dications. The former cause the window to shrink to a single segment, while the latter prevent the ATM rate from increasing. Furthermore there might be simultaneous retransmissions which can cause congestion to occur again, if the guaranteed sources are still sending a high volume of cells. The consequence is a fluctuating behavior not only of the goodput but even of the link utilization itself, as shown in fig. 6. The horizontal axis is the the sampling interval index (each interval is 1000 cell transmission slots long). Within the window of time plotted the utilization trace is oscillating for PRCA, though the traffic offered by the TCP connections is enough to fill the residual bandwdith, while FCVC achieves full utilization.

The TCP goodput achieved by FCVC is very close to a theoretical bound, plotted in fig. 5 by considering the average residual bandwidth B_{av} and the fraction η_{OH} of ATM bandwidth available to the application. This fraction, limited by the protocols overhead, is computed analytically in the following as a function of MSS and the distribution of the length of the application messages.

For each TCP segments a total of 48 bytes, including TCP and IP headers and a AAL-CS trailer, is added. The CS-PDU is then segmented in 48 bytes payloads and finally the 5 bytes ATM headers are added. Two integer-valued functions are defined:

$$\begin{aligned} N_{seg}(x) &= n \quad \textit{if} \quad (n-1)*MSS < x \le n*MSS \\ N_{cells}(x) &= m \quad \textit{if} \quad 48*(m-1) < x \le 48*m \end{aligned}$$

$N_{seg}(x)$ is the number of TCP segments generated by an application message of size x, whereas $N_{cells}(x)$ is the number of ATM cells generated by a CS-PDU of size x. Then the average number of cells generated by an application message can be computed as:

$$\begin{aligned} \overline{N_{cells}} \;&=\; \int_0^\infty [N_{cells}(MSS+48)(N_{seg}(x)-1)+ \\ &\quad N_{cells}(x-MSS\;(N_{seg}(x)-1)+48)]\;f_L(x)dx \end{aligned}$$

where $f_L(x)$ is the probability density function of the application message length, which has been assumed exponential with average length L, as already mentioned in B. Finally:

$$\eta_{OH} = \frac{L}{53\,\overline{N_{cells}}} \tag{2}$$

Computing the theoretical goodput as $\gamma_t = \eta_{OH}\,B_{av}$, we can see in fig. 5, that only for very short segments the FCVC curve slightly diverges from the maximum achievable goodput, because of some occurences of unnecessary retransmissions.

V. Conclusion and open issues

In this paper we have presented an unbiased comparison of the two most debated congestion control schemes for ABR services in ATM LANs running TCP/IP. FCVC features outstanding performances, although it requires a complex and costly switch architecture, which makes it hardly feasible. However, it sets an upper limit of effectiveness, which PRCA should try to approach, without requiring many additional functionalities (e.g by appropriately tuning *MDF* and *AIR*). The criteria to be used for this choice should be simple enough to be implemented run-time. Second, a smart way of discarding cells when congestion occurs, can avoid synchronized retransmissions and limit the throughput instability [RF94].

The same comparison should also be performed with larger networks and more complex topologies, where FCVC effectiveness can be limited, because it would be more affected by useless and wasteful TCP retransmissions.

All these issues are likely to be subject of further studies.

References

[BL90] R. Bagrodia and W. Liao. Maisie: A language and optimizing environment for distributed simulation. In *1990 Simulation Multiconference: Distributed Simulation*, San Diego (CA), January 1990.

[Com91] D. E. Comer. *Internetworking with TCP/IP*, volume vol. I. Prentice Hall, 1991.

[ea94] M. Hluchy et al. Closed loop rate-based traffic management. Technical report, ATM Forum, September 1994.

[GPS95] M. Gerla, C.M.D. Pazos, and V.A. Signore. Simulation study of PRCA and FCVC in ATM LANs supporting TCP/IP. Technical Report 950021, UCLA-CSD, June 1995.

[Jac88] V. Jacobson. Congestion avoidance and control. In *Proc. of SIGCOMM '88*, Stanford, CA, August 1988. ACM.

