set to the value of bit 5 of the Quality of Service Maintenance parameter value field, if present (i.e. sequencing vs. transit delay) and set to zero otherwise.

5.7.6.3.3.2.8.3 The CE field shall be set to the value of bit 4 in the Quality of Service Maintenance parameter value field.

5.7.6.3.3.2.8.4 The T/C field shall be set to the value of bit 3 in the Quality of Service Maintenance parameter value field.

5.7.6.3.3.2.8.5 The E/T field shall be set to the value of bit 2 in the Quality of Service Maintenance parameter value field.

5.7.6.3.3.2.8.6 The E/C field shall be set to the value of bit 1 in the Quality of Service Maintenance parameter value field.

5.7.6.3.3.2.9 EXP, Local-REF/A and Local-REF/B Fields

5.7.6.3.3.2.9.1 If the value of the Local Reference determined according to the procedure specified in 5.7.6.3.2.4 is less than 128, then the EXP field shall be set to zero.

5.7.6.3.3.2.9.2 In this case, only the Local-REF/A field shall be present in the PDU.

5.7.6.3.3.2.9.3 The Local-REF/A field value shall be set to the value of the Local Reference encoded as an unsigned integer.

5.7.6.3.3.2.9.4 If the value of the Local Reference is greater than or equal to 128, the EXP field shall be set to one, and both Local-REF/A and Local-REF/B fields shall be present in the PDU.

5.7.6.3.3.2.9.5 The Local Reference shall be encoded as a 15 bit unsigned integer, with the least significant eight bits placed in the Local-REF/B field, and the most significant seven bits placed in the Local-REF/A field.

5.7.6.3.3.2.10 PDU Identifier

5.7.6.3.3.2.10.1 If the Initial DT PDU allows segmentation (SP Flag is set to one), then the PDU Identifier field shall be included in the Compressed Initial Data PDU.

5.7.6.3.3.2.10.2 The PDU Identifier field shall contain the Data Unit Identifier as provided in the segmentation part of the Initial DT PDU.

5.7.6.3.3.2.10.3 If the Initial DT PDU does not allow segmentation (SP Flag is set to zero), then this field shall not be included in the Compressed Initial Data PDU.

5.7.6.3.3.2.11 PDU Segment Offset

5.7.6.3.3.2.11.1 This field shall not be present in the Compressed Data PDU for an Initial DT PDU.

Note.— The segment offset of an Initial DT PDU is always zero and is a priori known by the receiving SNDCEF.

5.7.6.3.3.2.12 PDU Total Length

5.7.6.3.3.2.12.1 This field shall not be present in the Compressed Data PDU for an Initial DT PDU.

Note.— The Total Length field value of an Initial DT PDU is the length of the entire PDU in octets. This value is identical to the value of the Segment Length field for an Initial DT PDU and both values may be recalculated by the receiving SNDCEF.

5.7.6.3.3.2.13 Network Service Data Unit Field

5.7.6.3.3.2.13.1 This field shall contain the Data Part of the original Initial DT PDU.

5.7.6.3.3.3 Derived DT PDU Compression

5.7.6.3.3.3.1 General

5.7.6.3.3.3.1.1 The original Derived DT PDU shall be compressed into the Compressed Derived Data PDU as shown in Figure 5.7-4.

5.7.6.3.3.3.1.2 The fields of the Compressed Derived Data PDU shall be set as defined in the following sections.

5.7.6.3.3.3.2 Type Field

5.7.6.3.3.3.2.1 The PDU Type field value shall be set according to the values of the original NPDU ER, SP and MS flags as defined in Table 5.7-5.

Table 5.7-5 Derived PDU Type Codes

PDU Type Values	CLNP NPDU ER Value	CLNP NPDU SP Value	CLNP NPDU MS Value
0 1 1 0	0	1	0
0 1 1 1	0	1	1
1 0 0 1	1	1	0
1 0 1 0	1	1	1

5.7.6.3.3.3.3 PDU Priority Field

5.7.6.3.3.3.3.1 This field shall be set as defined in 5.7.6.3.3.2.3.

5.7.6.3.3.3.4 PDU Lifetime Field

5.7.6.3.3.3.4.1 This field shall be set as defined in 5.7.6.3.3.2.4.

5.7.6.3.3.3.5 P bit Field

5.7.6.3.3.3.5.1 This field shall be set as defined in 5.7.6.3.3.2.5.

5.7.6.3.3.3.6 Q bit Field

5.7.6.3.3.3.6.1 This field shall be set as defined in 5.7.6.3.3.2.6.

5.7.6.3.3.3.7 S/T, CE , T/C, E/T, and E/C Fields

5.7.6.3.3.3.7.1 These fields shall be set as defined in 5.7.6.3.3.2.8.

5.7.6.3.3.3.8 EXP, Local-REF/A and Local-REF/B Fields

5.7.6.3.3.3.8.1 These fields shall be set as defined in 5.7.6.3.3.2.9.

5.7.6.3.3.3.9 PDU Identifier Field

5.7.6.3.3.3.9.1 The PDU Identifier field value shall be set to the Data Unit Identifier contained in the segmentation part of the original Derived DT PDU header.

5.7.6.3.3.3.10 PDU Segment Offset Field

5.7.6.3.3.3.10.1 The PDU Segment Offset field value shall be set to the Segment Offset value contained in the segmentation part of the original Derived DT PDU header.

5.7.6.3.3.3.11 PDU Total Length Field

5.7.6.3.3.3.11.1 The PDU Total Length field value shall be set to the value of the Total Length field contained in the Segmentation Part of the original Derived DT PDU.

5.7.6.3.3.4 Error Report PDU Compression

5.7.6.3.3.4.1 General

5.7.6.3.3.4.1.1 The original ER PDU shall be compressed into the Compressed Error Report PDU as shown in Figure 5.7-5.

5.7.6.3.3.4.1.2 The fields of the Compressed Error Report PDU shall be set as defined in the following sections.

5.7.6.3.3.4.2 PDU Type Field

5.7.6.3.3.4.2.1 The PDU Type field value shall be set to [1101].

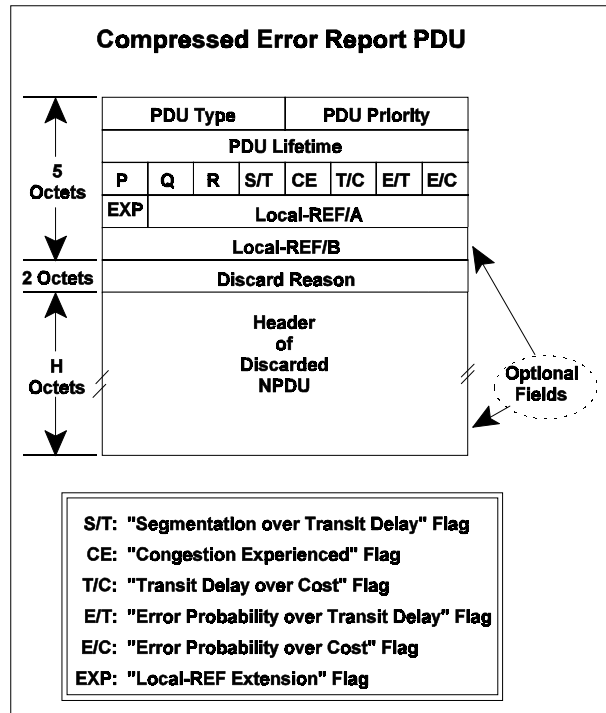


Figure 5.7-5 Compressed Error Report PDU

5.7.6.3.3.4.3 PDU Priority Field

5.7.6.3.3.4.3.1 This field shall be set as defined in 5.7.6.3.3.2.3.

5.7.6.3.3.4.4 PDU Lifetime Field

5.7.6.3.3.4.4.1 This field shall be set as defined in 5.7.6.3.3.2.4.

5.7.6.3.3.4.5 P bit Field

5.7.6.3.3.4.5.1 This field shall be as defined in 5.7.6.3.3.2.5.

5.7.6.3.3.4.6 Q bit Field

5.7.6.3.3.4.6.1 This field shall be set as defined in 5.7.6.3.3.2.6.

5.7.6.3.3.4.7 S/T, CE, T/C, E/T and E/C Fields

5.7.6.3.3.4.7.1 These fields shall be set as defined in 5.7.6.3.3.2.8.

5.7.6.3.3.4.8 EXP, Local-REF/A, Local-REF/B Fields

5.7.6.3.3.4.8.1 These fields shall be set as defined in 5.7.6.3.3.2.9.

5.7.6.3.3.4.9 Discard Reason Field

5.7.6.3.3.4.9.1 This field shall be set to the value of the Reason for Discard Parameter Value field contained in the original NPDU header.

5.7.6.3.3.4.10 Header of Discarded NPDU Field

5.7.6.3.3.4.10.1 This field shall contain the value of the Error Report Data Part if provided in the original Error Report PDU.

5.7.6.3.3.4.11 Transfer of Compressed ISO/IEC 8473 PDUs

5.7.6.3.3.4.11.1 The compressed ISO/IEC 8473 NPDU (i.e. Compressed Initial Data PDU, Compressed Derived Data PDU, or Compressed Error Report PDU) shall be inserted in the User Data field of an ISO/IEC 8208 Data packet and shall be sent over the virtual circuit, using the ISO/IEC 8208 M-bit segmentation procedure if appropriate.

5.7.6.3.4 Processing of Packets Received from the Subnetwork Service Provider

Note.— The following sections specify the processing of packets received from the Subnetwork Service provider.

5.7.6.3.4.1 Initial Processing of NPDU

5.7.6.3.4.1.1 On receipt of an incoming packet received from a virtual circuit, the SND CF shall inspect the first octet to determine the Network Layer Protocol ID or the compressed PDU type (see Table 5.7-6).

- a) If this value is set to **[1000 0001]** indicating that the NPDU is an ISO/IEC 8473 NPDU with an uncompressed header, then the NPDU shall be processed according to 5.7.6.3.4.2.2.
- b) If the first octet indicates either ISO/IEC 9542 (ES-IS), ISO/IEC 11577 (NLSP) or ISO/IEC 10589 (IS-IS), the SND CF shall generate an SN-UNITDATA.indication with the NPDU as its SN-Userdata parameter, and the SN-Source-Address and SN-Destination-Address parameters set to the remote and local DTE addresses for the virtual circuit over which the NPDU was received.
- c) If the value of the first four bits of the first octet is in the range binary **[0000]** to binary **[0011]** then the PDU is a compressed ISO/IEC 8473 Initial DT PDU which shall be decompressed using the procedures specified in 5.7.6.3.4.3.
- d) If the value of the first four bits of the first octet is in the range binary **[0110]** to binary **[1010]** (excluding 1000) then the PDU is a compressed ISO/IEC 8473 Derived PDU, which shall be decompressed using the procedures specified in 5.7.6.3.4.3.
- e) If the value of the first four bits of the first octet is binary **[1101]** then the PDU is a compressed ISO/IEC 8473 Error PDU, which shall be decompressed using the procedures specified in 5.7.6.3.4.4.

- f) If the value of the first four bits of the first octet is binary **[1110]** then the PDU is an SNDCF Error Report, which shall be processed according to the procedures of 5.7.6.3.4.5, and no SN-UNITDATA.indication generated.
- g) If the value of the first four bits of the first octet is binary **[0100]** or binary **[0101]**, then the PDU is respectively, a local reference cancellation request or response, which shall be processed according to the procedures of 5.7.6.3.6 and no SN-UNITDATA.indication generated.

Table 5.7-6 Mapping between Compressed PDU Type Fields and Uncompressed PDU Types

Compressed PDU Type Field	Uncompressed PDU Type
[0000] - [0011]	Compressed Initial DT PDU
[0110] - [0111] [1001] - [1010]	Compressed Derived DT PDU
[1101]	Compressed Error Report PDU
[1110]	SNDCF Error Report
[0100]	Cancellation Request PDU
[0101]	Cancellation Accept PDU

5.7.6.3.4.1.2 In all other cases, the PDU shall be discarded and an SNDCF Error Report Generated (see 5.7.6.3.5).

5.7.6.3.4.2 Incoming ISO/IEC 8473 PDU with Uncompressed Header

5.7.6.3.4.2.1 General

5.7.6.3.4.2.1.1 If the received NPDU is an ISO/IEC 8473 NPDU then the options part shall be inspected for the options field containing the local reference.

5.7.6.3.4.2.2 Processing of Unmodified ISO/IEC 8473 PDUs

5.7.6.3.4.2.2.1 If the local reference option is not present, then the SNDCF shall generate a SN-UNITDATA indication with the NPDU as its SN-Userdata, and the SN-Source-Address and SN-Destination-Address parameters set to the remote and local DTE addresses for the virtual circuit over which the NPDU was received.

5.7.6.3.4.2.3 Processing of Modified ISO/IEC 8473 PDUs

5.7.6.3.4.2.3.1 If the Local Reference option is present, it shall be removed, and the checksum and PDU header length indication and segment length shall be modified to reflect this removal.

5.7.6.3.4.2.3.2 If a Local Reference options field is present, then the local directory associated with the virtual circuit over which the PDU was received shall be inspected for the presence of the corresponding entry.

5.7.6.3.4.2.3.3 If no such entry is present, and the value of the Local Reference is in the range within which the remote SNDCF is permitted to create local directory entries, then the entry shall be created, and:

- a) The value of the inward NSAP Address set to the PDU's destination NSAP Address,
- b) The value of the outward NSAP Address set to the NSAP's source NSAP Address, and
- c) The values of the Version Number and Security Parameter, set to the corresponding values in the PDU header.

5.7.6.3.4.2.3.4 An SNDCF Error Report (see 5.7.6.3.5) shall be generated if the value of the Local Reference is not within the range within which the remote SNDCF is permitted to create local directory entries, or is greater than the maximum negotiated when the call was established.

5.7.6.3.4.2.3.5 Otherwise, the local directory entry shall be compared with the received PDU. If:

- a) The inward NSAP Address does not match the destination NSAP Address, or
- b) The outward NSAP Address does not match the source NSAP Address, or
- c) The Version Number does not match the Version Number present in the directory entry, or
- d) The value of the Security options parameter does not match the value in the directory, or is not correspondingly absent, then

an SNDCF Error Report shall be generated and returned over the same virtual circuit as the PDU was received.

5.7.6.3.4.2.3.6 The SNDCF shall then generate a SN-UNITDATA.indication with the NPDU as its SN-Userdata, and the SN-Source-Address and SN-Destination-Address parameters set to the remote and local DTE addresses for the virtual circuit over which the NPDU was received.

5.7.6.3.4.3 Incoming Compressed Data PDU

5.7.6.3.4.3.1 General

5.7.6.3.4.3.1.1 If the most significant four bits of the first octet of a received PDU (i.e. the PDU Type field) are in the range [0000] to [0011] binary, excluding [1000], then the packet is a compressed ISO/IEC 8473 Initial DT NPDU.

5.7.6.3.4.3.1.2 If the PDU Type field of a received compressed PDU is in the range [0110] to [1010] binary, then the PDU is a compressed ISO/IEC 8473 Derived DT NPDU.

5.7.6.3.4.3.1.3 Upon receipt, the SNDCF shall examine and validate the Local-REF in the compressed PDU.

5.7.6.3.4.3.1.4 The value of the Local Reference shall be extracted from the compressed header and the corresponding entry in the local directory located.

5.7.6.3.4.3.1.5 If no entry exists corresponding to the Local-REF present in the PDU, then an SNDCF Error Report shall be generated and returned over the same virtual circuit as the PDU was received, and the PDU shall be discarded.

5.7.6.3.4.3.1.6 If the Local-REF is valid, the original uncompressed NPDU shall be recreated by the procedures defined in 5.7.6.3.4.3.2 through 5.7.6.3.4.3.6.

5.7.6.3.4.3.1.7 The SNDCF then shall generate a SN-UNITDATA.indication with the SN-Source Address and SN-Destination Address parameters set to the remote and local DTE addresses for the virtual circuit over which the NPDU was received, and the SN-Userdata shall be set to the uncompressed DT NPDU.

5.7.6.3.4.3.2 Fixed Part

Note 1.— The Fixed Part of the NPDU header consists of the Network Layer Protocol Identifier, Length Indicator, Version/Protocol Identifier Extension, PDU Lifetime, SP flag, MS flag, E/R Flag, Type, Segment Length and Checksum fields as defined in ISO/IEC 8473.

Note 2.— If the EXP field is set to zero, the Local Reference is the seven bit integer value of the Local-REF/A field. If the EXP field is set to one, the Local Reference value consists of the fifteen bit unsigned integer as stored with the least significant eight bits placed in the Local-REF/B field, and the most significant seven bits placed in the Local-REF/A field.

5.7.6.3.4.3.2.1 Network Layer Protocol Identifier

5.7.6.3.4.3.2.1.1 This field shall be set to binary [1000 0001] to identify this Network Layer Protocol as ISO/IEC 8473.

5.7.6.3.4.3.2.2 Length Indicator

5.7.6.3.4.3.2.2.1 This field shall be set to the length of the uncompressed NPDU header in octets.

5.7.6.3.4.3.2.3 Version/Protocol Identifier Extension

5.7.6.3.4.3.2.3.1 The Version/Protocol Identifier Extension field shall be set to the values provided in the corresponding entry of the local directory.

5.7.6.3.4.3.2.4 PDU Lifetime

5.7.6.3.4.3.2.4.1 The eight bits of the PDU Lifetime field shall be set to the eight bits of the PDU Lifetime field of the Compressed Data PDU.

5.7.6.3.4.3.2.5 Segmentation Permitted, More Segments, Error Report Flags

5.7.6.3.4.3.2.5.1 The values of these flags shall be derived from the value of the Protocol ID field and Type field of the Compressed Data PDU.

5.7.6.3.4.3.2.5.2 These flag values shall be determined according to Table 5.7-4 for an Initial Data PDU and Table 5.7-5 for a Derived Data PDU.

5.7.6.3.4.3.2.6 Type Code

5.7.6.3.4.3.2.6.1 This field shall be set to binary [11100] to indicate a DT PDU.

5.7.6.3.4.3.2.7 Segment Length

5.7.6.3.4.3.2.7.1 This field shall indicate the entire length in octets of the PDU, including both header and data.

5.7.6.3.4.3.2.7.2 The value of this field shall be computed by the SND CF.

5.7.6.3.4.3.2.7.3 For an Initial DT NPDU, the value of this field shall be identical to the value of the Total Length field located in the Segmentation Part of the header.

5.7.6.3.4.3.2.8 PDU Checksum

5.7.6.3.4.3.2.8.1 The value of this field shall be set to zero if the R bit in the compressed header is zero.

5.7.6.3.4.3.2.8.2 Otherwise, a Checksum field shall be recomputed.

Note.— For the DT PDU, this includes the segmentation and options part (if present). For the Error Report PDU, this includes the reason for discard field as well.

5.7.6.3.4.3.3 Address Part

Note.— The Address Part consists of the Destination Address Length Indicator, Destination Address, Source Address Length Indicator and Source Address as defined in ISO/IEC 8473.

5.7.6.3.4.3.3.1 Destination and Source Address Length Indicators and Addresses

5.7.6.3.4.3.3.1.1 The Source and Destination NSAP addresses shall be set to the values provided in the corresponding entry of the local directory for the Local Reference number calculated.

5.7.6.3.4.3.3.1.2 The source NSAP Address shall be set to the value of the outward NSAP Address, and the destination NSAP Address set to the value of the inward NSAP Address.

5.7.6.3.4.3.3.1.3 The Length fields shall contain the length of each address in octets.

5.7.6.3.4.3.4 Segmentation Part

5.7.6.3.4.3.4.1 General

5.7.6.3.4.3.4.1.1 If the ISO/IEC 8473 SP field is set to one, then the Segmentation Part shall be generated.

5.7.6.3.4.3.4.1.2 The Segmentation Part shall consist of the Data Unit Identifier, Segment Offset, and Total Length field as defined in ISO/IEC 8473.

5.7.6.3.4.3.4.2 Data Unit Identifier

5.7.6.3.4.3.4.2.1 This field shall contain the value of the PDU Identifier field as provided in the compressed DT PDU.

5.7.6.3.4.3.4.3 Segment Offset

5.7.6.3.4.3.4.3.1 For an Initial DT PDU, this field shall be set to zero.

5.7.6.3.4.3.4.3.2 For a Derived DT PDU, this field shall be set to the PDU Segment Offset field as provided in the compressed DT PDU.

5.7.6.3.4.3.4.4 PDU Total Length

5.7.6.3.4.3.4.4.1 For a Derived DT PDU, this field shall contain the value of the PDU Total Length field as provided in the Compressed DT PDU.

5.7.6.3.4.3.4.4.2 For an Initial PDU, the entire length of the PDU in octets shall be calculated by the SNDCF and stored in this field.

5.7.6.3.4.3.5 Options Part

5.7.6.3.4.3.5.1 General

5.7.6.3.4.3.5.1.1 If the Q bit field is set to one, the Globally Unique QoS option shall be recreated according to 5.7.6.3.4.3.5.3.

5.7.6.3.4.3.5.1.2 If the Security option is present in the local reference directory entry, the Security option shall be recreated according to 5.7.6.3.4.3.5.4.

5.7.6.3.4.3.5.1.3 If the P bit field is set to one, the Priority option shall be recreated according to 5.7.6.3.4.3.5.2.

5.7.6.3.4.3.5.2 Priority

5.7.6.3.4.3.5.2.1 For the Priority option, the Parameter Code shall be set to binary [1100 1101] and the Parameter Length set to one octet.

5.7.6.3.4.3.5.2.2 The four most significant bits of the Parameter Value shall be set to zero, and the four least significant bits set to the PDU Priority field as provided in the compressed DT PDU.

5.7.6.3.4.3.5.3 Quality of Service Maintenance

5.7.6.3.4.3.5.3.1 For the Quality of Service Maintenance option, the Parameter Code shall be set to binary [1100 0011], the Parameter Length set to one octet.

5.7.6.3.4.3.5.3.2 The high order two bits of the Parameter Value shall be set to binary [11] to indicate Globally Unique, bit 6 shall be set to zero, and bits 5 through one set to the S/T, CE, T/C, E/T and E/C fields respectively as provided in the compressed Data PDU.

5.7.6.3.4.3.5.4 Security

5.7.6.3.4.3.5.4.1 This field shall be set to the value of the Security parameter contained in the corresponding Local Reference directory entry.

5.7.6.3.4.3.6 Data Part

5.7.6.3.4.3.6.1 The Data Part shall be copied from the Compressed Data PDU data part.

5.7.6.3.4.4 Incoming Compressed Error Report PDU

5.7.6.3.4.4.1 General

5.7.6.3.4.4.1.1 The original uncompressed header shall be recreated as defined in the following sections.

Note.— If the four most significant bits of the first octet (the PDU Type Field) of a received packet are [1101] then the packet is a compressed ISO/IEC 8473 ER NPDU.

5.7.6.3.4.4.2 Fixed Part

5.7.6.3.4.4.2.1 The Fixed Part of the ER PDU shall be composed in the same manner as defined in 5.7.6.3.4.3.2 except for the Type Code which shall be set to binary [00001] to indicate an ER PDU, and for the SP and MS flags which shall be set to zeros.

5.7.6.3.4.4.3 Address Part

5.7.6.3.4.4.3.1 The Address Part of the ER PDU shall be composed in the same manner as defined in 5.7.6.3.4.3.3.

5.7.6.3.4.4.4 Options Part

5.7.6.3.4.4.4.1 The Options Part of the ER PDU shall be composed in the same manner as defined in 5.7.6.3.4.3.5 for an Initial DT PDU.

5.7.6.3.4.4.5 Reason for Discard

5.7.6.3.4.4.5.1 To compose this field, the Parameter Code shall be set to binary [1100 0001], the Parameter Length set to two octets, and the Parameter Value set to the Discard Reason field as provided in the Compressed Error Report PDU.

5.7.6.3.4.4.6 Error Report Data Part

5.7.6.3.4.4.6.1 If the Compressed Error Report PDU contains the Header of Discarded NPDU field, then the Error Report Data Part shall be set to the value of the Header of Discarded NPDU field.

5.7.6.3.4.5 Incoming SNDCF Error Report

5.7.6.3.4.5.1 On receipt of an SNDCF Error Report with reason “compressed NPDU with unrecognized local reference”, the directory entry corresponding to the local reference returned in the SNDCF Error Report shall be reset to the unused state.

5.7.6.3.4.5.2 On receipt of an SNDCF Error Report with reason other than “compressed NPDU with unrecognized local reference” (see Table 5.7-7), the virtual circuit shall be reset (see 5.7.6.3.7) and the local reference directory associated with the virtual circuit shall be cleared to its initial state.

Note.— If the virtual circuit on which the error has been reported belongs to a connection group which shares the same LREF directory, there is no need to reset the remaining virtual circuits of that group.

5.7.6.3.4.5.3 **Recommendation.**— *The error should be notified to Systems Management.*

Note.— If the four most significant bits of the first octet (the PDU Type field) of an incoming packet are set to [1110], then a SNDCF Error Report has been received (see 5.7.6.3.5).

5.7.6.3.5 SNDCF Error Report

5.7.6.3.5.1 The SNDCF Error Report is a packet format unique to the Mobile SNDCF, and shall be used to report errors in the use of local references as specified below. The SNDCF Error Report PDU shall be constructed as follows:

- a) The most significant four bits (PDU Type) of the first octet are set to binary 1110, while the least significant four bits are set to 0000.
- b) The second octet is a discard reason encoded as an unsigned integer, with the following reason codes defined in the Table 5.7-7:

Table 5.7-7 SNDCF Error Report Diagnostic Codes

Code	Reason
[0000 0000]	Compressed NPDU with unrecognized Local Reference
[0000 0001]	Creation of directory entry outside of sender's permitted range
[0000 0010]	Directory entry exists
[0000 0011]	Local Reference greater than maximum value accepted
[0000 0100]	Data Unit Identifier missing when SP=1
[0000 0101]	Reserved
[0000 0111]	Compressed ISO/IEC 8473 PDU with unrecognized Type
[0000 1000]	Local Reference Cancellation Error

- c) The Local Reference contained in the PDU for which the error is being reported is placed in the remaining octet(s) of the SNDCEF Error Report PDU Header, unless the reason is Local Reference Cancellation Error, when the SNDCEF Error Report shall consist of three octets only, and the third octet shall contain the Cancellation Reference of the invalid Cancellation Request PDU.

5.7.6.3.5.2 The data portion of the SNDCEF Error Report shall be used to return a copy of the PDU in error, similar to the ISO/IEC 8473 Error Report PDU.

5.7.6.3.5.3 The Error Report PDU shall be sent as an ISO/IEC 8208 DATA packet(s) and, if needed, segmented using the M-bit procedures.

5.7.6.3.6 Local Reference Cancellation Option

5.7.6.3.6.1 General

Note.— When the implementation of this option has been agreed by both SNDCEFs using a virtual circuit during the call setup procedures, then the following procedures may be used to selectively cancel one or more Local References, i.e. make them available for re-use. An SNDCEF may only request the cancellation of Local References which are within the range in which it is permitted to assign Local References.

5.7.6.3.6.1.1 When an SNDCEF invokes the procedures for Local Reference cancellation it shall format a Cancellation Request PDU, as specified below, and send the PDU to the other SNDCEF over the virtual circuit to which it applies.

5.7.6.3.6.1.2 A Cancellation Request PDU shall be retransmitted periodically until it is acknowledged by a cancellation accept PDU, or an SNDCEF Error Report PDU is received indicating an error in the request.

5.7.6.3.6.1.3 When a Cancellation Accept PDU is received, the corresponding directory entries shall be cleared, and the Local References therefore become available for re-use.

5.7.6.3.6.1.4 When an SNDCEF receives a Cancellation Request PDU, it shall first check to ensure that the local references identified in the PDU are within the range in which the sending SNDCEF is permitted to assign local references.

5.7.6.3.6.1.5 If any one of them is not, then an SNDCEF error report shall be returned, and the request ignored.

5.7.6.3.6.1.6 Otherwise, the directory entries corresponding to the indicated local references shall be cleared, and a cancellation accept PDU be formatted and returned, in order to accept cancellation of these local references.

5.7.6.3.6.2 The Cancellation Request PDU

5.7.6.3.6.2.1 The PDU format shall be as illustrated in Figure 5.7-6. The first octet shall be set to [0100 0000]. The remainder of the PDU shall consist of:

- a) A Cancellation Reference expressed as a one octet unsigned integer, and which uniquely identifies this Cancellation Request within the context of the virtual circuit.

Note.— In most cases uniqueness will be assured if the reference is implemented as a sequence number starting at zero and incremented by one (modulo 256), each time a Cancellation Request is sent.

- b) A length octet (L1) given as an unsigned integer (0 to 255), which indicates the length in octets of the set of individual Local References to cancel.

PDU Type		Unused
Cancellation Reference		
L1		
EXP	Local-REF/A	
Local-REF/B		
.		
.		
.		
L2		
EXP	Local-REF/A	
Local-REF/B		
.		
.		
.		

Figure 5.7-6 Cancellation Request PDU

- c) One or more Local References expressed as one or two octets each, as appropriate, and encoded in successive octets, with the total number of octets containing such local references given by L1.
- d) A length octet (L2) given as an unsigned integer (0 to 255), which indicates the length in octets of the set of inclusive Local Reference ranges to cancel.

- e) One or more pairs of Local Reference ranges expressed as one or two octets each, as appropriate, and encoded in successive octets, with the total number of octets containing such Local References given by L2.

5.7.6.3.6.2.2 In each of the above cases, if the value of a local reference is less than 128, then bit eight of the first octet in which it is encoded shall be set to zero, and the remaining seven bits set to the value of the Local Reference encoded as an unsigned integer.

5.7.6.3.6.2.3 The extended Local Reference octet shall not be present.

5.7.6.3.6.2.4 Otherwise, bit eight shall be set to one, and the remaining seven bits and the next octet set to the value of the Local Reference encoded as a 15 bit unsigned integer, with the least significant eight bits placed in the extended Local Reference octet, and the most significant seven bits placed in the first octet.

Note.— This format allows for the Local References to be cancelled, to be expressed as either a set of individual references, or a set of inclusive ranges of individual references, or both.

5.7.6.3.6.3 The Cancellation Accept PDU

5.7.6.3.6.3.1 The PDU format shall be as illustrated in Figure 5.7-7.

PDU Type	Unused
Cancellation Reference	

Figure 5.7-7 Cancellation Accept PDU

5.7.6.3.6.3.2 The first octet shall be set to binary [0101 0000], and the second octet set to the Cancellation Reference of the Cancellation Request which is being accepted.

5.7.6.3.7 Call Reset Provisions

5.7.6.3.7.1 If at any time, a Reset Indication is received indicating a DCE originated reset, then this shall be confirmed and all other procedures associated with the Call Reset performed.

Note.— There is otherwise no impact on this SND CF.

5.7.6.3.7.2 If the Reset Indication indicates a DTE user originated reset then, additionally, the directory associated with the virtual circuit shall be cleared to its initial state.

5.7.6.3.8 Call Clearing and LREF Procedures

5.7.6.3.8.1 When a virtual circuit has been terminated and the corresponding Subnetwork Connection Group is now empty, then the Local Reference Directory associated with this group shall be discarded.

5.7.6.4 ATN NSAP Compression Algorithm (ACA)

5.7.6.4.1 General Overview

5.7.6.4.1.1 When negotiated in the Mobile SNDCF Call establishment phase, the optional ATN NSAP Compression Algorithm (ACA) shall be applied as follows:

- a) the compression processing (5.7.6.4.5) to data octets being output to the subnetwork, and
- b) the decompression processing (5.7.6.4.6) to data octets input from the subnetwork.

5.7.6.4.2 Address Length Determination

5.7.6.4.2.1 The address length for the address or address prefix to be compressed shall be extracted from the octet preceding the AFI octet in the uncompressed data stream.

5.7.6.4.2.2 If the extracted length lies in the range 7 through 20, the extracted length shall be used as the address “octet length” and the address length type shall be indicated as “normal”.

5.7.6.4.2.3 If the extracted length lies in the range 56 through 160 and is an integral multiple of 8, the extracted length shall be divided by 8 to compute the length in octets of the address prefix and the address length type shall be indicated as “IDRP”.

5.7.6.4.2.4 If the extracted length does not lie in either of these ranges, the input data does not form a compressible ATN NSAP address and the ACA shall not further process the current data as a compressible ATN NSAP address.

5.7.6.4.2.5 The octet length for ACA compressed address prefixes shall be encoded in the first header octet LEN/SEL subfield and the FP subfield shall be set to one.

5.7.6.4.2.6 If the octet length for the ACA compressed address is 20 (indicating a full address instead of a prefix) the FP subfield shall be set to zero.

5.7.6.4.2.7 The explicit address length octet shall be removed as part of the ACA compression processing.

Note 1.— No length octet is required for compressed ACA addresses. All information concerning address length and the presence or length of variable-length fields is contained in the header octets.

Note 2.— The shortest ATN NSAP address prefix that can be compressed is 7 octets and the length of a full ATN address is 20 octets.

Note 3.— Address lengths for normal addresses and prefixes are expressed in octet units. The address lengths for IDRP addresses and prefixes are expressed in bit units (even though the address lengths are always in full octets).

Note 4.— The IDRP subfield in the first header octet indicates whether the expanded address used octet or bit length units. Internal (compressed) addresses assume octet lengths for encoding.

5.7.6.4.3 Compressed Address Structure

5.7.6.4.3.1 General

5.7.6.4.3.1.1 An ACA compressed address or address prefix shall consist of the following components in the order shown in Table 5.7-8.

Table 5.7-8 Compressed NSAP Address Format

Name	Length (octets)	Reference
Address Marker	2	5.7.6.4.4
Header Octet 1	1	5.7.6.4.3.2.2
Header Octet 2	1	5.7.6.4.3.2.3
Compressed ADM	2 or 3	5.7.6.4.3.3
Variable Fields	0 to 14	5.7.6.4.3.4

5.7.6.4.3.1.2 The coding and use of each component shall be as defined below.

Note.— Multi-octet uncompressed ATN address fields (ADM, ARS, LOC, and SYS) are processed from left to right, i.e. from most-significant to least-significant octet.

5.7.6.4.3.2 Address Header Octets

5.7.6.4.3.2.1 General

5.7.6.4.3.2.1.1 Two header octets shall begin each compressed address or address prefix.

5.7.6.4.3.2.1.2 All bits of these header octets shall be set to zero unless otherwise specified in the following subparagraphs.

5.7.6.4.3.2.1.3 Bits in each header octet shall be assigned from the high-order (most-significant or left-most).

Note.— The value of the first header octet is never zero for any compressed address. This prevents confusing a compressed address with an embedded address marker (5.7.6.4.4.3).

5.7.6.4.3.2.2 First Header Octet

5.7.6.4.3.2.2.1 General

5.7.6.4.3.2.2.1.1 The first header octet of a compressed address shall be subdivided into four subfields as follows:

Table 5.7-9 Subfield Structure of First Header Octet

Name	Length (bits)	Comments
IDRP	1	Units of address length
FP	1	Full address or prefix
LEN/SEL	3	Address length or SEL code
CVER	3	Compressed VER value

5.7.6.4.3.2.2.1.2 The coding and use of each subfield shall be as defined below.

5.7.6.4.3.2.2.2 IDRP Subfield

5.7.6.4.3.2.2.2.1 If the address length determination process (5.7.6.4.1, 5.7.6.4.2) indicates that the address to be compressed expresses length in octet units, the IDRP subfield shall be set to zero.

5.7.6.4.3.2.2.2.2 If the address expresses length in bit units (i.e. IDRP address), the IDRP subfield shall be set to one.

5.7.6.4.3.2.2.3 FP Subfield

5.7.6.4.3.2.2.3.1 The FP subfield shall be set to one if the address to be compressed is an address prefix.

5.7.6.4.3.2.2.3.2 The FP subfield shall be set to zero if the address to be compressed is a full address (i.e. its octet length is 20).

5.7.6.4.3.2.2.4 LEN/SEL Subfield

5.7.6.4.3.2.2.4.1 If the address to be compressed is an address prefix (the FP subfield is set to one), the LEN/SEL subfield shall be set to the the prefix length encoded using the encodings in Table 5.7-10.

Table 5.7-10 Prefix Length Codes to be Used in LEN/SEL Subfield

Length	Encoding	Comments
--	0	reserved
7	1	end with ADM
8	2	end with RDF
11	3	end with ARS
13	4	end with LOC
19	5	end with SYS
--	6, 7	unassigned

5.7.6.4.3.2.2.4.2 If the length is not found in this encoding table then the input data does not form an ATN NSAP address prefix that can be compressed and the address prefix shall not be further processed.

5.7.6.4.3.2.2.4.3 If the address to be compressed is a full address (the FP subfield is set to zero), the LEN/SEL subfield shall be set to the encoded value of the address SEL field (5.4.3.8.7) using encodings in Table 5.7-11.

Table 5.7-11 SEL Field Value Codes to be Used in LEN/SEL Subfield

SEL	Encoding	Comments
--	0	reserved
00 hex	1	NET
fe hex	2	NET of an airborne router not supporting IDRP
--	3, 4, 5, 6	unassigned
--	7	other SEL codes

5.7.6.4.3.2.2.4.4 If the SEL field value in the address to be compressed is not one of the table entries above, the LEN/SEL encoding value shall be set to 7.

Note.— A LEN/SEL subfield value of zero is not allowed in either encoding to insure that the first header octet can never have the value [00] hexadecimal. Hence, no compressed address can be confused with an embedded address marker (5.7.6.4.4.3).

5.7.6.4.3.2.2.5 CVER Subfield

5.7.6.4.3.2.2.5.1 If the value of the VER field in the address is in the range [01- 07], [41- 47], [81- 87], or [c1- c7], then the CVER subfield shall be set to the low-order 3 bits of the VER value.

5.7.6.4.3.2.2.5.2 If the value of the VER field in the address is not in one of the above ranges, then the CVER subfield shall be set to zero.

Note.— The encoding of the VER field in an ATN address is defined in 5.4.3.8.1.

5.7.6.4.3.2.3 Second Header Octet

5.7.6.4.3.2.3.1 General

5.7.6.4.3.2.3.1.1 The second header octet of a compressed address shall be subdivided into 8 subfields as follows:

Table 5.7-12 Subfield Structure of Second Header Octet

Name	Length (bits)	Comments
ADMF	1	Flag-compressed ADM value
T/I	1	ATSC/AINSC
F/M	1	Fixed/Mobile
ARSD	1	Flag-defaulted ARS value
LOCD	1	Flag-defaulted LOC value
SYS6	1	Flag-octet 6 of SYS = 0
SYS5	1	Flag-octet 5 of SYS = 0
SYS4	1	Flag-octet 4 of SYS = 0

5.7.6.4.3.2.3.1.2 The encodings and use of each subfield shall be as defined below.

5.7.6.4.3.2.3.2 ADMF Subfield

5.7.6.4.3.2.3.2.1 The ADMF subfield shall be set to one if the ADM value in the address to be compressed may be encoded into two octets using the identifier metacharacter syntax.

5.7.6.4.3.2.3.2.2 The ADMF subfield shall be set to zero if the ADM value in the address to be compressed cannot be expressed using the identifier metacharacter syntax.

Note.— The ADM value can be compressed if each of its three octets contain a character from one of the following character classes:

- a) An upper-case letter “A-Z”
- b) A decimal digit “0-9”
- c) The “@” character.

5.7.6.4.3.2.3.3 T/I Subfield

5.7.6.4.3.2.3.3.1 The T/I subfield shall be set to zero if the VER value in the address to be compressed lies in the ranges [01]-[3f] or [41]-[7f], indicating that the address is in the AINSC domain.

5.7.6.4.3.2.3.3.2 The T/I subfield shall be set to one if the VER value in the address to be compressed lies in the ranges [81]-[bf] or [c1]-[ff], indicating that the address is in the ATSC domain.

5.7.6.4.3.2.3.3.3 If the VER value in the address to be compressed is either [00], [40], [80], or [c0], then the T/I subfield shall be set to zero.

Note.— The encoding of the VER field in an ATN address is defined in 5.4.3.8.1.

5.7.6.4.3.2.3.4 F/M Subfield

5.7.6.4.3.2.3.4.1 The F/M subfield shall be set to zero if the VER value in the address to be compressed lies in the ranges [01]-[3f] or [81]-[bf], indicating that the address is a fixed system.

5.7.6.4.3.2.3.4.2 The F/M subfield shall be set to one if the VER value in the address to be compressed lies in the ranges [41]-[7f] or [c1]-[ff], indicating that the address is a Mobile system.

Note.— The values [00], [40], [80] and [c0] are not used in the VER field of an ATN address (see 5.4.3.8.1)

5.7.6.4.3.2.3.5 ARSD Subfield

5.7.6.4.3.2.3.5.1 The ARSD subfield shall be set to zero if the ARS value in the address to be compressed is not the default value ([000001] hexadecimal) or if the address prefix to be compressed does not include an ARS field.

5.7.6.4.3.2.3.5.2 The ARSD subfield shall be set to one if the ARS value in the address to be compressed has the default value.

5.7.6.4.3.2.3.6 LOCD Subfield

5.7.6.4.3.2.3.6.1 The LOCD subfield shall be set to zero if the LOC value in the address to be compressed is not the default value ([0001] hexadecimal) or if the address prefix to be compressed does not include a LOC field. The LOCD subfield shall be set to one if the LOC value in the address to be compressed has the default value.

5.7.6.4.3.2.3.7 SYS6 Subfield

5.7.6.4.3.2.3.7.1 The SYS6 subfield shall be set to zero if the value of the high-order (6th) octet of the SYS field in the address to be compressed is zero or if the address prefix to be compressed does not include a SYS field.

5.7.6.4.3.2.3.7.2 The SYS6 subfield shall be set to one if the value of the high-order (6th) octet of the SYS field in the address to be compressed is nonzero.