[KC94] H. T. Kung and Alan Chapman. The FCVC (Flow-Controlled Virtual Channels) proposal for ATM networks. Technical report, January 1994. ATM Forum report.

[KC95] H. T. Kung and K. Chang. Receiver-oriented adaptive buffer allocation in credit-based flow control for atm networks. To appear in INFOCOM '95, April 1995.

[RF94] Allyn Romanow and Sally Floyd. Dynamic of TCP traffic over ATM networks. In *ACM SIGCOMM '94*, August 1994.

[SNS+89] H. Suzuki, H. Nagano, T. Suzuki, T. Takeuchi, and S. Iwasaki. Output-buffer switch architecture for asynchronous transfer mode. In *IEEE International Conference on Communications*, Boston (USA), June 1989.

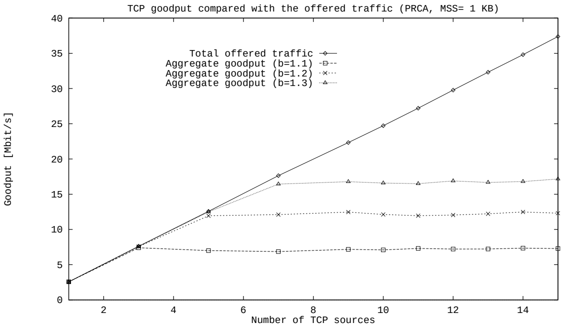

Figure 3: Total TCP goodput with variable b_g (PRCA).

for different values of the guaranteed sources burstiness and consequently of the bandwidth available on the common link.

We use the values $b_g = 1.1, 1.2, 1.3$, typical for almost CBR applications, and the TCP maximum segment size is $MSS = 1024\ bytes$. At ATM level the corresponding values of the average available bandwidth are $B_{av} = 13.6, 25, 34.6\ Mbit/s$ respectively. Both plots report also the total traffic the TCP applications generate. It is evident that all curves can be divided in an uncongested region, where the goodput follows the offered traffic, and a congested region, where the goodput reaches a "saturation" value, corresponding to the situation of a fully utilized link and as expected increases with b_g. However it is clear that all those saturation levels do not match the corresponding values of available bandwidth. Indeed PRCA achieves approximately $\gamma = 7.3, 12.2, 16.6\ Mbit/s$, for the respective values of b_g, whereas FCVC gets $\gamma = 9.6, 18.9, 27.4\ Mbit/s$. Part of the bandwidth available at ATM level is unusable by the TCP applications because of the intermediate protocols overhead and retransmissions triggered by sudden increases in the round-trip delay, even without actual losses. Moreover with PRCA, ATM cells can be lost and the need arises for segment retransmissions. This extra burden explains the gap of performances between PRCA and FCVC.

As we pointed out above, the limitation in the effective throughput of TCP applications is caused partly by the underlying protocols overhead. This consideration is stressed in fig. 5.

It is clear that the goodput increases with MSS for FCVC because the protocol overhead is reduced. With PRCA the situation is different. Although larger segment sizes reduce the protocol overhead, with PRCA, cells losses trigger TCP retransmissions, and the retransmission of very large segments is wasteful as well. Thus, with PRCA the goodput is not an increasing function of MSS: it reaches a maximum around $MSS = 2048\ bytes$, and decreases again for larger MSS. The gap in goodput is caused by the fact that, unlike FCVC, for PRCA there is no possibility of promptly stopping the ABR sources when there is a sudden burst from high priority connections. As a result, TCP connections can be simultaneously affected by cells losses and congestion in-

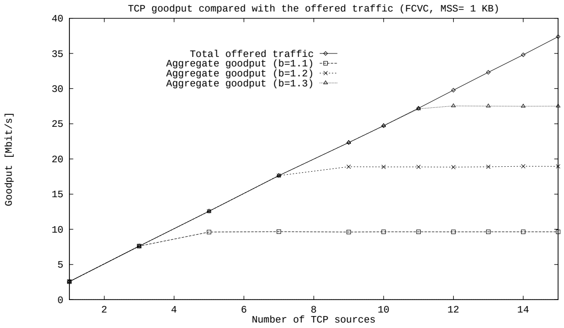

Figure 4: Total TCP goodput with variable b_g (FCVC).