5.7.6.4.3.2.3.8 SYS5 Subfield

5.7.6.4.3.2.3.8.1 The SYS5 subfield shall be set to zero if the value of the second to high-order (5th) octet of the SYS field in the address to be compressed is zero or if the address prefix to be compressed does not include a SYS field.

5.7.6.4.3.2.3.8.2 The SYS5 subfield shall be set to one if the value of the second to high-order (5th) octet of the SYS field in the address to be compressed is nonzero.

5.7.6.4.3.2.3.9 SYS4 Subfield

5.7.6.4.3.2.3.9.1 The SYS4 subfield shall be set to zero if the value of the third to high-order (4th) octet of the SYS field in the address to be compressed is zero or if the address prefix to be compressed does not include a SYS field.

5.7.6.4.3.2.3.9.2 The SYS4 subfield shall be set to one if the value of the third to high-order (4th) octet of the SYS field in the address to be compressed is nonzero.

5.7.6.4.3.3 Compressed ADM Field

5.7.6.4.3.3.1 If the ADM field value of the address to be compressed follows the syntax of an identifier then the compressed ADM field shall consist of two octets and shall contain the encoded value of the ADM field identifier.

5.7.6.4.3.3.2 If the ADM field value of the address to be compressed does not follow the identifier syntax then the compressed ADM field shall consist of three octets and shall contain the 3-octet ADM value unchanged.

Note.— The value of the ADMF subfield in the second header octet indicates whether the compressed ADM field has the 2-octet (compressed) or 3-octet (uncompressed) format.

5.7.6.4.3.4 Variable Fields

5.7.6.4.3.4.1 The variable fields shall have a minimum length of 0 octets and a maximum length of 13 octets. Variable field data octets shall be concatenated when required in the order that their fields occur in the ATN address (Figure 5.7-3) as follows:

- a) VER value (if > 7), 1 octet
- b) ARS value (if not default), 3 octets
- c) LOC value (if not default), 2 octets
- d) SYS octet 6 value (if nonzero), 1 octet
- e) SYS octet 5 value (if nonzero), 1 octet
- f) SYS octet 4 value (if nonzero), 1 octet
- g) SYS octets 3-1, 3 octets
- h) SEL value (if not defined in 5.4.3.8.7.2 or 5.4.3.8.7.3), 1 octet

5.7.6.4.3.4.2 The ACA compression of address prefixes shall omit those variable fields b) through h) which are not present in the uncompressed address prefix.

5.7.6.4.4 Compressed Address Marker

5.7.6.4.4.1 General

5.7.6.4.4.1.1 The ACA shall prefix each compressed address or address prefix with an address marker.

5.7.6.4.4.1.2 The address marker shall consist of two octets with the value [55aa] hexadecimal.

5.7.6.4.4.1.3 The ACA shall process the case of the address marker value occurring in the input octet stream as defined in 5.7.6.4.4.3 below.

5.7.6.4.4.2 Normal Address Case

5.7.6.4.4.2.1 In the case of a normal compressed address or address prefix, the header octets of the compressed address format (5.7.6.4.3) shall follow the address marker.

Note.— The first header octet of a compressed address can never have the value [00]. This distinguishes the normal address case from the embedded address case.

5.7.6.4.4.3 Embedded Address Marker Case

5.7.6.4.4.3.1 If two octets with the value of an address marker occur in data, a padding octet with value [00] hexadecimal shall be inserted into the data stream following the embedded address marker octets.

Note 1.— The likelihood of embedded address markers in the input data stream is very low. When they occur, however, the ACA algorithm must add the extra padding octet. Hence, it is possible (although highly unlikely) for the ACA to expand data.

Note 2.— The design of the ACA requires that the first header octet of a compressed address can never have the value [00] hexadecimal. Hence, the first header octet of a compressed address cannot be confused with the padding octet of an embedded address marker.

5.7.6.4.5 Compression Algorithm

5.7.6.4.5.1 General

5.7.6.4.5.1.1 The ACA shall perform compression by replacing ATN addresses or address prefixes identified in the input octet stream with compressed, encoded equivalents as defined below.

5.7.6.4.5.1.2 The format of a compressed address shall be as defined in 5.7.6.4.3.

5.7.6.4.5.1.3 Each compressed address shall be prefixed with a compressed address marker (5.7.6.4.4).

5.7.6.4.5.1.4 Any embedded address markers found in the input octet stream shall be padded with a null-value octet (5.7.6.4.4.3).

5.7.6.4.5.1.5 The overall logic flow of the ACA compression processing shall be as defined in 5.7.6.4.5.3.

5.7.6.4.5.2 Address Encoding Process

5.7.6.4.5.2.1 General

5.7.6.4.5.2.1.1 The process of encoding an ATN address or address prefix into the ACA compressed format (5.7.6.4.3) shall be performed using the sequence of steps defined in this paragraph.

5.7.6.4.5.2.1.2 The steps shall be performed in the order they are listed.

5.7.6.4.5.2.1.3 If any step of the encoding process fails, the ACA compression processing shall not consider the current input octets as an address and shall continue with the compression logic.

5.7.6.4.5.2.2 Encoding Address Length

5.7.6.4.5.2.2.1 Determination of the length in octets of an address to be compressed shall be performed as defined in 5.7.6.2.

5.7.6.4.5.2.2.2 If the address length is of type “normal”, the IDRP subfield in the first header octet shall be set to zero.

5.7.6.4.5.2.2.3 Otherwise, the IDRP subfield shall be set to one.

5.7.6.4.5.2.2.4 If the octet length of the address is 20 (indicating a full ATN address), the FP subfield in the first header octet shall be set to zero.

5.7.6.4.5.2.2.5 If the octet length of the address is less than 20 (indicating an address prefix), the FP subfield shall be set to one and the address length shall be encoded in the LEN/SEL subfield of the first header octet according to the table in 5.7.6.4.3.2.2.4.

5.7.6.4.5.2.2.6 If the address length is not found in the length table, the encoding process shall halt and the current input octet string shall not be treated as an ATN address.

5.7.6.4.5.2.3 Encoding the AFI and IDI Fields

5.7.6.4.5.2.3.1 No encoding shall be performed on the constant values of the address AFI and IDI fields.

5.7.6.4.5.2.3.2 These fields shall be omitted from the compressed address encoding.

5.7.6.4.5.2.4 Encoding the VER Field

5.7.6.4.5.2.4.1 If the VER value in the address to be compressed lies within the range [01]-[3f], the T/I subfield in the second header octet shall be set to zero and the F/M subfield in the second header octet shall be set to zero.

5.7.6.4.5.2.4.2 If the VER value lies within the range [01]-[07], then the low-order 3 bits of the VER value shall be stored in the CVER subfield of the first header octet.

5.7.6.4.5.2.4.3 If the VER value lies in the range [08]-[3f], then the CVER subfield shall be set to zero and the VER value octet shall be concatenated to the variable field of the encoded address.

5.7.6.4.5.2.4.4 If the VER value in the address to be compressed lies within the range [41]-[7f], the T/I subfield in the second header octet shall be set to zero and the F/M subfield in the second header octet shall be set to one.

5.7.6.4.5.2.4.5 If the VER value lies within the range [41]-[47], then the low-order 3 bits of the VER value shall be stored in the CVER subfield of the first header octet.

5.7.6.4.5.2.4.6 If the VER value lies in the range [48]-[7f], then the CVER subfield shall be set to zero and the VER value octet shall be concatenated to the variable field of the encoded address.

5.7.6.4.5.2.4.7 If the VER value in the address to be compressed lies within the range [81]-[bf], the T/I subfield in the second header octet shall be set to one and the F/M subfield in the second header octet shall be set to zero.

5.7.6.4.5.2.4.8 If the VER value lies within the range [81]-[87], then the low-order 3 bits of the VER value shall be stored in the CVER subfield of the first header octet.

5.7.6.4.5.2.4.9 If the VER value lies in the range [88]-[bf], then the CVER subfield shall be set to zero and the VER value octet shall be concatenated to the variable field of the encoded address.

5.7.6.4.5.2.4.10 If the VER value in the address to be compressed lies within the range [c1]-[ff], the T/I subfield in the second header octet shall be set to one and the F/M subfield in the second header octet shall be set to one.

5.7.6.4.5.2.4.11 If the VER value lies within the range [c1]-[c7], then the low-order 3 bits of the VER value shall be stored in the CVER subfield of the first header octet.

5.7.6.4.5.2.4.12 If the VER value lies in the range [c8]-[ff], then the CVER subfield shall be set to zero and the VER value octet shall be concatenated to the variable field of the encoded address.

5.7.6.4.5.2.4.13 If the VER value is either [00], [40], [80], or [c0], the encoding process shall halt and the current input octet string shall not be treated as an ATN address.

5.7.6.4.5.2.5 Encoding the ADM Field

5.7.6.4.5.2.5.1 If the three octets of the ADM field in the address to be compressed do not follow the rules for Identifier Syntax, the ADMF subfield in the second header octet shall be set to zero and the three octets of the ADM field value shall be concatenated to the compressed ADM of the encoded address.

5.7.6.4.5.2.5.2 If the ADM field value does follow the Identifier Syntax rules, the ADMF subfield shall be set to one and the two-octet compressed ADM value (5.7.6.4.3.3) shall be concatenated to the compressed ADM of the encoded address.

5.7.6.4.5.2.6 Encoding the RDF Field

5.7.6.4.5.2.6.1 If the address length indicates an address prefix whose length is less than or equal to 7, no RDF field value shall be encoded and the encoding process shall halt.

5.7.6.4.5.2.6.2 If the RDF value in the address to be compressed is not [00], the encoding process shall halt and the current input octet string shall not be treated as an ATN address.

5.7.6.4.5.2.7 Encoding the ARS Field

5.7.6.4.5.2.7.1 If the address length indicates an address prefix whose length is less than or equal to 8, no ARS field value shall be encoded and the encoding process shall halt.

5.7.6.4.5.2.7.2 If the ARS value of the address to be compressed has the default value ([000001] hexadecimal), the ARSD subfield in the second header octet shall be set to one.

5.7.6.4.5.2.7.3 If the ARS value of the address to be compressed is not default, the ARSD subfield shall be set to zero and the three octets of the ARS value shall be concatenated to the variable field data of the encoded address.

5.7.6.4.5.2.8 Encoding the LOC Field

5.7.6.4.5.2.8.1 If the address length indicates an address prefix whose length is less than or equal to 11, no LOC field value shall be encoded and the encoding process shall halt.

5.7.6.4.5.2.8.2 If the LOC value of the address to be compressed has the default value ([0001] hexadecimal), the LOCD subfield in the second header octet shall be set to one.

5.7.6.4.5.2.8.3 If the LOC value of the address to be compressed is not default, the LOCD subfield shall be set to zero and the two octets of the LOC value shall be concatenated to the variable field data of the encoded address.

5.7.6.4.5.2.9 Encoding the SYS Field

5.7.6.4.5.2.9.1 If the address length indicates an address prefix whose length is less than or equal to 13, no SYS field value shall be encoded and the encoding process shall halt.

5.7.6.4.5.2.9.2 If the high-order (6th) octet of the SYS field of the address to be compressed has a nonzero value, the SYS6 subfield in the second header octet shall be set to zero and the value of the SYS field octet shall be concatenated to the variable field data of the encoded address. Otherwise, the SYS6 subfield shall be set to one.

5.7.6.4.5.2.9.3 If the second to high-order (5th) octet of the SYS field of the address to be compressed has a nonzero value, the SYS5 subfield in the second header octet shall be set to zero and the value of the SYS field octet shall be concatenated to the variable field data of the encoded address.

5.7.6.4.5.2.9.4 Otherwise, the SYS5 subfield shall be set to one.

5.7.6.4.5.2.9.5 If the third to high-order (4th) octet of the SYS field of the address to be compressed has a nonzero value, the SYS4 subfield in the second header octet shall be set to zero and the value of the SYS field octet shall be concatenated to the variable field data of the encoded address.

5.7.6.4.5.2.9.6 Otherwise, the SYS4 subfield shall be set to one.

5.7.6.4.5.2.9.7 The three remaining octets of the SYS field shall be concatenated to the variable field data of the encoded address.

5.7.6.4.5.2.10 Encoding the SEL Field

5.7.6.4.5.2.10.1 If the address length indicates an address prefix whose length is less than or equal to 19, no SEL field value shall be encoded and the encoding process shall halt.

5.7.6.4.5.2.10.2 Since the address length indicates a full ATN address, the FP subfield in the first header octet shall be set to zero.

5.7.6.4.5.2.10.3 The SEL value shall be encoded into the LEN/SEL subfield in the first header octet according to the table in 5.7.6.4.3.2.2.4.

5.7.6.4.5.2.10.4 If the SEL value is not one of the table entries, the LEN/SEL subfield shall be set to 7 and the SEL value octet shall be concatenated to the variable field data of the encoded address.

5.7.6.4.5.3 Compression Logic Flow

5.7.6.4.5.3.1 The ACA compression logic shall process octets sequentially from the uncompressed data input stream.

5.7.6.4.5.3.2 For each input octet, a test shall be performed to determine if the current octet and the subsequent octets form an ATN address or address prefix.

5.7.6.4.5.3.3 If they do form an ATN address, the ACA shall attempt to encode the address into the compressed address format (5.7.6.4.3) as defined in the steps of 5.7.6.4.5.2.

5.7.6.4.5.3.4 If the encoding process is successful, a compressed address marker (5.7.6.4.4) shall be output to the compressed octet stream followed by the compressed address octets.

5.7.6.4.5.3.5 The compression processing shall then continue with the next uncompressed data octet not a part of the address just processed.

5.7.6.4.5.3.6 If the encoding process fails, or if the current octet does not begin an ATN address, the ACA processing shall check at the current uncompressed octet position in the input data stream for an embedded address marker (5.7.6.4.4.3).

5.7.6.4.5.3.7 If an embedded address marker is found, the ACA shall copy the address marker octets to the compressed output octet stream. A padding zero-valued octet shall be output as well as the address marker.

5.7.6.4.5.3.8 The compression processing shall then continue with the next uncompressed data octet not a part of the embedded address marker.

5.7.6.4.5.3.9 If neither an ATN address or embedded address mark is found, the ACA shall copy the current uncompressed input octet to the compressed output octet stream and shall continue processing with the next sequential input octet.

Note.— Since the ACA compression logic may not recognize the appearance of an ATN address or prefix in the data stream until after the uncompressed length octet has been processed (the length octet precedes the fixed-value ATN AFI and IDI fields that distinguish an ATN address), the ACA compression process will need to be able to recall the value of the previous input octet during compression processing. Hence, a one-octet “backup” may be necessary in the implementation of the ACA compression logic.

5.7.6.4.6 Decompression Algorithm

5.7.6.4.6.1 General

5.7.6.4.6.1.1 The ACA shall perform decompression by replacing compressed ATN addresses or address prefixes in the ACA compressed format (5.7.6.4.3) with their expanded equivalent as defined below. Address markers and padding octets shall be removed from the data stream during ACA decompression processing.

5.7.6.4.6.1.2 The overall logic flow of the ACA decompression processing shall be as defined in 5.7.6.4.5.3.

5.7.6.4.6.2 Address Decoding Process

5.7.6.4.6.2.1 General

5.7.6.4.6.2.1.1 The process of decoding a compressed ATN address or address prefix from the ACA compressed format (5.7.6.4.3) shall be performed using the sequence of steps defined in the following paragraphs.

5.7.6.4.6.2.1.2 The steps shall be performed in the order listed below. The expanded address or prefix shall include the decoded address length octet and the decoded 7-20 address octets.

5.7.6.4.6.2.2 Decoding Address Length

5.7.6.4.6.2.2.1 If the FP subfield in the first header octet is zero, the octet length of the compressed address shall be set to 20 (a full ATN address).

5.7.6.4.6.2.2.2 Otherwise, the octet length of the compressed address prefix shall be decoded from the LEN/SEL subfield in the first header octet according to the table in 5.7.6.4.3.2.2.4.

5.7.6.4.6.2.2.3 The address octet length shall be used in the further decoding process steps.

5.7.6.4.6.2.2.4 If the IDRP subfield in the first header octet is zero, the output address length shall be the address octet length.

5.7.6.4.6.2.2.5 Otherwise, the output address length shall be 8 times the address octet length.

Note.— The address octet length is an internal variable used in the decoding process. The output length prefixed to the expanded address after the decoding process is completed is either the same as the octet length (normal case) or 8 times the octet length (IDRP case, length in bits).

5.7.6.4.6.2.3 Decoding the AFI and IDI Fields

5.7.6.4.6.2.3.1 The AFI field of the decoded address shall be set to its constant value of [47] hexadecimal.

5.7.6.4.6.2.3.2 The IDI field of the decoded address shall be set to its constant value of [0027] hexadecimal.

5.7.6.4.6.2.4 Decoding the VER Fields

5.7.6.4.6.2.4.1 If the CVER subfield in the first header octet is zero, the VER octet shall be extracted from the next octet in the variable field of the compressed address.

5.7.6.4.6.2.4.2 If the CVER subfield is non-zero, then the VER field value in the expanded address shall be computed as follows:

- a) If the T/I subfield in the second header octet is zero and the F/M subfield in the second header octet is zero, then the VER field value shall be set to the CVER value.
- b) If the T/I subfield in the second header octet is zero and the F/M subfield in the second header octet is one, then the VER field value shall be set to the CVER value plus 64.
- c) If the T/I subfield in the second header octet is one and the F/M subfield in the second header octet is zero, then the VER field value shall be set to the CVER value plus 128.
- d) If the T/I subfield in the second header octet is one and the F/M subfield in the second header octet is one, then the VER field value shall be set to the CVER value plus 192.

5.7.6.4.6.2.5 Decoding the ADM Fields

5.7.6.4.6.2.5.1 If the ADMF subfield in the second header octet is set to zero, the three octets of the ADM field shall be extracted from the next three octets in the variable field data.

5.7.6.4.6.2.5.2 Otherwise, the ADM field value shall be decoded from the compressed ADM which is extracted from the next two octets in the variable field data of the compressed address.

5.7.6.4.6.2.5.3 The decoding of the compressed ADM value shall be performed as defined in 5.4.2.3.7.

5.7.6.4.6.2.6 Decoding the RDF Fields

5.7.6.4.6.2.6.1 The RDF field in the expanded address shall be set to zero.

5.7.6.4.6.2.7 Decoding the ARS Fields

5.7.6.4.6.2.7.1 If the address length indicates an address prefix whose length is less than or equal to 8, no ARS field value shall be decoded and the decoding process shall halt.

5.7.6.4.6.2.7.2 If the ARSD subfield in the second header octet of the compressed address is set to one, the expanded ARS field shall be set to the default value ([000001] hexadecimal).

5.7.6.4.6.2.7.3 Otherwise, the expanded ARS field value shall be extracted from the next three octets in the variable field data of the compressed address.

5.7.6.4.6.2.8 Decoding the LOC Fields

5.7.6.4.6.2.8.1 If the address length indicates an address prefix whose length is less than or equal to 11, no LOC field value shall be decoded and the decoding process shall halt.

5.7.6.4.6.2.8.2 If the LOCD subfield in the second header octet of the compressed address is set to one, the expanded LOC field shall be set to the default value ([0001] hexadecimal).

5.7.6.4.6.2.8.3 Otherwise, the expanded LOC field value shall be extracted from the next two octets in the variable field data of the compressed address.

5.7.6.4.6.2.9 Decoding the SYS Fields

5.7.6.4.6.2.9.1 If the address length indicates an address prefix whose length is less than or equal to 13, no SYS field value shall be decoded and the decoding process shall halt.

5.7.6.4.6.2.9.2 If the SYS6 subfield in the second header octet has the value one, the high-order (6th) octet of the expanded SYS field shall be extracted from the next octet in the variable data field of the compressed address.

5.7.6.4.6.2.9.3 Otherwise, the high-order (6th) octet of the expanded SYS field shall be set to zero.

5.7.6.4.6.2.9.4 If the SYS5 subfield in the second header octet has the value one, the second to high-order (5th) octet of the expanded SYS field shall be extracted from the next octet in the variable data field of the compressed address. Otherwise, the second to high-order (5th) octet of the expanded SYS field shall be set to zero.

5.7.6.4.6.2.9.5 If the SYS4 subfield in the second header octet has the value one, the third to high-order (4th) octet of the expanded SYS field shall be extracted from the next octet in the variable data field of the compressed address.

5.7.6.4.6.2.9.6 Otherwise, the third to high-order (4th) octet of the expanded SYS field shall be set to zero.

5.7.6.4.6.2.9.7 The remaining three octets of the expanded SYS field shall be extracted from the next three octets in the variable data field of the compressed address.

5.7.6.4.6.2.10 Decoding the SEL Fields

5.7.6.4.6.2.10.1 If the address length indicates an address prefix whose length is less than or equal to 19, no SEL field value shall be decoded and the decoding process shall halt.

5.7.6.4.6.2.10.2 If the FP subfield in the first header octet has the value zero (indicating a full ATN address), then the value of the SEL field shall be decoded from the LEN/SEL subfield in the first header octet.

5.7.6.4.6.2.10.3 If the value of the LEN/SEL subfield lies in the range 1-2 the SEL value shall be decoded using the SEL encoding table in 5.7.6.4.3.2.2.4.

5.7.6.4.6.2.10.4 If the LEN/SEL subfield encoding has the value 7, the SEL field value shall be extracted from the next octet in the variable data field of the compressed address.

Note.— Only a full ATN address (not a prefix) includes a SEL field.

5.7.6.4.6.3 Decompression Logic Flow

5.7.6.4.6.3.1 The ACA decompression logic shall process octets sequentially from the compressed data input stream.

5.7.6.4.6.3.2 If the octet at the current input position and the next octet do not form a compressed address marker (5.7.6.4.4), the current input octet shall be copied to the decompressed output octet stream and decompression processing shall continue with the next input octet.

5.7.6.4.6.3.3 When a compressed address marker is found in the input octet stream, the decompression processing shall examine the value of the next octet beyond the address marker.

5.7.6.4.6.3.4 If the value of this octet is zero (indicating an embedded address mark (5.7.6.4.4.3), the compressed address marker octets shall be copied to the decompressed output octet stream and the zero-value octet shall be dropped from the output stream.

5.7.6.4.6.3.5 If the value is nonzero (indicating a compressed ATN address), the compressed address shall be decoded according to 5.7.6.4.6.2.

5.7.6.4.6.3.6 The decoded address octets shall be copied to the decompressed octet output stream and decompression processing shall continue with the next input octet beyond those that formed the compressed ATN address.

5.7.6.4.6.3.7 The compressed address marker octets shall not be copied to the output.

5.7.6.5 Stream Mode Compression Using Deflate

Note 1.— The Deflate algorithm was originally specified in IETF RFC 1951 and through example ‘C’ code available from the algorithm’s authors.

Note 2.— The Deflate algorithm is a combination of two public domain and well known data compression algorithms. These are the LZ77 algorithm (Lempel-Ziv 1977) and Huffman Codes. LZ77 removes redundancy in the data stream by replacing re-occurring strings by backward references to previous occurrences of such strings. Huffman Codes are variable length symbols that are used to compress strings of fixed length symbols. The Huffman Codes are chosen such that frequently occurring symbols are replaced by shorter bitstrings whilst rarely occurring symbols are replaced by longer bitstrings. They are also chosen such that no code is the prefix of another code in the same set of Huffman Codes. In Deflate, the uncompressed data is first compressed using LZ77 and the result of this compression stage is further

compressed using a set of standard Huffman Codes in order to compress both the literal value of strings for which no backward reference can be given, and the backward references themselves.

Note 3.— Deflate further optimises the data compression by monitoring the stream of uncompressed data and dynamically generating a set of more optimal Huffman Codes. These can be communicated to the receiver at any time and used to improve the compression ratio.

Note 4.— The Deflate specification also permits the compressor, when it detects an uncompressible string, to send that string as plain text.

Note 5.— The Deflate algorithm has significant memory requirements when providing high compression efficiency. This extensive memory demand per compressed data stream may limit the number of virtual circuits which can be simultaneously supported by a given ATN Router implementation over an air/ground adjacency. Consequently, ATN operators may choose to not support data stream compression when the demand for simultaneous air/ground connections exceeds the available memory resources.

5.7.6.5.1 Service Description

Note.— The Deflate encoder operates on NPDUs submitted via the SN-Service and after compression by the LREF function if used. The Deflate decoder operates on data packets received from the subnetwork service provided by ISO/IEC 8208. The decoded NPDUs may then be further decompressed by the LREF compression procedures, if in use, or passed to the SN-Service user. The positioning of the Deflate encoder and decoder is illustrated in Figure 5.7-8.

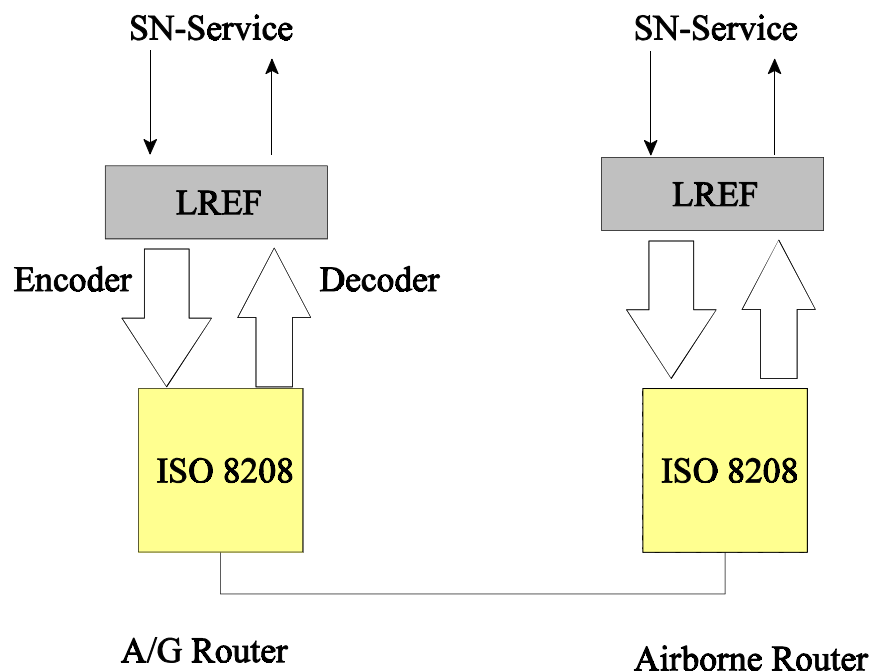


Figure 5.7-8. Relationship of the Deflate Encoder and Decoder to ISO/IEC 8208 and LREF Functions

5.7.6.5.1.1 When the use of the Deflate algorithm has been proposed in the Call Request User Data and either implicitly accepted by Call Acceptance in the absence of the Fast Select procedures, or explicitly

accepted in the Call Accept when Fast Select is in use, then user data on all subsequent data packets shall be encoded using this algorithm.

Note.— ISO/IEC 8208 packets other than data packets may also contain user data. The above requirement excludes the encoding of user data on control packets as they may be delivered out of sequence.

5.7.6.5.2 Encoded Packet Format

5.7.6.5.2.1 Each NPDU shall be encoded into the compressed representation shown in Figure 5.7-9.

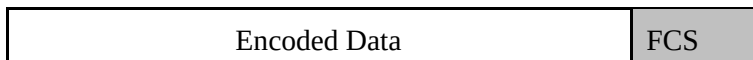


Figure 5.7-9 Compressed Packet Format

5.7.6.5.2.2 The compressed packet format shall comprise:

- a) The Encoded Data, and
- b) A two-octet Frame Check Sum (FCS).

Note.— The length of the encoded data need not be explicitly specified as the encoded block is delimited by ISO/IEC 8208.

5.7.6.5.2.3 The sender shall ensure that the encoded representation of an NPDU is complete, i.e. that the receiver can recover the original NPDU without requiring information contained in any subsequent packets.

Note.— In IETF RFC 1951, an encoded data stream may comprise an arbitrary number of compressed blocks. This is also true for this specification. The purpose of the Deflate Data Blocks is to delimit the scope of uncompressed data strings, strings compressed using the standard set of Huffman Codes, and those compressed using dynamically determined Huffman Codes. The compressor may decide to change between either one of these strategies at any time and not just at an NPDU boundary. A compressed NPDU will always start at a Deflate Data Block boundary and end at the end of a Deflate Data Block.

5.7.6.5.2.4 The encoded representation of the NPDUs shall be a data stream that is subdivided into a number of bit-aligned blocks of arbitrary length.

5.7.6.5.2.5 Each such block shall be in the format shown in Figure 5.7-10.

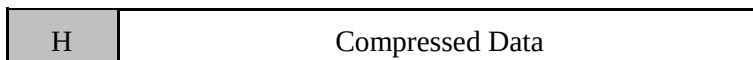


Figure 5.7-10 Format of Deflate Data Blocks

5.7.6.5.2.6 Each Deflate Data Block shall comprise:

- a) A 3-bit header (H), and

- b) A stream of self-delimited compressed data.

5.7.6.5.2.7 The first bit of the 3-bit header (i.e. the first bit transmitted) shall always be set to zero.

Note.— In IETF RFC 1951, setting the first bit to one indicates that it is the last block in an encoded data stream. This semantic is not required by this specification, as the end of a subnetwork connection fulfils this requirement.

5.7.6.5.2.8 The remaining two bits of the header shall be used to indicate the compression type according to Table 5.7-13.

Table 5.7-13 Compression Type Identifiers (bits shown in transmission order)

Encoding	Compression Type
00	no compression
01	compressed with fixed Huffman codes
10	compressed with dynamically determined Huffman Codes
11	reserved

5.7.6.5.3 Uncompressed Deflate Data Blocks

5.7.6.5.3.1 When the encoder determines that no benefit can be derived by data compression of a given string, then that string shall be sent uncompressed.

5.7.6.5.3.2 The 3-bit header shall be right-padded with zeroes to the next octet boundary, and the remainder of the encoded data shall be formatted as shown in Figure 5.7-11.



Figure 5.7-11 Format of Uncompressed Deflate Data Blocks

5.7.6.5.3.3 An Uncompressed Deflate Data Block shall comprise:

- An unsigned 16-bit length indicator (LEN), giving the number of octets of literal data in the block;
- The ones complement of the 16-bit length indicator (NLEN);
- The Literal Data.

5.7.6.5.3.4 The two length fields (LEN and NLEN) shall be encoded and sent least significant octet first.

5.7.6.5.3.5 The literal data shall be encoded in the same byte order as encountered in the uncompressed data stream.

Note 1.— The procedures by which the encoder determines that there is no benefit in compressing an NPDU are outside of the scope of this specification.

Note 2.— Even though the string is not compressed, this does not prevent the data in this block being referenced as part of the data stream by a subsequent LZ77-encoded NPDU.

5.7.6.5.4 Compressed Deflate Data Blocks using Fixed Huffman Codes

5.7.6.5.4.1 General

Note.— Encoded data blocks in the Deflate format consist of sequences of symbols drawn from three conceptually distinct alphabets: either literal bytes, the alphabet of byte values (0..255), or <length, backward distance> pairs, where the length is drawn from (3..258) and the distance is drawn from (1..32,768). The literal and length alphabets are merged into a single alphabet (0..285), where values 0..255 represent literal bytes, and values 257..285 represent length codes (possibly in conjunction with extra bits following the symbol code). The value 256 indicates end-of-block and the block is hence self-delimiting without requiring an explicit length indicator.

5.7.6.5.4.1.1 A compressed NPDU shall be sent as a bit stream of bit-aligned symbols (the Huffman Codes representing literal values or length distance pairs), starting with the first bit transmitted after the 3-bit header.

5.7.6.5.4.1.2 The Huffman Codes used to encode the literal/length code in the LZ77 compressed data stream shall be as specified in Table 5.7-14.

Note.— Although Table 5.7-14 includes values 286 and 287, these are not used by the compression algorithm and are included only for completeness of the set of valid Huffman Codes.

Table 5.7-14 Huffman Codes Used for Deflate

Value	Code Length (Bits)	Huffman Code
0 - 143	8	00110000 through 10111111
144 - 255	9	110010000 through 111111111
256 - 279	7	0000000 through 0010111
280 - 287	8	11000000 through 11000111

5.7.6.5.4.1.3 Huffman encoded values 0 to 255 inclusive shall represent literal values, i.e. the single octet values of a literal string.

Note.— The term “Huffman Encoded Value” is used to identify a symbol value that is represented by a Huffman code taken from Table 5.7-14. For example, the “Huffman Encoded Value 145” is encoded as a 9-bit bit string “110010001”.

5.7.6.5.4.1.4 The Huffman Encoded value of 256 shall be used to indicate end-of-block and shall be appended at the end of each intermediate compressed Deflate Data Block.

Note.— An octet containing this value may be removed from a Deflate Data Block, if this data block is the last one in an NPDU. In this case the block is delimited by the NPDU boundary and by not using this value, the size of the compressed data stream is reduced.

5.7.6.5.4.1.5 The Huffman codes shall be encoded (packed) into the compressed data block, most significant bit first.

5.7.6.5.4.2 Length/Distance Codes

5.7.6.5.4.2.1 Length Codes

5.7.6.5.4.2.1.1 Huffman-encoded values in the range 257 to 285 shall represent a length code and shall always be followed by an associated distance code.

5.7.6.5.4.2.1.2 Each length code shall represent a particular string length, as specified in Table 5.7-15.

5.7.6.5.4.2.2 Extra Bits for Length Codes

5.7.6.5.4.2.2.1 Where a non-zero Extra Bit is specified for a given code, then a range of length values is represented by the length indicator, and the encoded representation of the length indicator shall be followed by exactly that number of additional bits.

5.7.6.5.4.2.2.2 The Extra Bits shall be interpreted as an integer stored with the most significant bit first.

Note.— For example, bits 1110 represent the value 14.

5.7.6.5.4.2.2.3 The value of the extra bits shall be added to the first length value in the range identified by such Length Code in order to determine the actual string length.

Note 1.— For example, Length Code 277 is followed by four extra bits. If these are 1110 then the actual string length indicated is 81.

Note 2.— Extra bits are not encoded as Huffman Codes.

Table 5.7-15 String Length Code Values

Code	Extra Bits	Length(s)	Code	Extra Bits	Lengths	Code	Extra Bits	Length(s)
257	0	3	267	1	15,16	277	4	67-82
258	0	4	268	1	17,18	278	4	83-98
259	0	5	269	2	19-22	279	4	99-114
260	0	6	270	2	23-26	280	4	115-130
261	0	7	271	2	27-30	281	5	131-162
262	0	8	272	2	31-34	282	5	163-194
263	0	9	273	3	35-42	283	5	195-226
264	0	10	274	3	43-50	284	5	227-257
265	1	11,12	275	3	51-58	285	0	258
266	1	13,14	276	3	59-66			

5.7.6.5.4.2.3 Distance Codes

5.7.6.5.4.2.3.1 Each length code in the encoded data stream shall be followed by a Huffman-encoded distance code according to Table 5.7-16.

5.7.6.5.4.2.3.2 In this block format, the Huffman Codes for the distance codes shall be the 5-bit value of the distance code completed with leading zero-bits.

Note.— As this implies, the distance codes are assumed to each have the same probability of occurrence and hence there is no possibility of compression using Huffman Codes.

5.7.6.5.4.2.4 Extra Bits for Distance Codes

5.7.6.5.4.2.4.1 Where a non-zero Extra Bit is specified for a given distance code, then a range of distances is represented by the distance code, and the encoded representation of the length indicator shall be followed by exactly that number of additional bits.

5.7.6.5.4.2.4.2 The Extra Bits shall be interpreted as an integer stored with the most significant bit first.

Note.— For example, bits 1110 represent the value 14.

5.7.6.5.4.2.4.3 The value of the extra bits shall be added to the first distance value in the range identified by such a distance code in order to determine the actual string length.

5.7.6.5.4.2.5 Semantic

5.7.6.5.4.2.5.1 The semantic of the distance value shall be the string (of length given by the length indicator) in the previously received data, at exactly the number of octets given by the distance value from the current position.

Note 1.— For example, the most recently received octet has a distance of one from the current position.

Note 2.— It is therefore possible under this specification to refer to a previously occurring string within the previous 32KB of data transmitted.

5.7.6.5.4.2.5.2 A backward reference shall not refer to a string on any other subnetwork connection, or transmitted before a network reset has been performed.

Note 1.— A string reference may refer to a string in a previous block; i.e., the backward distance may cross one or more block boundaries. However a distance cannot refer past the beginning of the subnetwork connection, or since the most recent network service reset due to the fact that the receiving user may not have received those blocks transmitted immediately prior to a reset.

Note 2.— The referenced string may overlap the current position; for example, if the last 2 bytes decoded have values X and Y, a string reference with <length = 5, distance = 2> adds X,Y,X,Y,X to the output stream.

Table 5.7-16 Distance Codes

Code	Extra Bits	Distance	Code	Extra Bits	Distance	Code	Extra Bits	Distance
0	0	1	10	4	33-48	20	9	1025-1536
1	0	2	11	4	49-64	21	9	1537-2048
2	0	3	12	5	65-96	22	10	2049-3072
3	0	4	13	5	97-128	23	10	3073-4096
4	1	5,6	14	6	129-192	24	11	4097-6144
5	1	7,8	15	6	193-256	25	11	6145-8192
6	2	9-12	16	7	257-384	26	12	8193-12288
7	2	13-16	17	7	385-512	27	12	12289-16384
8	3	17-24	18	8	513-768	28	13	16385-24576
9	3	25-32	19	8	769-1024	29	13	24577-32768

5.7.6.5.5 Compressed Deflate Data Blocks using Dynamically Determined Huffman Codes

5.7.6.5.5.1 General

Note 1.— The fixed set of Huffman Codes represent an initial “guess” as to the entropy of the original data stream and hence what are the optimal Huffman Codes. However, it is likely that analysis of an actual data stream will reveal a more appropriate set. This specification allows for this by providing a means to communicate a set of dynamically determined Huffman Code Tables from compressor to decompressor and to identify the scope of applicability for those codes. This is achieved through the Deflate Data Block format specified in this section. The data block includes a new set of Huffman Code Tables at the beginning of the block and the remainder of the block comprises a compressed LZ77 data stream, compressed using these Huffman Code Tables.

Note 2.— In order to avoid the overhead of exchanging the actual Huffman Code Tables, the Huffman Codes are required to comply with a set of rules that permits a Huffman Code Table to be generated from knowledge of the code lengths and the encoded alphabet only. As the alphabet is known by the decompressor a priori, only the code lengths have to be communicated.

Note 3.— A further level of compression is achieved by encoding the lists of code lengths as Huffman Codes. The Huffman Codes for the code lengths are themselves communicated at the start of this Deflate Data Block format, by communicating their code lengths only.

Note 4.— The mechanism by which the compressor decides to make use of dynamically determined Huffman Codes is outside of the scope of this specification.

5.7.6.5.5.1.1 The Huffman codes used for each alphabet in the Deflate format shall obey the following rules:

- a) All codes of a given bit length have lexicographically consecutive values, in the same order as the symbols they represent; and
- b) Shorter codes lexicographically precede longer codes.

5.7.6.5.5.1.2 The sequences of code length shall themselves be compressed using a Huffman code and the alphabet for code lengths specified in Table 5.7-17.

Table 5.7-17 Alphabet for Code Lengths

Code	Semantic
0 .. 15	Represent code length of 0 to 15
16	Copy the previous code length 3 to 6 times; the next 2 bits indicate the repeat length (0 = 3, ... 3 = 6)
17	Repeat a code length of 0 for 3 to 10 times; the next 3 bits indicate the repeat length
18	Repeat a code length of 0 for 11 to 138 times; the next 7 bits indicate the repeat length

Note.— For example, codes 8, 16(+ binary 11), 16(+ binary 10) will expand to 12 code length of 8.

5.7.6.5.5.1.3 A code length of 0 shall indicate that the corresponding symbol in the literal/length or distance alphabet will not occur in the block and does not participate in the Huffman code construction algorithm.

5.7.6.5.5.2 Block Format

5.7.6.5.5.2.1 The format of a Deflate Data Block using Dynamically Determined Huffman Codes shall comprise the following bit-aligned fields starting immediately after the 3-bit header, and encoded consecutively:

- a) The 5-bit HLIT field, set to (number of Literal/Length codes - 257);

Note.— The number of Literal/Length codes is in the range 257 to 286.

- b) The 5-bit HDIST field, set to (number of Distance codes - 1);

Note.— The number of Distance codes is in the range 1 to 32.

- c) The 4-bit HCLEN field, set to (number of Code Length codes - 4);

Note.— The number of Code Length codes is in the range 4 to 19.

- d) $(HCLEN + 4) \times 3$ bits: code lengths for the code length alphabet given in Table 5.7-17, in the order: 16, 17, 18, 0, 8, 7, 9, 6, 10, 5, 11, 4, 12, 3, 13, 2, 14, 1, 15;

Note.— The code lengths are interpreted as 3-bit integers (0-7); as above, a code length of 0 means the corresponding symbol (literal/length or distance code length) is not used.

- e) $(HLIT + 257)$ code lengths for the literal/length alphabet, encoded using the code length Huffman code

- f) $(HDIST + 1)$ code lengths for the distance alphabet, encoded using the code length Huffman code

- g) The actual compressed data of the block, encoded using the literal/length and distance Huffman codes defined in the first part of this block

- h) The literal/length symbol 256 (end of data), encoded using the literal/length Huffman code.

Note.— The code-length repeat codes can cross from $HLIT + 257$ to the $HDIST + 1$ code lengths. In other words, all code lengths form a single sequence of $HLIT + HDIST + 258$ values.

5.7.6.5.5.3 Decoding of Dynamically Determined Huffman Codes

Note.—The following algorithm generates the Huffman Codes from the encoded bit-length codes as integers, intended to be read from most- to least-significant bit. A version of this algorithm expressed in 'C' code may be found in IETF RFC 1951.