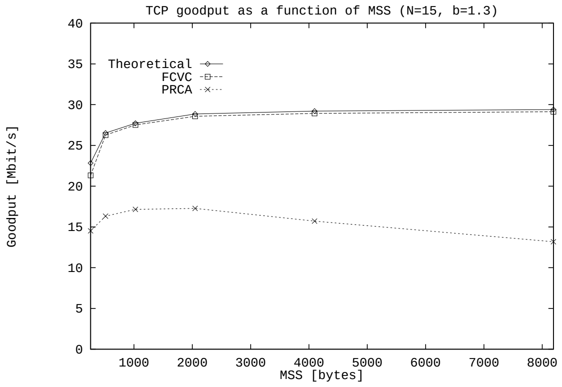

Figure 5: TCP goodput as a function of MSS.

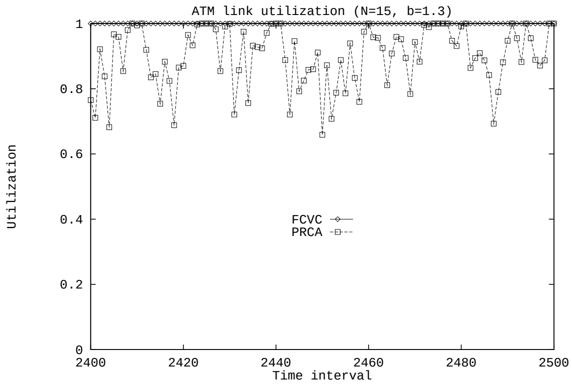

Figure 6: ATM link utilization trace

A. Guaranteed traffic generator

For the sake of simplicity we assume the guaranteed traffic comes through a single input link. It is generated as an MMDP process resulting from aggregating VBR sources. Each source has the following characteristics:

- peak rate $B_{pg} = 10\ Mbit/s$ at the user level;
- mean burst length $L_g = 4.8\ Kbytes$;

The number of sources $N_g = 14$ has been chosen such that, when they are all simultaneously active, no residual bandwidth is left on the common link. By varying the value of b_g, it is possible to get the desired average bandwidth B_{ag}. The instant value of the bandwidth available to TCP sources varies as time goes by, because of the variations of the number of simultaneously active gururanteed sources. By doing so we can study which scheme allows for a better fill in.

B. ATM workstations

A client application, represented by an ON-OFF model with exponential active and idle periods, runs over TCP/IP. The main traffic parameters are:

- average message length $L = 2.5\ Kbytes$;
- average bandwidth $B_a = 2.5\ Mbit/s$, at the user level;

The number N_{tcp} of applications, and consequently of required TCP connections, is varied from 1 to 15, such as to simulate different levels of congestion. All TCP connections are kept opened throughout the whole simulation (20 seconds of actual time).

The TCP level at the source site performs window based flow control functions as described in [Jac88]. The application byte stream is segmented in TCP segments of up to MSS bytes, with a 20 bytes header. Each successfully transmitted segment is acknowledged by the destination side TCP protocol. Upon receiving a positive acknowledgment the source TCP increases the window size by MSS if it is in the slow-start phase, whereas in the congestion avoidance phase it is increased only when all the segments in the current window have been acknowledged. In case of a segment timer expiration, the window size is reduced to MSS. Timer are set based on round trip estimations according to Karn's algorithm [Com91]. In our simulation, timers and round-trip estimations do not include transmission times, meaning that the reference instant is the time of a segment transmission completion. Although this choice may not be feasible for real TCP implementations, it makes the round-trip estimations independent of the segment sizes, allowing for a better representation of the delay, and consequently of the level of congestion inside the network.

For our purposes the IP level performs minor functions. A 20 bytes header is added to each TCP segment.