5.7.6.5.3.1 Dynamically determined Huffman codes shall be decoded as follows:

- 1) Count the number of codes for each code length.
- 2) Find the numerical value of the first code for each code length, by applying the rule that no Huffman Code in the same table can be the prefix of another. For the smallest code length this is zero. For each subsequent code length, this is determined by identifying the next unallocated code for the preceding code length (by adding the number of codes to the first code) and then representing the result as a binary number, and right-padding the number with zero-bits so that the number has the same number of bits as required by the code length.
- 3) Assign numerical values to all codes, using consecutive values for all codes of the same length with the base values determined at step 2. Codes that are never used (which have a bit length of zero) must not be assigned a value.

Note.— For example, consider the alphabet ABCDEFGH with code lengths defined to be (3,3,3,3,3,2,4,4). Applying the above algorithm would generate the following Huffman Codes for each member of the alphabet:

<i>Symbol</i>	<i>Length</i>	<i>Code</i>
<i>A</i>	<i>3</i>	<i>010</i>
<i>B</i>	<i>3</i>	<i>011</i>
<i>C</i>	<i>3</i>	<i>100</i>
<i>D</i>	<i>3</i>	<i>101</i>
<i>E</i>	<i>3</i>	<i>110</i>
<i>F</i>	<i>2</i>	<i>00</i>
<i>G</i>	<i>4</i>	<i>1110</i>
<i>H</i>	<i>4</i>	<i>1111</i>

5.7.6.5.6 Frame Check Sum (FCS)

5.7.6.5.6.1 A two-octet, octet-aligned, frame checksum shall be appended to the end of each encoded packet.

5.7.6.5.6.2 The frame checksum shall be computed according to the same procedures as specified in ISO/IEC 8073 for computation of the transport protocol class 4 checksum.

5.7.6.5.6.3 The two octets of the frame checksum shall be the two partial sums C0 and C1 as specified in ISO/IEC 8073 Annex D.

5.7.6.5.6.4 The value of C0 shall be the first octet of the frame checksum parameter and the value of C1 shall be the second octet.

5.7.6.5.6.5 The checksum shall be computed on the NPDU prior to application of the Deflate data compression procedure, i.e. it is a checksum on the uncompressed NPDU.

Note.— The Frame Check Sum may be used by the decompression procedure to verify correct decompression of the NPDU.

5.7.6.5.7 Compression Procedure

5.7.6.5.7.1 General

5.7.6.5.7.1.1 Each NPDU received from the SN-Service User, possibly after compression by the LREF algorithm, shall be encoded into a single compressed data block in the format given by Figure 5.7-9 and specified in section 5.7.6.5.2.

5.7.6.5.7.1.2 The resulting data block shall be a complete encoded representation of the NPDU.

5.7.6.5.7.1.3 **Recommendation.**— *An implementation should use the full 32KB range of distance values permitted by the compressed data format.*

Note 1.— This permits an implementation of the compressor to autonomously limit the size of the backwards window used to compress data in order to optimise the use of memory resources. However, the result will be a poorer compression ratio. On the other hand, the decompressor must always be able to accept any valid distance value, i.e. must maintain a 32KB buffer.

Note 2.— The actual procedure by which an implementation locates matches for strings in previously sent data, or even the length of the strings it looks for, is out of the scope of this specification.

5.7.6.5.7.2 NPDU Encoding

5.7.6.5.7.2.1 The NPDU shall be encoded in the same sequence in which it would have been transmitted if it had not been compressed.

5.7.6.5.7.2.2 Octet sequences for which no preceding match is found shall be encoded as literal values using their corresponding Huffman codes (i.e. Huffman Codes representing values in the range 0..255).

5.7.6.5.7.2.3 Octet sequences for which a match has been found within the last 32KB of encoded data shall be encoded as length/distance pairs.

5.7.6.5.7.2.4 The length of the octet string shall be encoded first, where necessary followed by the appropriate extra bits needed to fully define the length value.

5.7.6.5.7.2.5 The distance to the duplicate string shall similarly be encoded using the Huffman Code specified in Table 5.7-15 for the required distance, where necessary also followed by the appropriate extra bits needed to fully define the distance.

5.7.6.5.7.2.6 The Huffman Codes used shall be defined by the type of Deflate Data Block (i.e. using the set of Fixed Huffman codes or a dynamically determined set).

5.7.6.5.7.2.7 NPDUs shall be compressed and passed to the ISO/IEC 8208 subnetwork in exactly the same order that they were given to the Deflate compression function by the SN-Service User.

5.7.6.5.7.2.8 When all octets in the NPDU have been encoded, the bit stream shall be padded with zero-bits until the next octet boundary is reached.

5.7.6.5.7.2.9 The Frame Check Sum (FCS) shall then be appended to the compressed block.

Note.— The FCS is encoded as its binary value. It is not subject to Huffman Encoding.

5.7.6.5.8 Decompression Procedures

5.7.6.5.8.1 General

5.7.6.5.8.1.1 NPDUs shall be decompressed in exactly the same order that they have been received from the ISO/IEC 8208 subnetwork.

5.7.6.5.8.1.2 Each data packet received from an ISO/IEC 8208 subnetwork shall be assumed to be in the format given by Figure 5.7-9, and thus comprises one or more Deflate Data Blocks.

5.7.6.5.8.2 Compressed Deflate Data Block

5.7.6.5.8.2.1 Each compressed Deflate Data Block shall be interpreted as a sequence of Huffman-encoded symbols.

5.7.6.5.8.2.2 Huffman-encoded values in the range 0..255 shall be taken as literal octet values and appended to the NPDU that is being decompressed in the order that they are found.

5.7.6.5.8.2.3 The Huffman-encoded value 256 shall be taken as end-of-block indication and not appended to the NPDU that is being decompressed.

5.7.6.5.8.2.4 Huffman-encoded values in the range 257..285 shall be taken as length indicators and as introducing a length/distance pair.

5.7.6.5.8.2.5 The length and distance values shall be decoded and the referenced string shall be appended to NPDU that is being decompressed.

5.7.6.5.8.3 Uncompressed Deflate Data Block

5.7.6.5.8.3.1 Octets from uncompressed Deflate Data Blocks shall be appended to the NPDU in the order in which they are encoded.

5.7.6.5.8.4 FCS Verification

5.7.6.5.8.4.1 The Frame Check Sum for the uncompressed NPDU shall be the last two octets of the received packet and shall be verified for all received NPDUs.

5.7.6.5.8.4.2 If this verification check fails, then the NPDU shall be discarded and a Network Reset initiated on the ISO/IEC 8208 subnetwork connection.

Note.— This network reset will be indicated to the receiving peer entity as a DTE-originated reset.

5.7.6.5.8.4.3 In this case, the history compression window shall be reset to the initial state.

Note.— As the sender is not permitted to reference strings prior to a network reset, this procedure ensures that a backwards reference cannot be made into a corrupt NPDU.

5.7.6.5.8.4.4 **Recommendation.**— *The error should be notified to System Management.*

5.7.6.5.9 Call Reset Provisions

5.7.6.5.9.1 If, at any time, a Reset Indication is received indicating a DCE-originated reset, then this shall be confirmed and all other procedures associated with the Call Reset performed.

5.7.6.5.9.2 If, at any time, a Reset Indication is received indicating a DTE-originated reset, then additionally the history compression window shall be reset to the initial state.

Note.—The history decompression window does not need to be cleared because Deflate will never refer to any prior history (Deflate is a sliding-window compressor).

5.7.7 ATN SNDCF Protocol Requirements List

5.7.7.1 Conformance

5.7.7.1.1 An implementation of the ATN Mobile SNDCF shall be used in ATN Airborne and Air/Ground Routers if and only if its PICS is in compliance with the APRLs given in 5.7.7.8.

5.7.7.1.2 An implementation of the ISO/IEC 8802 SNDCF shall be used in ATN End Systems and Routers if and only if its PICS is in compliance with the APRLs given in 5.7.7.2.

5.7.7.1.3 An implementation of the SNDCF for General Topology ISO/IEC 8208 Subnetworks shall be used in ATN End Systems and Routers if and only if its PICS is in compliance with the APRLs given in 5.7.7.4.

5.7.7.2 Subnetwork Dependent Convergence Functions SNDCF for use with ISO/IEC 8802-2 Subnetworks - Functions

Item	Function	ISO/IEC 8473-2 Reference	Status	ATN Support
S802SNUD	Is subnetwork user data of at least 512 octets transferred transparently by the SNDCF ?	5.2	M	M
S802SNTD	Is Transit Delay determined by the SNDCF prior to the processing of User Data ?	5.2	M	M

5.7.7.3 Subnetwork Dependent Convergence Functions SNDCF for use with ISO/IEC 8802-2 Subnetworks - Multi Layer Dependencies

Item	Dependency	ISO/IEC 8473-2 Reference	ATN Support
S802SSg-r	<r> Maximum SN data unit size (RX)	5.2	>=512
S802SSg-s	<s> Maximum SN data unit size (TX)	5.2	>=512

5.7.7.4 Subnetwork Dependent Convergence Functions SNDCF for use with ISO/IEC 8208 Subnetworks - Functions

Item	Function	ISO/IEC 8473-3 Reference	Status	ATN Support
XSNUD	Is Subnetwork User Data of at least 512 octets transferred transparently by the SNDCF ?	5.2	M	M
XSNTD	Is Transit Delay determined by the SNDCF prior to the processing of user data ?	5.2	M	M
	Call Setup Considerations	5.31		
	Is a new call setup:			
XCalla	a. when no suitable call exists ?	5.3.1 a.	O.3	O.3
XCallb	b. when queue threshold reached ?	5.3.1 b.	O.3	O.3
XCallc	c. by systems management ?	5.3.1 c.	O.3	O.3
XCalld	d. when queue threshold reached and timer expires ?	5.3.4	O.3	O.3
XCalle	e. by other local means ?	5.3.1	O.3	O.3
	Call clearing considerations Are calls cleared:	5.3.2		
XClra	a. when idle timer expires	5.3.2 a., 5.3.4	O	O
XClrb	b. when need to re-use circuit	5.3.2 b.	O	O
XClrc	c. by systems management	5.3.2 c.	O	O
XClrd	d. by provider ?	5.3.2 d.	M	M
XClrer	e. by other local means ?	5.3.2	O	O
XPD	X.25 Protocol Discrimination	5.3.3	M	M
XVCC	Resolution of VC collisions	5.3.5	M	M
XMCR	Multiple VCs responding	5.3.6	M	M
XMCI	Multiple VCs initiating	5.3.6	O	O
Xpri	X.25 Priority procedure	5.3.7	O	M

5.7.7.5 Subnetwork Dependent Convergence Functions SNDCF for use with ISO/IEC 8208 Subnetworks - X.25 Call User Data

Item	Parameter	ISO/IEC 8473-3 Reference	Status	ATN Support
PD-s	<s> Protocol Discriminator	5.3.3	M	M
PD-r	<r> Protocol Discriminator	5.3.3	M	M
LI-s	<s> Length Indication	5.3.6	XMCI:M	XMCI:M
LI-r	<r> Length Indication	5.3.6	M	M
Ver-s	<s> SNCR Version	5.3.6	XMCI:M	XMCI:M
Ver-r	<r> SNCR Version	5.3.6	M	M
SNCR-s	<s> SNCR Value	5.3.6	XMCI:M	XMCI:M
SNCR-r	<r> SNCR Value	5.3.6	M	M

5.7.7.6 Subnetwork Dependent Convergence Functions SNDCF for use with ISO/IEC 8208 Subnetworks - ISO/IEC 8208 SNDCF Timers

Item	Timer	ISO/IEC 8473-3 Reference	Status	Values	ATN Support
XIDL	X25 VC Idle	5.3.4	XClra:O	Any	XClra:O
XNVC	additional VC	5.3.4	O	Any	M

5.7.7.7 Subnetwork Dependent Convergence Functions SNDCF for use with ISO/IEC 8208 Subnetworks - SNDCF Multi Layer Dependencies

Item	Dependency	ISO/IEC 8473-3 Reference	Status	Values Supported
XSSg-r	<r> Maximum SN data unit size (Rx)	5.2	>=512	>=512
XSSg-s	<s> Maximum SN data unit size (Tx)	5.2	>=512	>=512

Item	Dependency	ISO/IEC 8473-3 Reference	Status	ATN Support
Xvc	X.25 Virtual call service	5.3.8	M	M
Xdt	X.25 Data transfer	5.3.8	M	M

Item	Dependency	ISO/IEC 8473-3 Reference	Status	ATN Support
Xfc	X.25 flow control procedures	5.3.8	M	M
Xfrp	X.25 flow control + reset packets	5.3.8	M	M
Xccp	X.25 call setup and clear packets	5.3.8	M	M
Xdp	X.25 DTE and DCE data packets	5.3.8	M	M
Xrs	X.25 restart procedures	5.3.8	M	M
XDct	X.25 DCE timeouts	5.3.8	M	M
XDtT	X.25 time limits	5.3.8	M	M
Xpco	X.25 network packet coding	5.3.8	M	M
Xfcn	X.25 flow control parameter negotiation	5.3.8	O	O
Xtd	X.25 transit delay selection and negotiation	5.3.8	O	O
Xtc	X.25 throughput class negotiation	5.3.8	O	O
Xoth	Other X.25 elements	5.3.8	O	O

5.7.7.8 ATN Requirements for Mobile SNDCFs

Note.— This section specifies the requirements for the Mobile SNDCF in Airborne and Air/Ground Routers.

5.7.7.8.1 Major Capabilities

Item	Capability	ATN SARPs Reference	ATN Support
*mcNego	Negotiation of Compression Algorithm	5.7.6.2	M
*mcLocRef	Local Reference Header Compression	5.7.6.3	M
*mcCan	Local Reference Cancellation	5.7.6.3.6	O
McM/I	Local Reference directory maintenance	5.7.6.3	Snvdl:M ^Snvdl:O

Item	Capability	ATN SARPs Reference	ATN Support
*mcACA	ICAO Address Compression Algorithm (ACA)	5.7.6.4	O
mcDeflate	Deflate Compression	5.7.6.5	O

Note.— Snvdl is true when the VDL SNDCF is implemented.

5.7.7.8.2 Call Setup and Clearing Procedures

Item	Function	ATN SARPs Reference	ATN Support
clInit	Call Initiator	5.7.6.2	O.1
clRspd	Call Responder	5.7.6.2	O.1
csDynam	Dynamic Call Setup	5.7.6.2.1.1.1	clInit:O.2
csSys	Call Setup by Systems Management	5.7.6.2.1.1	clInit:O.2
csDef	Use of non-standard Default packet size	5.7.6.2.1.3	clInit:M
csFast	Use of Fast Select	5.7.6.2.1.4	M
csM/I	Local Reference directory maintenance request/acceptance	5.7.6.2.1.5.13 5.7.6.2.2.2.3	^csFast: - McM/I:M
csOther	Use of other optional User Facilities and CCITT-specified DTE facilities	5.7.6.2.1.1.3	O
csCol	Call Collision Resolution	5.7.6.2.2.1.2	clInit:M

Note.— Fast Select only required if supported by subnetwork

Call Setup and Clearing Procedures continued..

Item	Function	ATN SARPs Reference	ATN Support
csAcp	Call Acceptance Procedures	5.7.6.2.1.6	clRspd:M
csRej	Call rejection Procedures	5.7.6.2.1.7	clRspd:M
csOrd	Order of compression Procedures	5.7.6.2.3.2	M
csDiag	Use of call rejection diagnostic codes	5.7.6.2.1.7.3	clInit:M
csClear	Call Clearing Procedures	5.7.6.2.4	M

5.7.7.8.3 Negotiation of Compression Algorithm

Item	Function	ATN SARPs Reference	ATN Support
caMaxd	Indication of the maximum of directories entries in the call user Data	5.7.6.2.1.5.11	mcNego:O

5.7.7.8.4 Local Reference Header Compression

Item	Function	ATN SARPs Reference	ATN Support
lrVC	Opening additional virtual circuits	5.7.6.3.2.1.2	M
*lrDirSize	Local Directory with more than 128 entries	5.7.6.3.1	O
lrProt	Identification of Network Layer Protocol	5.7.6.3.2.2	M

Item	Function	ATN SARPs Reference	ATN Support
lrMod	Processing of SN-UnitData Requests	5.7.6.3.2	M
lrEst	Establishment of new local reference	5.7.6.3.2.4	M
lrTransfer	Transfer of modified ISO 8473 PDU	5.7.6.3.2.6	M
lrInitial	Initial DT PDU Compression	5.7.6.3.3.2	M
lrDerived	Derived DT PDU Compression	5.7.6.3.3.3	M
*lrError-s	Generation of Error PDU Compression	5.7.6.3.3.3	M
lrDiscard	Compression of discarded PDU encapsulated within Error PDU	5.7.6.3.3.4	lrError-s:M
lrCompTr	Transfer of compressed PDUs	5.7.6.3.3.4.1 1	M
lrReceived	Processing of received PDUs	5.7.6.3.4	M
lrUncomp-r	Processing of received uncompressed PDUs	5.7.6.3.4.2	M
LrReset	Purging directories entries on Reset	5.7.6.3.7	mcLocRef:M
lrUnMod-r	Processing of received unmodified PDUs	5.7.6.3.4.2.2	M
lrComp-r	Processing of received compressed data PDUs	5.7.6.3.4.3	M

Item	Function	ATN SARPs Reference	ATN Support
lrError-r	Processing of received compressed Error PDUs	5.7.6.3.4.4	M
lrSNDCFerr-s	Generation of SNDCF Error Report	5.7.6.3.5	M
lrSNDCFerr-r	Processing of received SNDCF Error Report	5.7.6.3.4.5	M

5.7.7.8.5 Local Reference Cancellation

Item	Function	ATN SARPs Reference	ATN Support
lrcMgmt	Management of local references	5.7.6.3.2.5	mcCan:M
lrcRequest-s	Generation of Cancellation Request PDU	5.7.6.3.6	mcCan:M
lrcRequest-r	Processing of incoming Cancellation Request PDU	5.7.6.3.6	mcCan:M
lrcReliable	Reliable transfer of Cancellation Request	5.7.6.3.6	mcCan:M
lrcAccept-s	Generation of Cancellation Accept PDU	5.7.6.3.6	mcCan:M
lrcAccept-r	Processing of incoming Cancellation Accept PDU	5.7.6.3.6	mcCan:M

5.7.7.8.6 ICAO Address Compression Algorithm

Item	Function	ATN SARPs Reference	ATN Support
acOut	Compression of outgoing PDUs	5.7.6.4.1	mcACA:M

Item	Function	ATN SARPs Reference	ATN Support
acIn	Decompression of incoming PDUs	5.7.6.4.1	mcACA:M
acAddr	Address Length Determination	5.7.6.4.2	mcACA:M
acComp	Compression of NSAP Addresses and address prefixes	5.7.6.4.5	mcACA:M
acDecomp	Decompression of NSAP Addresses and address prefixes	5.7.6.4.6	mcACA:M

5.7.7.8.7 PDU Formats

5.7.7.8.7.1 Call Request User Data

Item	Description	ATN SARPs Reference	ATN Support
crLen	Length Indicator	5.7.6.2.1.5.3	M
crVersion	Version Indicator	5.7.6.2.1.5.4	M
crSNCR	Subnetwork Connection Reference (SNCR)	5.7.6.2.1.5.5	M
crComp	Offered Compression Techniques	5.7.6.2.1.5.7	M
crDir	Maximum Directory Size <i>Note.—Dynamically, this field is only generated if Local Reference Compression is offered.</i>	5.7.6.2.1.5.11	M
crAdd-s	Additional User Data on send	5.7.6.2.1.5.12	O

Item	Description	ATN SARPs Reference	ATN Support
crAdd-r	Additional User Data on receive	5.7.6.2.1.5.12	O
MaxDir	Maximum number of directory entries supported	5.7.6.2.1.5.7	≥128

5.7.7.8.7.2 Call Accept User Data

Item	Description	ATN SARPs Reference	ATN Support
caComp	Offered Compression Techniques	5.7.6.2.2.4.3	mcNegot:M
caAdd-s	Additional User Data on send	5.7.6.2.2.4.4	mcNegot:O
caAdd-r	Additional User Data on receive	5.7.6.2.2.4.4	mcNegot:O

5.7.7.8.7.3 Modified ISO/IEC 8473 NPDU

Item	Description	ATN SARPs Reference	ATN Support
npLocRef-s	Local Reference Option field	5.7.6.3.2.3	M

5.7.7.8.7.4 Compressed Initial PDU

Item	Description	ATN SARPs Reference	ATN Support
inType	PDU Type	5.7.6.3.3.2.2	M
inPri	Priority	5.7.6.3.3.2.3	M
inLifetime	Lifetime	5.7.6.3.3.2.4	M

Item	Description	ATN SARPs Reference	ATN Support
inFlags	Flag Bits	5.7.6.3.3.2.5 to 5.7.6.3.3.2.9	M
inLocRef	Local Reference (1 octet)	5.7.6.3.3.2.9	M
inLocRef2	Local Reference (2 octet)	5.7.6.3.3.2.9	l _{rDirSize} :M ^l _{rDirSize} :X
inPDUId	PDU Identifier	5.7.6.3.3.2.10	M
inNSDU	User Data	Figure 5.7-4	M

5.7.7.8.7.5 Compressed Derived PDU

Item	Description	ATN SARPs Reference	ATN Support
drType	PDU Type	5.7.6.3.3.3.2	M
drPri	Priority	5.7.6.3.3.3.3	M
drLifetime	Lifetime	5.7.6.3.3.3.4	M
drFlags	Flag Bits	5.7.6.3.3.3.5 to 5.7.6.3.3.3.8	M
drLocRef	Local Reference (1 octet)	5.7.6.3.3.2.8	M
drLocRef2	Local Reference (2 octet)	5.7.6.3.3.2.8	l _{rDirSize} :M ^l _{rDirSize} :X

Item	Description	ATN SARPs Reference	ATN Support
drPDUIId	PDU Identifier	5.7.6.3.3.3.9	M
drSegOff	Segment Offset	5.7.6.3.3.3.10	M
drTotalLen	Total Length	5.7.6.3.3.3.11	M
drNSDU	User Data	Figure 5.7-4	M

5.7.7.8.7.6 Compressed Error PDU

Item	Description	ATN SARPs Reference	ATN Support
erType	PDU Type	5.7.6.3.3.4.2	M
erPri	Priority	5.7.6.3.3.4.3	M
erLifetime	Lifetime	5.7.6.3.3.4.4	M
erFlags	Flag Bits	5.7.6.3.3.4.5 to 5.7.6.3.3.4.8	M
erLocRef	Local Reference (1 octet)	5.7.6.3.3.2.8	M
erLocRef2	Local Reference (2 octet)	5.7.6.3.3.2.8	lrDirSize:M ^lrDirSize:X
erReason	Discard Reason	5.7.6.3.3.4.9	M
erNSDU	Compressed Header of discarded PDU	5.7.6.3.3.4	M

5.7.7.8.7.7 SNDCF Error Report PDU

Item	Description	ATN SARPs Reference	ATN Support
sfType	PDU Type	5.7.6.3.5	M
sfReason	Discard Reason	5.7.6.3.5	M
sfLocRef	Local Reference	5.7.6.3.5	M
sfLocRef2	Local Reference (2 octet)	5.7.6.3.3.2.9	lrDirSize:M ^lrDirSize:X

5.7.7.8.7.8 Cancellation Request

Item	Description	ATN SARPs Reference	ATN Support
cqType	PDU Type	5.7.6.3.6	mcCan:M
cqRef	Cancellation Reference	5.7.6.3.6	mcCan:M
cqLocRef	Local Reference	5.7.6.3.6	M
cqLocRef2	Local Reference (2 octet)	5.7.6.3.3.2.9	lrDirSize:M ^lrDirSize:X

5.7.7.8.7.9 Cancellation Accept

Item	Description	ATN SARPs Reference	ATN Support
ccType	PDU Type	5.7.6.3.6	mcCan:M
ccRef	Cancellation Reference	5.7.6.3.6	mcCan:M

5.8 ROUTING INFORMATION EXCHANGE SPECIFICATION

5.8.1 Introduction

5.8.1.1 Scope

Note.— This chapter provides requirements and recommendations pertaining to the use of the ISO/IEC 10747 Inter-Domain Routing Protocol over Air/Ground and Ground/Ground Data Links, and the use of ISO/IEC 9542 in support of Route Initiation over Air/Ground Data Links. This chapter is concerned with the interoperability of protocol implementations and provides a compliance statement and APRL for each of the above protocols. It does not specify how Routing Information exchanged using ISO/IEC 10747 is used by Routers when forwarding ISO/IEC 8473 NPDUs, or the application of Routing Policy controlling route aggregation and re-advertisement of routes. These subjects are covered in 5.3.

5.8.1.2 Applicability of Requirements

5.8.1.2.1 All ATN Airborne Routers, with the exception of Airborne Routers implementing the procedures for the optional non-use of IDRP, shall comply with the provisions contained in 5.8.2, 5.8.3, 5.8.3.2.2 to 5.8.3.2.5 inclusive, 5.8.3.2.8 to 5.8.3.2.11 inclusive, 5.8.3.3.2.1, 5.8.3.3.3 and the APRLs specified for an Airborne Router in 5.8.3.4.

5.8.1.2.2 Airborne Routers implementing the procedures for the optional non-use of IDRP shall be compliant with 5.8.2.

5.8.1.2.3 All ATN Air/Ground Routers shall comply with the provisions contained in 5.8.2, 5.8.3, 5.8.3.2.2 to 5.8.3.2.11 inclusive, 5.8.3.3.2.2, 5.8.3.3.3 and the APRLs specified for an Air/Ground Router in 5.8.3.4.

5.8.1.2.4 All Ground/Ground Inter-Domain Routers shall comply with the provisions contained in 5.8.2, 5.8.3.2.2 to 5.8.3.2.11 inclusive, 5.8.3.3.2.2, 5.8.3.3.3 and the APRLs specified for an Ground/Ground Router in 5.8.3.4.

5.8.2 End System to Intermediate System Routing Information Exchange Protocol (ES-IS) over Mobile Subnetworks

5.8.2.1 General

5.8.2.1.1 ATN Airborne and Air/Ground Routers directly connected to a Mobile Subnetwork (e.g. Mode S, AMSS or VDL) shall operate ISO/IEC 9542 over each such Mobile Subnetwork.

5.8.2.1.2 Configuration Information shall be exchanged by both ATN Air/Ground and Airborne Routers over each Mobile Subnetwork connection supporting an adjacency between them.

Note.— The use of ISO/IEC 9542 Configuration Information over Mobile Subnetworks in support of Air/Ground route initiation is specified in 5.3.

5.8.2.1.3 The Mobile Subnetwork Capabilities Parameter

5.8.2.1.3.1 ATN Air/Ground and Airborne Routers shall support the Mobile Subnetwork Capabilities Parameter in the options part of an ISO/IEC 9542 ISH PDU.

5.8.2.1.3.2 The Mobile Subnetwork Capabilities Parameter shall be used in the ATN to convey information about the ATSC Class and the traffic type(s) supported by an ATN Mobile Subnetwork.

5.8.2.1.3.3 The Mobile Subnetwork Capabilities Parameter shall consist of three fields, as illustrated in Figure 5.8-1, and shall not occur more than once in the options part of an ISO/IEC 9542 ISH PDU.

Subnetwork Capabilities Parameter Code	Subnetwork Capabilities Parameter Length	Subnetwork Capabilities Parameter Value
Octet 1	2	3 ... 4

Figure 5.8-1: The Mobile Subnetwork Capabilities Parameter

5.8.2.1.3.4 Encoding of the Mobile Subnetwork Capabilities Parameter

5.8.2.1.3.4.1 The Mobile Subnetwork Capabilities Parameter code field shall be one octet in length and shall always be encoded as binary [1000 0001] to indicate the Mobile Subnetwork Capabilities Parameter.

Note.— The above parameter code and its associated semantics are defined by this specification for the ATN in addition to the parameter codes specified by ISO/IEC 9542. ISO/IEC 9542 only uses eight bit parameter codes with bits 8 and 7 set to one and has reserved a parameter code of 255 for possible future extensions. The future use of the above ATN parameter code by an ISO standard cannot be ruled out but is highly unlikely.

5.8.2.1.3.4.2 The Mobile Subnetwork Capabilities Parameter length field shall be one octet long and shall define the length in octets of the Mobile Subnetwork Capabilities Parameter value field.

5.8.2.1.3.4.3 Mobile Subnetwork Capabilities Parameter Value Field

5.8.2.1.3.4.3.1 The first octet of this field shall indicate the traffic type(s) allowed to pass over the Air/Ground Subnetwork over which the ISO/IEC 9542 ISH PDU is exchanged.

5.8.2.1.3.4.3.2 This octet shall comprise a bit map, where each bit corresponds to a different traffic type.

5.8.2.1.3.4.3.3 The assignment of bits to traffic types shall be according to Table 5.8-4, where bit 0 is the low order bit.

5.8.2.1.3.4.3.4 Setting a bit to one shall indicate that the corresponding traffic type is allowed to pass over the air/ground subnetwork.

5.8.2.1.3.4.3.5 The semantics of bits 5 to 7 shall be reserved for future use and shall always be set to one.

Note 1.— A value of FFh is used to imply no restrictions.

Note 2.— The first octet of the Mobile Subnetwork Capabilities Parameter Value field has the same encoding and semantics as the second octet of the Air/Ground Subnetwork type security Tag Set of the IDRP Security Path Attribute which is defined in 5.8.3.2.3.2.3.

5.8.2.1.3.4.3.6 If bit 0 of the first octet of the Mobile Subnetwork Capabilities Parameter Value field is set to one, then this field shall contain a second octet which defines the ATSC Class supported by that Air/Ground Subnetwork.

Note.— Bit 0 of the first octet set to one indicates that the Air/Ground Subnetwork is available to the ATN Operational Communications traffic type - Air Traffic Service Communications traffic category.

5.8.2.1.3.4.3.7 If present, the second octet of the Mobile Subnetwork Capabilities Parameter Value field shall be encoded according to Table 5.8-1.

Table 5.8-1: Encoding of Supported ATSC Class

Value	ATSC Class
0000 0001	A
0000 0010	B
0000 0100	C
0000 1000	D
0001 0000	E
0010 0000	F
0100 0000	G
1000 0000	H

Note.— ATSC Class “H” is the lowest and Class “A” is the highest class.

5.8.2.1.4 Route Redirection information shall not be exchanged between an ATN Air/Ground and Airborne Router.

5.8.2.2 ATN Protocol Requirements List - ISO/IEC 9542

5.8.2.2.1 An implementation of the ISO/IEC 9542 protocol shall be used in ATN Airborne and Air/Ground Routers, if and only if its PICS is in compliance with the APRL given in Table 5.8-2.

Table 5.8-2 ISO/IEC 9542 - Intermediate System

Item	Protocol Function	Clauses	ISO Status	ATN Support
CI	Is configuration information supported over the associated subnetwork?	ATN SARPs Ref.: 5.8.2	O	M
RI	Is redirection information supported over the associated subnetwork?	ATN SARPs Ref.: 5.8.2	O	OX
	Are the following functions supported?			
ErrP	Protocol Error Processing	6.13	M	M
HCsV	PDU Header Checksum Validation	6.12	M	M

Item	Protocol Function	Clauses	ISO Status	ATN Support
HCsG	PDU Header Checksum Generation	6.12	O	O
RpCf	Report Configuration	6.2, 6.2.2	CI:M	M
RcCf	Record Configuration	6.3, 6.3.1	CI:M	M
FlCf	Flush Old Configuration	6.4	CI:M	M
RqRd	Request Redirect	6.8	RI:M	OX
CfNt	Configuration Notification	6.7	CI:O	OX
CTGn	ESCT Generation	6.3.2	CI:O	OX
AMGn	Address Mask (only) generation	6.8	RI:O	OX
SMGn	Address mask and SNPA Mask generation	6.8	RI:O	OX
	Are the following PDUs Supported?			
ESH-r	<r> End System Hello	7.1, 7.5	CI:M	O
ISH-<r>	<r> Intermediate System Hello	7.1, 7.6	CI:O	M
ISH-<s>	<s> Intermediate System Hello	7.1, 7.6	CI:M	M
RD-s	<s> Redirect	7.1, 7.7	RI:M	OX
RD-r	<r> (ignore) Redirect	6.9, 7.1, 7.7	M	M
	Are the following PDU fields supported?			
FxPt	<s> Fixed Part <r> Fixed Part	7.2.1, 7.2.7 7.2.1, 7.2.7	M M	M M
SA-r	<r> Source Address, one or more NSAPs	7.3.1/2/3	CI:M	M
NET-s	<s> Network Entity Title	7.3.1/2/4	M	M
NET-r	<r> Network Entity Title	7.3.1/2/4	ISH-r:M	ISH-r:M
DA-s	<s> Destination Address	7.3.1/2/5	RI:M	OX
BSNPA-s	<s> Subnetwork Address	7.3.1/2/6	RI:M	OX
Scty-s	<s> Security	7.4.2	O	O
Scty-r	<r> Security	7.4.2	O	O

Item	Protocol Function	Clauses	ISO Status	ATN Support	
Pty-s	<s> Priority	7.4.3	O	O	
Pty-r	<r> Priority	7.4.3	O	O	
QoSM-s	<s> QOS Maintenance	7.4.4	RI:O	OX	
AdMk-s	<s> Address Mask	7.4.5	RI:O	OX	
SNMk-s	<s> SNPA Mask	7.4.6	RI:O	OX	
MSNC-s	<s> Mobile Subnetwork Capabilities	ATN SARPs Ref. 5.8.2.1.3, 5.3.5.2.6.5	- -	ISH-s:M	
MSNC-r	<r> Mobile Subnetwork Capabilities	ATN SARPs Ref: 5.8.2.1.3, 5.3.5.2.6.9	- -	ISH-r:M	
ESCT-s	<s> Suggested ES Configuration Timer	7.4.7	CI:O	OX	
ESCT-r	<r> (ignore) Suggested ES Configuration Timer	7.4.7	ISH-r:M	ISH-r:M	
OOpt-r	<r> (ignore) unsupported or unknown options	7.4.1	M	M	
OOpt-s	<s> Other options		P	P	
	Parameter Ranges				
HTv	What range of values can be set for the Holding Time Field in transmitted PDUs ?	ATN SARPs Ref.: 5.3.5.2.9	M	M	from: 0 seconds to: 65535 seconds with a tolerance of: 10%
CTv	If configuration information is supported, what range of values can be set for the Configuration Timer ?	ATN SARPs Ref.: 5.3.5.2.5	CI:M	M	from: 0 seconds to: 65535 seconds with a tolerance of: 10%

Note 1.— In case where IDRP is used over the Air/Ground link, the Holding Time field of transmitted ISH PDUs is preferably set to 65534 seconds as recommended in 5.3.5.2.9. The purpose of this recommendation is to effectively suppress the regular generation of ISH PDUs on the Air/Ground link.

Note 2.— In case where the procedures for the optional non-use of IDRP are used on the Air/Ground link, the Holding Time field of the transmitted ISH PDUs and the Configuration Timer are set appropriately based on operational experience so that the exchange of ISH PDUs ensures a regular update of the respective FIBs in both the Air/Ground and Airborne Routers, without overloading the Air/Ground link.

5.8.3 Intermediate System to Intermediate System Inter-Domain Routing Information Exchange Protocol

5.8.3.1 General

5.8.3.1.1 With the exception of Airborne Routers that implement the procedures for the optional non-use of IDRP, ATN Routers shall implement ISO/IEC 10747, including the ATN Specific Features specified in this section, and the APRLs specified in 5.8.3.4.

5.8.3.2 ATN Specific Features

5.8.3.2.1 Purpose of ATN Specific Features

Note.—The ATN Specific Features specified in the following subsections support user requirements concerned with:

- a) *Ensuring that application data passed over Air/Ground data links conforms with any national and/or ITU restrictions applicable to that Air/Ground data link;*
- b) *Ensuring that a classification scheme can be applied to routes throughout the ATN Ground Environment, reflecting the expected QoS available over each such route;*
- c) *Ensuring that information on Air/Ground subnetwork types that a route passes over is available for determining which route to choose for a given application's data;*
- d) *Ensuring that changes to routing information that report negative changes (e.g. a downgrading of the classification of a route) are reported in a timely manner.*

5.8.3.2.2 Use of the Security Path Attribute

5.8.3.2.2.1 ATN Routers supporting inter-domain routing shall support the IDRP Security Path Attribute with a Security Registration Identifier set to the value defined in 5.6.2.2.6 for the ATN Security Registration Identifier.

5.8.3.2.2.2 The Security Information provided with a so identified IDRP Security Path Attribute shall consist of zero one or more Security Tag Sets as defined in 5.6.2.2.6.

5.8.3.2.2.3 The following Security Tag Sets shall be supported:

- a) The Air/Ground Subnetwork type, as defined in 5.8.3.2.3.2, and
- b) The ATSC Class, as defined in 5.8.3.2.3.3.

5.8.3.2.2.4 **Recommendation.** — *When an ATN Router supports data classified according to a security policy and for the purpose of implementing mandatory access controls, then the ATN Router should also support the security classification Security Tag Set defined in 5.6.2.2.6.*

5.8.3.2.2.5 When a route is available over more than one Air/Ground subnetwork type, then a separate Security Tag Set shall be encoded into this field to identify each Air/Ground subnetwork that may support the route.

5.8.3.2.2.6 When an Air/Ground Subnetwork is restricted to carrying data of only certain traffic types, then the Security Tag Set that identifies that Air/Ground Subnetwork shall enumerate the Traffic Types that may pass over that subnetwork.

5.8.3.2.2.7 At most one ATSC Class Security Tag Set shall be present in a route's Security Path Attribute.

5.8.3.2.2.8 An ATSC Class Security Tag Set shall not be present when one or more Air/Ground Subnetwork Security Tag Sets are also present, and when none of these Air/Ground Subnetwork Security Tag Sets indicates support of ATN Operational Communications traffic type — Air Traffic Service Communications traffic category.

5.8.3.2.3 Encoding of the Security Path Attribute Security Information Field

5.8.3.2.3.1 General

5.8.3.2.3.1.1 The Security Path Attribute Security Information Field shall comprise zero, one or more Security Tag Sets as defined in 5.6.2.2.6.

Note.— The Security Tag Set format defined for use with CLNP in 5.6, has been adopted here as a convenient method for the extensible encoding of security related information.

5.8.3.2.3.2 Encoding of the Air/Ground Subnetwork Type Security Tag Set

5.8.3.2.3.2.1 The Tag Set Name of the Air/Ground Subnetwork Type Security Tag Set shall be set to [0000 0101], and the Security Tag shall always be two octets in length.

5.8.3.2.3.2.2 The first (lowest numbered) octet of the Security Tag shall define the Air/Ground subnetwork type over which the route may be available according to Table 5.8-3.

Table 5.8-3 Air/Ground Subnetwork Type Security Tag Values

Subnetwork Type	Security Tag (1st Octet)
Mode S	0000 0001
VDL	0000 0010
AMSS	0000 0011
Gatelink	0000 0100
HF	0000 0101

5.8.3.2.3.2.3 The second (highest numbered) octet of the Security Tag shall indicate the Traffic Types allowed to pass over the Air/Ground subnetwork identified in the first octet.

5.8.3.2.3.2.4 This octet shall comprise a bit map, where each bit corresponds to a different Traffic Type. A value of FFh shall be used to imply no restrictions.

5.8.3.2.3.2.5 The assignment of bits to Traffic Type shall be according to Table 5.8-4, where bit 0 is the low order bit:

Table 5.8-4 Identification of Permissible Traffic Types

Bit Number	Traffic Type
0	ATN Operational Communications — Air Traffic Service Communications
1	ATN Operational Communications — Aeronautical Operational Control
2	ATN Administrative Communications
3	General Communications
4	ATN Systems Management Communications

5.8.3.2.3.2.6 The semantics of bits 5 to 7 shall be reserved for future use and shall always be set to one.

5.8.3.2.3.3 Encoding of the ATSC Class Security Tag Set

5.8.3.2.3.3.1 The Tag Set Name of the ATSC Class Security Tag Set shall be set to [0000 0110] if the associated route is available to both ATSC and non-ATSC traffic.

5.8.3.2.3.3.2 The Tag Set Name of the ATSC Class Security Tag Set shall be set to [0000 0111] if the associated route is available to ATSC traffic only.

5.8.3.2.3.3.3 The Security Tag shall always be one octet in length.

5.8.3.2.3.3.4 If a Security Tag with one of these Tag Set Names is received which is longer than one octet, then all octets after the first octet shall be ignored.

5.8.3.2.3.3.5 When a Security Tag with one of these Tag Set Names is present, the Security Tag shall identify the ATSC Class(es) supported by the route.

5.8.3.2.3.3.6 The ATSC Class(es) supported shall be identified according to Table 5.8-5, where bit 0 is the low order bit, and setting a bit to one shall indicate that the corresponding ATSC Class is supported.

5.8.3.2.3.3.7 A bit set to zero shall indicate that the corresponding ATSC Class is not supported.

Table 5.8-5 Identification of Supported ATSC Classes

Bit Number	ATSC Class
0	A
1	B
2	C
3	D
4	E
5	F
6	G
7	H

5.8.3.2.4 Update of Security Information

5.8.3.2.4.1 The Air/Ground Subnetwork Type

5.8.3.2.4.1.1 When a Route is:

- a) either advertised or received by an Air/Ground Router over an adjacency supported by one or more Air/Ground Subnetworks, and
- b) contains a Security Path Attribute, and
- c) has the ATN Security Policy Identifier, as the Security Path Attribute's Security Registration Identifier, then

the Security Path Attribute's Security Information shall be updated as follows:

- 1) an Air/Ground Subnetwork Type Security Tag shall be added for each Air/Ground Subnetwork supporting the adjacency and which is not already contained in the Security Information;
- 2) for each Air/Ground Subnetwork Type Security Tag present in or added to the route, if ITU requirements or local policies restrict the Traffic Types that may pass over that subnetwork then the second octet of the security tag shall be modified to set to zero the bits corresponding to each traffic type not supported by that Air/Ground Subnetwork.