The AAL level is based on the AAL5 protocol. An 8 bytes CS-PDU trailer is added to each IP packet. AAL and ATM levels perform also operations related to the specific congestion control scheme.

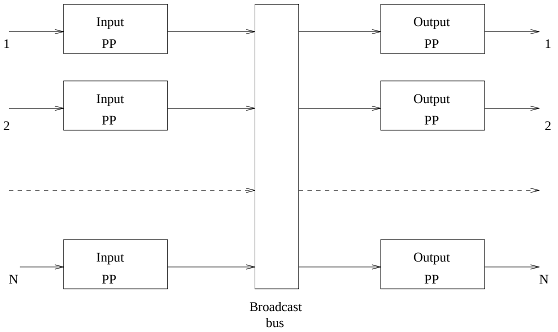

Figure 2: Basic elements of the simulated ATOM switch.

C. ATOM switch

The simulation of the ATOM switch is based on three MAISIE entities as described in fig. 2. The Input Packet Processor performs ATM cell header processing and tags cells for internal routing. The Broadcast Bus routes cells at speed N times the link speed (up to N cells can be switched per time unit). Cells are queued in the appropriate output buffer, and scheduled for trasmission by the Output Packet Processor. For both FCVC and PRCA a double priority buffering is used. ABR cells are taken from the apposite buffer only when the guaranteed cells buffer is empty. Since all guaranteed cells are assumed to come through a single incoming link, a single cell buffer is enough. The ABR buffer size is 165 cells. The link propagation delay, including processing time, is 70 μs.

D. Protocol specific assumptions: FCVC

With FCVC the buffer is divided into up to 15 VC buffers of 11 cells each. The VC buffer size allows to store $N_3 = 1$ cells with the link propagation delay of 70 μs at a targeted user bandwidth of $B_{VC} = 2.5\ Mbit/s$ (see eq. 1), and to send a credit cell every $N_2 = 10$ data cells sent in the downstream direction. Best effort cells are scheduled for transmission according to a simple round-robin policy among all the active VCs.

E. Protocol specific assumptions: PRCA

With PRCA the buffer is shared by all best-effort VCs. The congestion threshold is set to 50% of the buffer size. The parameters are chosen such as to sustain a user rate of 2.5 Mbit/s when the network is not congested (all RM cells carry a no congestion indication). In detail: $PCR = 12\ Mbit/s$, $MCR = 500\ Kbit/s$, $ADR = ACR/2^{MDF}$ (when a positive RM cell is received) and $MDF = 5$, $NRM = 32$, $AIR = 3.2\ Mbit/s$.

IV. Simulation results

Our major concern is the effectiveness of the ABR schemes under comparison, expressed by means of the TCP sources goodput. More results concerning fairness and responsiveness of the same techniques are reported in [GPS95].

A first set of results is presented in fig. 3 and fig. 4 showing the goodput γ as a function of the number of TCP sources hanced version of this algorithm, enriched with a bandwidth advertising mechanism, is described in [ea94].

B. Flow Controlled Virtual Circuit

The algorithm is conceptually straightforward. Each node (including the destination interface) has an allocated buffer for every crossing VC. The size of a single buffer is $N_2 + N_3$, where: N_2 is the amount of cells the node has to forward before sending a credit cell back to the previous node along the considered VC; N_3 is the amount of cells that could arrive within a link Round-Trip Time RTT at a targeted bandwdith B_{VC}. If N_{bits} is the number of bits in an ATM cells ($53*8 = 424$ bits/cell):

$$N_3 = \left\lceil \frac{RTT * B_{VC}}{N_{bits}} \right\rceil \qquad (1)$$

The node keeps a credit count for each VC, which decreases by one every time a cell is transmitted. Of course a cell transmission is possible only when the credit count is non null. Moreover it keeps a count C of the cells sent through each VC. This value is inserted in an apposite field of the credit cell, when sent back to the upstream node. Upon receiving a credit cell the upstream node updates its own credit count summing $C_{new} - C_{old}$ to the current value, where C_{new} and C_{old} are the respective values of C when the current and last credit cells are sent. An enhanced version of this scheme, featuring an adpative buffer allocation technique, is described in [KC95]

C. FCVC versus PRCA: implementation issues

Although the link-by-link flow control appears conceptually simple, the implementation complexity is significant. Each node needs to keep track of the amount of cells transmitted through every VC crossing it, in order to let the upstream node update its credit balance upon receiving the credit cell. Memory management is probably the most negative aspect of the original FCVC. The switch controller has to handle independently several separated buffers, which have in general different sizes in an heterogeneous traffic environment. The amount of memory is proportional to the link propagation delays leading to very large buffers requirements in the wide area scenario and no scalability of the switch design. Finally, the static buffer allocation is probably wasteful for high burstiness traffic. In conclusion the implementation seems to be feasible only in ATM LANs with short propagation delays and a reasonably low number of end-to-end VCs.

On the other hand PRCA has a far lower impact on the ATM switch architecture, the only problem being the parameter tuning at connection setup time. Indeed most of the parameters (PCR, AIR, ADR) are meant to be negotiated and defined during the signalling procedure, even though for connectionless best-effort traffic not enough information can be specified a priori by the connection originator. Anyway, the same problem may appear with FCVC for the definition of the VC targeted bandwidth.

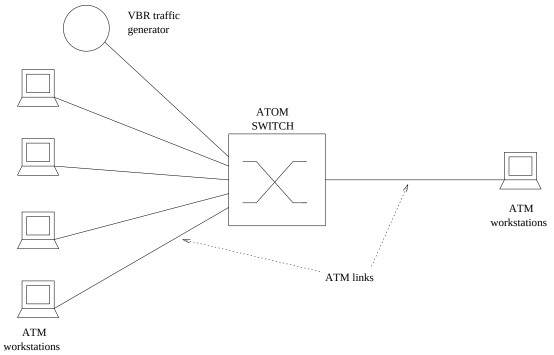

Figure 1: Scheme of the ATM LAN model.

D. FCVC versus PRCA: interaction with TCP

FCVC guarantees no cell losses and low priority best-effort cells are backpressured in case of congestion and used to fill in the "holes" in the transmission stream of high-priority guaranteed traffic. Thus, FCVC is expected to show high effectiveness.

PRCA does not insure zero losses and effective bandwdith fill-in can be achieved if the rate adjustment algorithm quickly converges to a fair share of the residual bandwidth, the shorter the propagation delays the faster the convergence. However, with PRCA the link utilization itself might not be a meaningful measure of effectiveness. Indeed, if TCP/IP is supported, cell losses trigger TCP segment retransmission, and a portion of bandwdith is wasted along the path for unsuccessful transmissions. Moreover, since many TCP sources are likely to be simultaneously affected by cell losses in case of congestion, deep throughput ripples can occur if synchronized retransmission are not avoided. For this reason defining an appropriate TCP Maximum Segment Size (MSS) can be a critical issue with PRCA. With small segments the TCP/IP header overhead can be remarkable, but increasing the segment size can be harmful because of the need of retransmitting large amounts of data. On the contrary FCVC can take more advantage of an increase of the MSS, because TCP retransmissions are in general not required.

In the following, FCVC and PRCA are compared through the simulation of an ATM LAN supporting TCP/IP. The main performance measure is the goodput, that is the throughput as seen by the application on top of TCP.

III. Simulation of an ATM LAN: assumptions

The model of the ATM LAN used in our simulation program, written in MAISIE language [BL90], is based on a single switch (fig. 1). We use the ATOM [SNS$^+$89] architecture as a model for the basic switching functions and specific features are added at ATM and AAL levels in order to support either PRCA and FCVC congestion controls. All traffic sources share a single outgoing link, where bandwidth is allocated only to guaranteed sources, while best-effort sources are competing for the residual bandwidth. A more detailed description of each element of the simulator follows.