Note.— According to the procedures specified in 5.3.5.2.12 for the optional non-use of IDRP over an air-ground data link, this update of the Security information also includes routes which have been originated by an Air/Ground router on behalf of an Airborne router not implementing IDRP.

5.8.3.2.4.1.2 When a route containing one or more Air/Ground Subnetwork Tags is advertised over an adjacency that supports only ATSC traffic, the Air/Ground Subnetwork Tags shall be updated such that the second octet of the security tag shall be modified to set to zero the bits corresponding to all Traffic Types other than ATSC.

5.8.3.2.4.1.3 Any Air/Ground Subnetwork Security Tags with a second octet that is all zeroes shall be removed from the route.

5.8.3.2.4.1.4 If all Air/Ground Subnetwork Security Tags present have a zero second octet then the route shall not be advertised over this adjacency.

5.8.3.2.4.2 The ATSC Class

5.8.3.2.4.2.1 When a route is advertised to an adjacent BIS, then:

- a) if the route has been originated by an Air/Ground Router according to the procedures for the optional non-use of IDRP (as specified in 5.3.5.2.12), and the adjacency with the Airborne Router is over an Air/Ground Data Link approved for ATSC use, then an ATSC Class Security Tag shall be added to the route identifying the ATSC Class(es) supported by the Adjacency with that Airborne Router;
- b) if the route had been received from an Airborne Router by an Air/Ground Router, over an Air/Ground Data Link approved for ATSC use, then an ATSC Class Security Tag shall be added, replacing any that may already be present, identifying the ATSC Class(es) supported by the adjacency with that Airborne Router;
- c) if the route
 - 1) has been originated locally (i.e. within the same Routing Domain), by a Router other than an Airborne Router, and
 - 2) is to be advertised to an adjacent BIS over an adjacency supported by one or more subnetworks approved for ATSC traffic, thenan ATSC Class security tag shall be added to the route identifying the ATSC Class(es) supported by the adjacency;

Note.— In the case of an Airborne Router, the ATSC Class is inserted by the Air/Ground Router (see case (b) above), and this avoids an Airborne Router having to know which Air/Ground data links are approved for ATSC use.

- d) if the route
 - 1) has been received from another BIS, and
 - 2) is to be advertised to an adjacent BIS over an adjacency supported by one or more subnetworks approved for ATSC traffic, and

- 3) has an ATSC Class security tag that is higher than the ATSC Class that the System Administrator has specified as being supported by the adjacency, then

the ATSC class of the route shall be downgraded, as specified below, to the ATSC Class supported by the adjacency.

- e) if the route

- 1) has been received from another BIS and
- 2) is to be advertised to an adjacent BIS over an adjacency supported by subnetworks that are not approved for ATSC Traffic, then

the ATSC Class security tag shall be removed from the route before it is advertised to the adjacent BIS.

5.8.3.2.4.2.2 When an ATSC Class Security tag is added to a route, then the value of the Tag Set Name shall be set according to 5.8.3.2.3.3 and depending upon whether the adjacency has been specified to support ATSC traffic only or both ATSC and non-ATSC traffic.

5.8.3.2.4.2.3 When the ATSC Class Security Tag indicating support for both ATSC and non-ATSC traffic is updated then the Tag Set Name shall be changed to that indicating support for ATSC only traffic if the adjacency is specified to support only ATSC traffic.

5.8.3.2.4.2.4 In all other cases, the ATSC Class Security Tag Name shall not be modified.

Note.— The Tag Set Name is set to [0000 0110] when both ATSC and non-ATSC traffic is supported, and to [0000 0111] when only ATSC traffic is supported.

5.8.3.2.4.2.5 When the ATSC Class is downgraded, the ATSC Class Security Tag Set shall be modified such that all bits indicating support for an ATSC Class higher than that supported by the local policy shall be set to zero, and the bit corresponding to the highest ATSC Class supported by local policy shall be set to one. All remaining bits shall be unaffected.

5.8.3.2.4.2.6 An ATSC Class Security Tag shall not be present in a route's security information, if an Air/Ground Subnetwork Security Tag is also present indicating that the Air/Ground Subnetwork does not support ATSC Traffic.

5.8.3.2.4.2.7 When an ATSC Class Security Tag indicating support for ATSC only is present in a route, an Air/Ground Subnetwork Security Tag when present in the same route shall not indicate support for any traffic type other than ATSC.

5.8.3.2.4.3 The Security Classification

5.8.3.2.4.3.1 When it is required by the local Security Policy that:

- a) the router supports classified data, and

- b) a route is advertised to an adjacent BIS, and
- c) the highest level of protection offered by the subnetworks supporting the adjacency is lower than that reported by a Security Classification Security Tag,

then that Security Tag shall be replaced by a Security Classification Security Tag reporting the highest protection offered by those subnetworks, as specified in the applicable security policy.

5.8.3.2.5 Route Selection

Note.— ISO/IEC 10747 clause 7.16.2 permits a Loc-RIB that is identified by a RIB_Att containing the Security Path Attribute, to contain more than one route to the same NLRI, provided that those routes provide the same level of protection.

5.8.3.2.5.1 When the Security Registration Identifier in the IDRP Security Path Attribute is the ATN Security Registration Identifier, and when no security classification is present in the route's security information, then all such routes shall be assumed to offer the same level of protection.

Note.— The purpose of this statement is to permit, within the limitations imposed by ISO/IEC 10747, the existence in the Loc-RIB of multiple routes to the same aircraft which differ in the security related information.

5.8.3.2.5.2 During the Phase 2 Routing Decision process, when:

- a) two or more routes to the same or overlapping destination are found in the Adj-RIB-Ins identified by a RIB_Att that includes the Security Path Attribute, but which differ in the security information contained in their security path attribute, then all such routes shall be selected and copied to the corresponding Loc-RIB.
- b) two routes are found in the Adj-RIB-Ins identified by a RIB_Att that includes the Security Path Attribute, which differ in the security information contained in their security path attribute, and when the NLRI of the less preferable route is a proper subset of the NLRI of the more preferable route, then only the more preferable route shall be copied to the corresponding Loc-RIB. Otherwise, both such routes shall be copied to the corresponding Loc-RIB.

5.8.3.2.6 Route Aggregation and Route Information Reduction

5.8.3.2.6.1 General

5.8.3.2.6.1.1 ATN Routers shall implement the procedures for Route Aggregation and Route Information Reduction when required to do so according to 5.8.3.4.2.

Note 1.— Route Aggregation is defined by ISO/IEC 10747 as a procedure for the merging or aggregation of two routes in order to form a single replacement route. Route Aggregation may be applied as the result of a Routing Policy decision in order to reduce the routing information advertised to an adjacent Router. It is also necessary to aggregate two routes in the same Loc-RIB and with identical NLRI prior to being advertised to an adjacent Router. This latter case of Route Aggregation is automatic, not subject to Routing Policy, and necessary for the proper dissemination of routing information.

Note 2.— Route Information Reduction is defined by ISO/IEC 10747 as a procedure for replacing two or more NSAP Address Prefixes in a route's NLRI by a single shorter NSAP Address Prefix. The decision on when to apply Route Information Reduction is also subject to Routing Policy and is typically associated with the application of Route Aggregation when applied as a result of Routing Policy.

5.8.3.2.6.2 Policy Based Route Aggregation

5.8.3.2.6.2.1 **Recommendation.**— *An Air/Ground Router should aggregate all routes to destinations in Routing Domains in its own ATN Island, other than those to destinations in its own Routing Domain.*

5.8.3.2.6.2.2 **Recommendation.**— *An Air/Ground Router should aggregate all routes to destinations in ATN Islands, other than those to destinations in its own ATN Island.*

5.8.3.2.6.2.3 **Recommendation.**— *ATN Ground/Ground Routers should perform Route Aggregation and Route Information Reduction on routes to ground destinations, in line with local policy requirements for reducing the amount of routing information distributed within the ATN Ground Environment.*

Note.— The need for this will be determined according to local topology and NSAP Address Assignment and is outside of the scope of this specification. However, this feature is a necessary condition for the development of a large scale and scaleable internet.

5.8.3.2.6.2.4 The selection of candidate routes for aggregation shall be performed separately for each adjacent BIS according to a filter on each route's destination, with a combination of inclusion and exclusion filters.

Note.— For example, filters might be applied in order to select all routes to NSAP Address Prefixes within the local ATN Island, while excluding those to the local Administrative Domain.

5.8.3.2.6.3 Aggregation of Routes in the Same Loc-RIB with Identical NLRI

5.8.3.2.6.3.1 When two or more routes exist in the same Loc-RIB which have identical NLRI, then such routes shall be aggregated after the application of local policy rules that select routes for re-advertisement to each adjacent BIS.

5.8.3.2.6.3.2 Such routes shall be consequently copied to the associated Adj-RIB-Out.

5.8.3.2.6.3.3 For each adjacent BIS, the resulting aggregated route shall be inserted into the associated Adj-RIB-Out.

5.8.3.2.6.3.4 In order to aggregate such routes, an ATN Router shall apply one of the following two strategies:

- a) **True Route Aggregation:** the routes are aggregated according to ISO/IEC 10747 route aggregation procedures and the procedures for aggregation of the security path attribute specified in 5.8.3.2.6.4 below.
- b) **Route Merging:** the routes are merged by arbitrarily selecting one of these routes and updating its security path attribute to the value that would have resulted had the

routes been aggregated, as above. The selected route with its updated security path attribute is then the result of the merging procedure.

Note 1.— The former of the two strategies is preferred.

Note 2.— The second strategy has been introduced as an interim measure to simplify initial implementations. However, this second strategy leads to a situation where routing decisions based on RD_PATH information cannot be performed, as this information is lost in the merging process. The second strategy may therefore be deleted in a later revision of these SARPs.

Note 3.— Whenever local policy rules that select routes for advertisement to adjacent BISs select different combinations of routes from the same Loc-RIB and with identical NLRI, for advertisement to different adjacent BISs, then the Route Aggregation or Merging procedure has to be carried out separately for each Adj-RIB-Out. For each Adj-RIB-Out, only those routes which are eligible for advertisement to the corresponding BIS will be input to the merging/aggregation procedure. For example, a route may not be eligible for advertisement to an adjacent BIS due to distribution restrictions or a potential route loop recognised from the RD_PATH information.

Note 4.— An aggregated route resulting from these procedures may also be aggregated with other routes in an Adj-RIB-Out, due to the application of local policy rules.

5.8.3.2.6.4 Aggregation of the Security Path Attribute Information Field

5.8.3.2.6.4.1 General

5.8.3.2.6.4.1.1 ATSC and non-ATSC routes with dissimilar NLRI shall not be aggregated.

Note 1.— An ATSC Route is a route containing an ATSC Class Security Tag in its Security Path Attribute. A non-ATSC Route is similarly a route that does not contain an ATSC Class Security Tag in its Security Path Attribute.

Note 2.— Two possible strategies for aggregating such routes were considered. However, neither gave a satisfactory outcome. This is because the aggregated route must either be identified as an ATSC route, or a non-ATSC route. If the aggregated route is identified as a non-ATSC route, then this would result in ATSC routes being “hidden” when aggregated with non-ATSC routes. On the other hand, if the aggregated route is identified as an ATSC route, then this would result in a situation where an aggregated route that was apparently approved for ATSC Traffic, included a destination which could not be reached over a path that was approved end-to-end for ATSC Traffic. This runs the risk of creating a “black hole” for ATSC Traffic.

5.8.3.2.6.4.1.2 Similarly, routes available to ATSC traffic only and routes available to both ATSC and non-ATSC traffic with dissimilar NLRI shall not be aggregated.

5.8.3.2.6.4.1.3 Otherwise, the aggregation rules for the security information field contained in security path attributes that include the ATN Security Registration Identifier shall be as follows.

5.8.3.2.6.4.2 Air/Ground Subnetwork Security Tag

5.8.3.2.6.4.2.1 The aggregated security path attribute shall comprise each air/ground subnetwork security tag contained in the security path attribute of the component routes.

5.8.3.2.6.4.2.2 When an air/ground subnetwork type security tag for the same air/ground subnetwork type occurs in more than one component route, then these shall be combined by a logical “OR” of the second octet of the Air/Ground Subnetwork type security tags.

5.8.3.2.6.4.2.3 Only a single air/ground subnetwork type security tag for each distinct air/ground subnetwork type shall be present in the aggregated route.

5.8.3.2.6.4.3 ATSC Class Security Tag

5.8.3.2.6.4.3.1 General

5.8.3.2.6.4.3.1.1 If an ATSC Class Security Tag is not present in any component route, then the aggregated route shall not contain an ATSC Class Security Tag.

5.8.3.2.6.4.3.2 Non-Identical NLRI in Component Routes

5.8.3.2.6.4.3.2.1 If the NLRI of the component routes is not identical then, when an ATSC Class security tag with the same Tag Set Name occurs in all component routes the aggregated route shall contain an ATSC Class security tag with the same Tag Set Name.

5.8.3.2.6.4.3.2.2 The ATSC Class of the aggregated route shall be the lowest ATSC Class of the aggregated route's component routes, indicated by setting the value of the corresponding bit in the security tag value **to one**.

5.8.3.2.6.4.3.2.3 All the other bits in this tag shall be set to zero.

5.8.3.2.6.4.3.3 Identical NLRI in Component Routes

5.8.3.2.6.4.3.3.1 If the NLRI of the component routes is identical then, when an ATSC Class security tag occurs in one or more component routes then the aggregated route shall contain an ATSC Class security tag.

5.8.3.2.6.4.3.3.2 If an ATSC Class Tag Set occurs in all component routes and the ATSC Class Tag Set Names in all such tag sets are identical, then the Tag Set Name of the aggregated route shall be the same as in the component routes.

5.8.3.2.6.4.3.3.3 If the ATSC Class Tag Set Names in the component routes are different, or one or more component routes do not include an ATSC Class Security Tag, then the ATSC Class Security Tag Set in the aggregated route shall use the Tag Set Name that indicates that the route is available for both ATSC and non-ATSC traffic.

Note.— This Tag Set Name is defined by 5.8.3.2.3.3.1 to take the value [0000 0110].

5.8.3.2.6.4.3.3.4 The ATSC Class of the aggregated route shall be formed by a logical 'OR' of the encoded representation of the supported ATSC Class in each of the aggregated route's component routes that contains an ATSC Class security Tag.

5.8.3.2.6.4.3.3.5 If none of the component routes contains an ATSC Class security tag, then the aggregated route shall not contain an ATSC Class security tag.

5.8.3.2.6.4.3.4 Security Classification Security Tag

5.8.3.2.6.4.3.4.1 When a Security Classification security tag occurs in all component routes, then the aggregated route shall contain a Security Classification security tag.

5.8.3.2.6.4.3.4.2 This tag shall be set to the lowest classification from the classifications given to the aggregated route's component routes.

5.8.3.2.6.4.3.4.3 If a Security Classification security tag is not present in at least one component route then the aggregated route shall not contain a Security Classification security tag.

5.8.3.2.6.5 Route Information Reduction

5.8.3.2.6.5.1 **Recommendation.**— *An Air/Ground Router should perform Route Information Reduction as permitted by the ATN Addressing Plan, before advertising aggregated routes to an Airborne Router.*

Note.— *It is intended that the result of Route Information Reduction is a single NSAP Address Prefix to each destination group to which aggregation is performed. However, this will only be possible if NSAP Addresses have been allocated appropriately (e.g. all systems within the same ATN Island have a single common prefix for all such addresses).*

5.8.3.2.6.5.2 Route Information Reduction shall be performed using local policy rules, with such routing policy rules required to specify when a set of NSAP Address Prefixes is replaced by a shorter NSAP Address Prefix. Two types of rules shall be supported:

- a) The explicit replacement of a set of NSAP Address Prefixes by another shorter NSAP Address Prefix, only when all members of the set are present, or
- b) The explicit replacement of a set of NSAP Address Prefixes by another shorter NSAP Address Prefix when any members of the set are present.

5.8.3.2.7 Frequency of Route Advertisement

Note.— *ISO/IEC 10747 clause 7.17.3.1 requires that the advertisement of feasible routes to some common set of destinations received from BISs in other Routing Domains must be separated in time by at least **minRouteAdvertisementInterval** except for certain identified cases. The list of exceptions to this requirement is extended by this specification.*

5.8.3.2.7.1 If a selected route to a given destination changes in respect of the Security Information contained in its Security Path Attribute, then that route shall be immediately re-advertised to all adjacent BISs to which that route had previously been advertised and not since withdrawn.

5.8.3.2.7.2 The procedure for ensuring a minimum time interval of minRouteAdvertisementInterval between successive advertisements of routes to the same destination shall not apply in this case.

5.8.3.2.8 Interpretation of Route Capacity

5.8.3.2.8.1 For the ATN environment, the CAPACITY path attribute shall contain one of the values listed in Table 5.8-6, and shall be assumed to have the semantics given there:

Table 5.8-6 Interpretation of Capacity Route Metric

Value	Meaning
1 ... 9	Unassigned
13	0 - 19.2 Kbits/sec
12	19.2 - 56 Kbits/sec
11	56 - 1500 Kbits/sec
10	> 1500 Kbits/sec
14 .. 255	Unassigned

Note.— The CAPACITY path attribute is a well known mandatory attribute that is used to denote the traffic handling capacity of the RD_PATH listed in the same UPDATE PDU. Higher values indicate a lower traffic handling capacity than do low values.

5.8.3.2.9 Network Layer Reachability Information

5.8.3.2.9.1 General

5.8.3.2.9.1.1 In support of ATN communications, ATN Routers shall encode the NLRI Addr_info field of each route as a list of NSAP Address Prefixes.

5.8.3.2.9.1.2 The proto_type, and proto_length fields shall be set to 1 and the Protocol field shall be set to X'81' in order to signal support of ISO/IEC 8473.

5.8.3.2.9.2 NSAP Address Prefix Alignment

5.8.3.2.9.2.1 When originating a route or performing route information reduction, an ATN Router shall only generate NSAP address prefixes that are octet-aligned.

Note 1.— For IDRP, ATN NSAP address prefixes will be eleven octets (or less).

Note 2.— 5.8.3.2.12 specifies the RIB-Atts that an ATN Router must support.

Note 3.— The above requirement does not modify the requirement in ISO/IEC 10747 to be able to accept and correctly handle a non-octet aligned NSAP Address Prefix.

Note 4.— The above requirement simplifies prefix matching.

5.8.3.2.10 BISPDU Authentication

5.8.3.2.10.1 ATN Routers shall support the validation of BISPDU using Authentication Type 1.

5.8.3.2.10.2 When an ATN Router initiates a BIS-BIS connection, it shall set the value of the Authentication Code in the OPEN PDU to 1, in order to indicate that the Validation field in the header of all BISPDU sent over the BIS-BIS connection will contain an unencrypted checksum.

5.8.3.2.10.3 When an authentication code of 1 is specified in the Authentication Code of the OPEN BISPDU that initiated a BIS-BIS connection then, an ATN Router shall generate a validation pattern according to clause 7.7.1 of ISO/IEC 10747, for each BISPDU that it sends over that connection, and similarly validate the validation pattern of all received BISPDU on such a connection.

Note.— The use of ISO/IEC 10747 type 2 authentication is under consideration for specification in future versions of these SARPs.

5.8.3.2.10.4 The type 1 authentication code shall be generated according to the MD4 specification published in RFC 1320.

Note 1.— The interpretation of MD4 given in Annex B of ISO/IEC 10747 is open to ambiguous interpretation and may lead to interoperability problems.

Note 2.— RFC 1320 supersedes RFC 1186 which was the basis for ISO/IEC 10747 Annex B. Specifications of MD4 algorithm contained in these two RFC documents are technically equivalent.

5.8.3.2.11 Restrictions on Route Advertisement

5.8.3.2.11.1 A route shall not be advertised to a BIS in another RD when:

- a) The route contains the receiving RD's RDI in its RD_PATH path attribute, or
- b) The route's RD_PATH path attribute contains the RDI of a routing domain confederation which is being entered when the route is advertised to the other RD.

Note.— This is essential to avoid long lived black holes following the explicit withdrawal of an unfeasible route and when many alternate paths are available (e.g. within an ATN Island Backbone RDC).

5.8.3.2.12 RIB_Att Support

Table 5.8-7 ISO/IEC 10747 Mandatory Requirements, for Which Support is Optional for ATN Airborne Routers

	ISO Mandatory Requirement	Notes
1.	Internal Update Procedures	<i>Note 1.— There is only ever a single BIS per routing domain on board an aircraft, and hence, internal update is not applicable.</i>

	ISO Mandatory Requirement	Notes
2.	Operation of minRouteAdvertisementInterval Timer	<i>Note 2.— An aircraft is always an End Routing Domain, and hence will never re-advertise routes.</i>
3.	Recognition of Next Hop Attribute	<i>Note 3.— No requirement for support in the ATN.</i>
4.	Recognition of Residual Error, Expense, Transit Delay and Priority Distinguishing Path Attributes	<i>Note 4.— Never negotiated for use in the ATN.</i>
5.	Support of RIB Refresh	<i>Note 5.— RIB Refresh is necessary for long lived adjacencies rather than the short lived adjacencies anticipated for ATN Mobiles.</i>
6.	Support of DIST_LIST_EXCL	<i>Note 6.— There are no known user requirements to control the distribution of routes to or from Mobile Systems. Implementation may also be problematic due to changing point of attachment to the Fixed ATN.</i>
7.	Support of Partial Source Routing	<i>Note 7.— There are no known user requirements for partial source routing.</i>

5.8.3.2.12.1 An ATN Router incorporating IDRP shall support the following RIB_Att sets:

- a) The empty RIB_Att
- b) SECURITY

and shall attempt to negotiate the use of all those RIB_Atts it supports when opening a BIS-BIS connection.

5.8.3.2.12.2 The semantics of the empty RIB_Att shall be taken as implying that routes advertised under the empty RIB_Att:

- a) have a classification of “Unclassified”,
- b) have not passed over any mobile subnetworks; and
- c) are not available to ATSC traffic.

5.8.3.2.13 Additional Update PDU Error Handling

5.8.3.2.13.1 When an UPDATE PDU is received with a Security Path Attribute containing an ATN Security Registration Identifier and Security Information that contains:

- a) an ATSC Class Security Tag Set, and
- b) One or more Air/Ground Subnetwork type Security Tag Sets, such that none of these Security Tag Sets indicates support of ATN Operational Communications — Air Traffic Service Communications, then the UPDATE PDU shall be discarded and an IDRP ERROR PDU generated with an Error_Code indicating an UPDATE_PDU_Error, and an error subcode set to 64.

5.8.3.2.14 CLNP Data PDU Parameters

5.8.3.2.14.1 The CLNP Data PDU that carries a BISPDUs between two ATN Routers shall include:

- a) A Security Parameter providing an ATN Security Label indicating a traffic type of “Systems Management”
- b) A priority parameter indicating a PDU priority of 14.

Note.— To ensure the exchange of ISO/IEC 10747 BISPDUs over an air/ground adjacency under the above traffic type classification, the air/ground router or airborne router respectively must be configured in a way that includes ATN Systems Management Communications in the set of permissible traffic types allowed to pass over the air/ground subnetwork(s) forming the air/ground adjacency. Otherwise, an IDRP connection may not be established over the air/ground adjacency; consequently no CLNP PDUs will ever flow over it and the adjacency will be unusable.

5.8.3.3 Compliance with ISO/IEC 10747

5.8.3.3.1 General

5.8.3.3.1.1 The IDRP protocol exchange shall use the connectionless network service provided by ISO/IEC 8473, as specified in ISO/IEC 10747.

5.8.3.3.2 ISO/IEC 10747 Mandatory Requirements

5.8.3.3.2.1 Airborne Router

5.8.3.3.2.1.1 An ATN Airborne Router supporting the ISO/IEC 10747 Inter-Domain Routing Protocol shall support all mandatory requirements as specified in clause 12.1 of ISO/IEC 10747 with the exception of the requirements listed in Table 5.8-7, for which support is optional.

Note.— This specification deviates from ISO/IEC 10747 for Airborne Routers, in order to simplify the specification of operational equipment by removing all non-applicable requirements.

5.8.3.3.2.2 Ground Router

Note.— This section refers to both Air/Ground and Ground/Ground Routers generically as Ground Routers.

5.8.3.3.2.2.1 An ATN Ground Router supporting the ISO/IEC 10747 Inter-Domain Routing Protocol shall support all mandatory requirements as specified in clause 12.1 of ISO/IEC 10747.

5.8.3.3.2.2.2 However, over adjacencies with Airborne Routers, ATN Air/Ground Routers shall exclude the dynamic use of the following functions and features:

- a) The Next Hop Path Attribute
- b) The DIST_LIST_EXCL Path Attribute
- c) RIB Refresh Request
- d) The Residual Error Path Attribute
- e) The Expense Path Attribute
- f) The Priority Path Attribute
- g) The Transit Delay Path Attribute
- h) The Locally Defined QoS Path Attribute
- i) Hierarchical Recording
- j) Support of Partial Source Routing.

5.8.3.3.3 ISO/IEC 10747 Optional Requirements

5.8.3.3.3.1 An ATN Router shall support the Security Path Attribute as specified in 5.8.3.2.2 and 5.8.3.2.3.

5.8.3.3.3.2 **Recommendation.**— *An ATN Air/Ground Router should implement Route Aggregation and Route Information Reduction Procedures.*

5.8.3.3.3.3 **Recommendation.**— *An ATN Ground/Ground Router should implement Route Aggregation and Route Information Reduction Procedures.*

5.8.3.4 KeepAlive Timer

5.8.3.4.1 **Recommendation.**— Air/Ground Routers and Airborne Routers (i.e. Router Classes 5 and 6) should utilize initial keepAlive timer values on air/ground BIS-BIS connections as follows:

Table 5.8-8 KeepAlive Timer Values on Air/Ground BIS-BIS Connections

Router Capability	Nominal KeepAlive Value
AMSS and/or HFDL	180 minutes
Mode S and/or VDL only	30 minutes

Note 1.— Choice of nominal keepAlive timer value is based on the longest adjacency equipage.

Note 2.— The Leave Event is the primary means of reporting the loss of connectivity on air/ground adjacencies. A lost Leave Event in AMSS is trapped by the timer event, and routing tables are thus cleared.

5.8.3.4.2 **Recommendation.**— Ground/Ground Routers and Air/Ground Routers (i.e. Router Classes 4 and 5) should utilize initial keepAlive timer values in the range of 5 to 60 seconds on ground/ground BIS-BIS connections.

5.8.3.4.3 Air/Ground and Airborne Router implementations (i.e. Router Classes 5 and 6) shall implement the capability of different timer values on separate BIS-BIS connections.

Note.— ISO/IEC 10747 section 11.4 in the definition of the adjacentBISpkg-P PACKAGE requires each BIS-BIS connection to operate a separate hold and keepAlive timer.

5.8.3.5 APRLs

5.8.3.5.1 General

5.8.3.5.1.1 An implementation of the ISO/IEC 10747 protocol shall be used in ATN Routers if and only if its PICS is in compliance with the APRLs specified in the following sections.

Note.— The IDRPs requirements list is a statement of which capabilities and options of the protocol at minimum are required to be implemented for the ATN environment. The requirements list may be used by the protocol implementor as a check list to conform to this standard; by the supplier and procurer to provide a detailed indication of the capabilities of an implementation; by the user to check the possibility of interworking between two different implementations; and by the protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance to the protocol.

5.8.3.5.2 ATN Specific Protocol Requirements

Item	Description	ATN SARPs Ref	G-G Router	A/G Router	Airborne Router
ATNIDRP1	Does this BIS encode and use the Security Path Attribute?	5.8.3.2.2, 5.8.3.2.3	M	M	M

Item	Description	ATN SARPs Ref	G-G Router	A/G Router	Airborne Router
ATNIDRP2	Does this BIS immediately re-advertise routes if the security information contained in the route's security path attribute changes?	5.8.3.2.7	M	M	-
ATNIDRP3	Does this BIS support 'policy based route aggregation'?	5.8.3.2.6.2	O	O	-
ATNIDRP4	Does this BIS support 'policy based route information reduction'?	5.8.3.2.6.5	O	O	-
ATNIDRP5	Does this BIS support aggregation of routes with identical NLRI using 'true route aggregation'?	5.8.3.2.6.3	O.1	O.1	-
ATNIDRP6	Does this BIS support aggregation of routes with identical NLRI using 'route merging'?	5.8.3.2.6.3	O.1	O.1	-
ATNIDRP7	Does this BIS support aggregation of security path attribute information field?	5.8.3.2.6.4	M	M	-

5.8.3.5.3 IDRP General

Item	Description	ISO/IEC 10747 Ref.	ISO Status	G-G Router	A/G Router	Airborne Router
BASIC	Are all basic BIS functions implemented?	12.1	M	M	M	M
MGT	Is this system capable of being managed by the specified management information?	11	M	O	O	O
VER	Does this BIS support Version Negotiation?	7.8	M	M	M	M
RTSEP	Does this BIS support the ROUTE_SEPARATOR attribute?	7.12.1	M	M	M	M
HOPS	Does this BIS support the RD_HOP_COUNT attribute?	7.12.13	M	M	M	M
PATH	Does this BIS support the RD_PATH attribute?	7.12.3	M	M	M	M

Item	Description	ISO/IEC 10747 Ref.	ISO Status	G-G Router	A/G Router	Airborne Router
CAPY	Does this BIS support the Capacity Attribute?	7.12.15	M	M	M	M
FSM	Does this BIS manage BIS-BIS connections according to the BIS FSM description?	7.6.1	M	M	M	M
FCTL	Does this BIS provide flow control?	7.7.5	M	M	M	M
SEQNO	Does this BIS provide sequence number support?	7.7.4	M	M	M	M
INTG1	Does this BIS provide Data Integrity using authentication type 1?	7.7.1	O.1	M	M	M
INTG2	Does this BIS provide Data Integrity using authentication type 2?	7.7.2	O.1	O	O	O
INTG3	Does this BIS provide Data Integrity using authentication type 3?	7.7.3	O.1	O	O	O
ERROR	Does this BIS handle error handling for IDRP?	7.20	M	M	M	M
RIBCHK	Does this BIS operate in a “fail-stop” manner with respect to corrupted routing information?	7.10.2	M	M	M	M

Note.— The interpretation of the Item MGT is that mandatory compliance requires that access to the MO is provided via a Systems Management Agent. Remote Systems Management is not required for this version of the SARP and hence it is not reasonable to require mandatory support for this requirement.

5.8.3.5.4 IDRP Update Send Process

Item	Description	ISO/IE C 10747 Ref.	ISO Status	G-G Router	A/G Router	Airborne Router
INT	Does the BIS provide the internal update procedures?	7.17.1	M	M	M	O

Item	Description	ISO/IEC 10747 Ref.	ISO Status	G-G Router	A/G Router	Airborne Router
RTSEL	Does this BIS support the MinRouteAdvertisementInterval Timer (except in the case specified in ATNIDRP2)?	7.17.3.1	M	M	M	O
RTORG	Does this BIS support the MinRDOriinationInterval Timer?	7.17.3.2	M	M	M	M
JITTER	Does this BIS provide jitter on its timers?	7.17.3.3	M	M	M	M

5.8.3.5.5 IDR P Update Receive Process

Item	Description	ISO/IEC 10747 Ref.	ISO Status	G-G Router	A/G Router	Airborne Router
INPDU	Does the BIS handle inbound BISPDU s correctly?	7.14	M	M	M	M
INCONS	Does this BIS detect inconsistent routing information?	7.15.1	M	M	M	INT:O

5.8.3.5.6 IDR P Decision Process

Item	Description	ISO/IEC 10747 Ref.	ISO Status	G-G Router	A/G Router	Airborne Router
TIES	Does this BIS break ties between candidate routes correctly?	7.16.2.1	M	M	M	M
RIBUPD	Does this BIS update the Loc-RIBs correctly?	7.16.2	M	M	M	M
AGGRT	Does this BIS support route aggregations?	7.18.2.1, 7.18.2.2, 7.18.2.3	O	ATNIDRP3 or ATNIDRP5: M	ATNIDRP3 or ATNIDRP5: M	-

Item	Description	ISO/IEC 10747 Ref.	ISO Status	G-G Router	A/G Router	Airborne Router
LOCK	Does this BIS provide interlocks between its Decision Process and the updating of the information in its Adj-RIBs-In?	7.16.4	M	M	M	M

5.8.3.5.7 IDR P Receive

Item	Description	ISO/IEC 10747 Ref.	ISO Status	G-G Router	A/G Router	Airborne Router
RCV	Does the BIS process incoming BISPDU s and respond correctly to error conditions?	7.14, 7.20	M	M	M	M
OSIZE	Does this BIS accept incoming OPEN PDU s whose size in octets is between MinBISPDULength and 3000?	6.2,7.20	M	M	M	M
MXPDU	Does the BIS accept incoming UPDATE, IDR P ERROR and RIB REFRESH PDU s whose size in octets is between minBISPDULength and maxBISPDULength?	6.2,7.20	M	M	M	BISREF: OX ^BISREF: M

BISREF: if RIB REFRESH PDU **then** true **else** false

5.8.3.5.8 Peer Entity Authentication

Item	Description	ISO/IEC 10747 Ref.	ISO Status	G-G Router	A/G Router	Airborne Router
AUTH	Does this BIS correctly authenticate the source of a BISPDU?	7.7.2	O	M	M	M

Note.— Only support for an Authentication Code 1 is required.

5.8.3.5.9 IDR P CLNS Forwarding

Item	Description	ISO/IEC 10747 Ref.	ISO Status	G-G Router	A/G Router	Airborne Router
PSRCRT	Does the BIS correctly handle ISO/IEC 8473 NPDUs that contain a partial source route?	8	M	O	OX	O
DATTS	Does the BIS correctly extract the NPDUs-derived Distinguishing Attributes from an ISO/IEC 8473 NPDUs?	8.2	M	M	M	M
MATCH	Does the BIS correctly match the NPDUs-derived Distinguishing Attributes with the corresponding FIB-Atts?	8.3	M	M	M	M
EXTF	Does the BIS correctly forward NPDUs with destinations outside its own routing domain?	8.4	M	M	M	M
INTF	Does the BIS correctly forward NPDUs with destinations inside its own routing domain?	8.1	M	M	M	M

5.8.3.5.10 IDR P Optional Transitive Attributes

Item	Description	ISO/IEC 10747 Ref.	ISO Status	G-G Router	A/G Router	Airborne Router
MEXIT	Does this BIS support use of the MULTI-EXIT DISC attribute?	7.12.7	O	O	O	O

5.8.3.5.11 Generating Well-Known Discretionary Attributes

Item	Description	ISO/IEC 10747 Ref.	ISO Status	G-G Router	A/G Router	Airborne Router
EXTG	Does the BIS support generation of the EXT_INFO attribute?	7.12.2	O	O	O	O

Item	Description	ISO/IEC 10747 Ref.	ISO Status	G-G Router	A/G Router	Airborne Router
NHRS	Does the BIS support generation of the NEXT_HOP attribute in support of route servers?	7.12.4	O	O	IDRPAG: OX ^IDRPAG: O	O
NHSN	Does the BIS support generation of the NEXT_HOP attribute to advertise SNPsAs?	7.12.4	O	O	IDRPAG: OX ^IDRPAG: O	O
DLI	Does the BIS support generation of the DIST_LIST_INCL attribute?	7.12.5	O	O	O	O
DLE	Does the BIS support generation of the DIST_LIST_EXCL attribute?	7.12.6	O	O	IDRPAG: OX ^IDRPAG: O	O
TDLY	Does the BIS support generation of the TRANSIT DELAY attribute?	7.12.8	O	O	IDRPAG: OX ^IDRPAG: O	O
RERR	Does the BIS support generation of the RESIDUAL ERROR attribute?	7.12.9	O	O	IDRPAG: OX ^IDRPAG: O	O
EXP	Does the BIS support generation of the EXPENSE attribute?	7.12.10	O	O	IDRPAG: OX ^IDRPAG: O	O
LQOSG	Does the BIS support generation of the LOCALLY DEFINED QOS attribute?	7.12.11	O	OX	OX	OX

Item	Description	ISO/IEC 10747 Ref.	ISO Status	G-G Router	A/G Router	Airborne Router
HREC	Does the BIS support generation of the HIERARCHICAL RECORDING attribute?	7.12.12	O	OX	OX	OX
SECG	Does the BIS support generation of the SECURITY attribute?	7.12.14	O	M	M	M
PRTY	Does the BIS support generation of the PRIORITY attribute?	7.12.16	O	O	IDRPAG: OX ^IDRPAG: O	O

IDRPAG: if Air/Ground adjacency **then** true **else** false

5.8.3.5.12 Propagating Well-Known Discretionary Attributes

Item	Description	ISO/IEC 10747 Ref.	ISO Status	G-G Router	A/G Router	Airborne Router
EXTGP	Does the BIS support propagation of the EXT_INFO attribute?	7.12.2	M	M	M	-
NHRSP	Does the BIS support propagation of the NEXT_HOP attribute in support of route servers?	7.12.4	O	O	IDRPAG: OX ^IDRPAG: O	-
NHSP	Does the BIS support propagation of the NEXT_HOP attribute to advertise SNPs?	7.12.4	O	O	IDRPAG: OX ^IDRPAG: O	-
DLIP	Does the BIS support propagation of the DIST_LIST_INCL attribute?	7.12.5	O	M	M	-
DLEP	Does the BIS support propagation of the DIST_LIST_EXCL attribute?	7.12.6	O	M	IDRPAG: OX ^IDRPAG: M	-

Item	Description	ISO/IEC 10747 Ref.	ISO Status	G-G Router	A/G Router	Airborne Router
TDLYP	Does the BIS support propagation of the TRANSIT DELAY attribute?	7.12.8	O	O	IDRPAG: OX ^IDRPAG: O	-
RERRP	Does the BIS support propagation of the RESIDUAL ERROR attribute?	7.12.9	O	O	IDRPAG: OX ^IDRPAG: O	-
EXPP	Does the BIS support propagation of the EXPENSE attribute?	7.12.10	O	O	IDRPAG: OX ^IDRPAG: O	-
LQOSP	Does the BIS support propagation of the LOCALLY DEFINED QOS attribute?	7.12.11	O	OX	OX	-
HRECP	Does the BIS support propagation of the HIERARCHICAL RECORDING attribute?	7.12.12	O	OX	OX	-
SECP	Does the BIS support propagation of the SECURITY attribute?	7.12.14	O	M	M	-
PRTYP	Does the BIS support propagation of the PRIORITY attribute?	7.12.16	O	O	IDRPAG: OX ^IDRPAG: O	-

5.8.3.5.13 Receiving Well-Known Discretionary Attributes

Item	Description	ISO/IEC 10747 Ref.	ISO Status	G-G Router	A/G Router	Airborne Router
EXTR	Does the BIS recognise upon receipt the EXT_INFO attribute?	7.12.2	M	M	M	M
NHRSR	Does the BIS recognise upon receipt the NEXT_HOP attribute ?	7.12.4	M	M	M	O
DLIR	Does the BIS recognise upon receipt the DIST_LIST_INCL attribute?	7.12.5	M	M	M	M
DLER	Does the BIS recognise upon receipt the DIST_LIST_EXCL attribute?	7.12.6	M	M	M	O
TDLYR	Does the BIS recognise upon receipt the TRANSIT DELAY attribute?	7.12.8	M	M	M	O
RERRR	Does the BIS recognise upon receipt the RESIDUAL ERROR attribute?	7.12.9	M	M	M	O
EXPR	Does the BIS recognise upon receipt the EXPENSE attribute?	7.12.10	M	M	M	O
LQOSR	Does the BIS recognise upon receipt the LOCALLY DEFINED QOS attribute?	7.12.11	M	O	O	O
HRECR	Does the BIS recognise upon receipt the HIERARCHICAL RECORDING attribute?	7.12.12	M	M	M	O
SECR	Does the BIS recognise upon receipt the SECURITY attribute?	7.12.14	M	M	M	M
PRTYR	Does the BIS recognise upon receipt the PRIORITY attribute?	7.12.16	M	M	M	O

IDRP Timer

Item	Description	ISO/IEC 10747 Ref.	ISO Status	G-G Router	A/G Router	Airborne Router
Ta	KeepAlive timer	11.4 ATN SARP Ref: 5.8.3.4	M	M	M	M
Tr	Retransmission (tr) timer	7.6.1.2, 7.6.1.3	M	M	M	M
Tmr	maxRIBIntegrityCheck timer	7.10.2	M	M	M	M
Tma	MinRouteAdvertisement timer	7.17.3.1	M	M	M	O
Trd	MinRDOriGinationInterval timer	7.17.3.2	M	M	M	M
Tcw	closeWaitDelay timer	7.6.1.5	M	M	M	M

5.9 Systems Management Provisions

5.9.1 Introduction

5.9.1.1 **Recommendation.**— *ATN managed resources should be grouped into Management Domains in order to assign responsibility for control of the resources.*

5.9.1.2 **Recommendation.**— *States and Organizations should assign an administrative authority to establish and maintain the respective management of each of their Management Domains, and to manage the transfer of control of resources from one Management Domain to another.*

Note.— *The definition and implementation of a global ATN Systems Management solution may be specified in future amendments to these SARPs. Currently:*

- a) *No exchange of Systems Management information is required between routers of different Administrative Domains.*
- b) *No exchange of Systems Management information is required by means of a management protocol over the Air/Ground links. This does not preclude the exchange of routing information, by means of routing information exchange protocols.*
- c) *The exchange of Systems Management information within an Administrative Domain is considered a local matter and can be achieved by any means deemed appropriate.*

— END —