Comparing ATM Credit-Based and Rate-Based Controls for TCP Sources*

Mario Gerla, Carlos M. D. Pazos†
UCLA-Computer Science Departement
Los Angeles, CA 90024
E-mail: gerla,pazos@cs.ucla.edu

Vincenzo A. Signore‡
Politecnico di Bari - DEE
Via Orabona 4, Bari, Italy
E-mail: signore@cs.ucla.edu

***Abstract*—**

TCP/IP traffic is likely to be one of the most heavily carried over ATM networks. However this kind of traffic is of connectionless nature and for this reason will be carried on an Available Bit Rate (ABR) basis.

In order to prevent the ABR traffic from congesting the network, special congestion control techniques have been devised. These schemes can be categorized in two major groups: rate-based and credit-based schemes. The most important representatives of these categories are respectively the PRCA and the FCVC. Since TCP/IP traffic is also controlled by the TCP window flow control, in this paper we present a comparison between PRCA and FCVC under TCP traffic. In this sense the performance measure we are mostly concerned with is the goodput. We investigate the impact of the burstiness of the guaranteed sources used as background traffic and the maximum size of TCP segments, which seems to be critical for the PRCA approach.

I. Introduction

It is likely that the next generation of LANs will be based on an ATM subnetwork and support TCP/IP protocols, in order to guarantee the compatibility with existing LAN services. Hence, the problem facing us is how to carry the TCP/IP connectionless traffic on an ATM network, without disrupting the other, possibly multimedia, connection oriented traffic. Namely, since IP is a connectionless protocol, while ATM is a connection oriented network, the ATM resources negotiation phase at call setup time is unfeasible for TCP/IP connections. ATM interfaces supporting TCP/IP based applications are unable to declare their own traffic parameters, thus no QoS can be guaranteed. As a result, such services are to be included in the ABR class, meaning that they do not have reserved resources, but they are allowed to use the bandwidth left available by high-priority CBR and VBR sources along the path toward the destination. Given this scenario, it is of great importance to understand how ATM congestion control schemes for ABR services are affected by the TCP window flow control. In this paper credit-based and rate-based schemes are compared via simulation of an ATM LAN, with ATM workstations running a sample TCP/IP application and background traffic generated by VBR sources.

II. Congestion control schemes for ABR services

Recently, the two congestion control schemes which have most attracted the attention of the ATM Forum are the PRCA (Proportional Rate Control Algorithm) [ea94], which is rate-based, and FCVC (Flow Controlled Virtual Circuit) [KC94], a credit based scheme. A detailed description of these two algorithms follows.

A. Proportional Rate Control Algorithm

The basic idea is that the source is allowed to increase its rate only when it receives an explicit indication of no congestion, otherwise it keeps additively decreasing the rate by ADR (Additive Decrease Rate) after every cell transmission, down to the minimum value MCR (Minimum Cell Rate). ADR depends on a negotiated Multiplicative Decrease Factor (MDF). The network is assumed to be segmented into domains each delimited by virtual or actual source-destination couples. The domain end points run a closed management loop: every cell is sent with the Explicit Forward Indication (EFCI) bit set to zero, and eventually will be marked by one of the intermediate switches in case of congestion.

Each domain chooses its way to detect congestion. One of the traditional ways is monitoring the buffer length and marking cells when a predefined threshold is exceeded. After NRM data cells, a Resource Management cell is sent by the source in order to probe the domain. A Congestion Indication (CI) bit, provided in each RM cell, is set by the destination if the last data cell was received marked, or left zero otherwise, and the RM cell is sent back to the source. Upon receiving a RM cell with no congestion indication, the source is allowed not only to compensate for the rate decrease in the last cycle, but also to increase it further by an agreed upon Additive Increase Rate (AIR), without exceeding a negotiated Peak Cell Rate (PCR). If the congestion bit is set, the source continues reducing the rate. An en-

---

[*] This work was supported by NSF through grants NSF 9305376 and NSF-INT 9115882.

[†] Bolsista do CNPq – Brasilía/Brasil.

[‡] This work was performed while V.A. Signore was visiting the CSD of UCLA.