Ph.D. Qualifying Reading Lists

****

CS 6352 – Performance of Computer Systems and Networks

List of topics:

Properties of Poisson streams of customer arrivals.
Analysis and performance figures of the M/M/1 queue.
Continuous parameter Markov chains.
Single state dependent (continuous time) Markovian queueing systems.
Various applications of such state dependent cases in computer systems and data communication networks.
Generalized Little's result for multiple non-FIFO queues.
Development and analysis of Markov chains for simple priority queues.
Development of Pollaczek-Khinchin mean value formula for the M/G/1 queue. Applications.
Development of discrete parameter Markov chains for discrete time queues.
Analysis of discrete parameter Markov chains.
Evaluation of performance figures.
Applications of discrete time queues in computer systems and data networks (such as, for examples, cross-bar and simple multistage switches).
Product form solutions for networks of continuous time open and closed Markovian queues (unlimited buffer, state independent service rates). Convolution algorithm and Mean Value Analysis techniques for such closed queuing networks.

Type of questions:

Questions will be combinations of theoretical development, analysis of given systems, development of appropriate models and follow up analysis starting from verbal descriptions of physical systems. In most cases, students should attempt to solve problems from fundamental principles rather than trying to remember and apply formulae for various special cases. A set of helpful formulae, etc. (such as the Pollaczec-Khinchin mean value formula and the MVA algorithm) will be supplied along with the question paper.
The following list of references include the commonly used text book, other reference books on queues, and a sample of books on Probability Theory. Students are responsible for correcting errors in the reference material.

Text Book:

T. G. Robertazzi, Computer Networks and Systems: Queueing Theory and Performance Evaluation. Springer, 2000.

Other References:

D. Gross and C. M. Harris, Fundamentals of Queueing Theory. Wiley, 1997.

L. Kleinrock, Queueing Systems, Volume 1, Theory. Wiley, 1975.

J. J. Higgins and S. Keller-McNulty, Concepts in Probability and Stochastic Modeling. Duxbury Press, 1995.

K.S. Trivedi, Probability and Statistics with reliability, Queueing, and Computer Science Applications. First Edition or Second Edition (2001, Wiley)

C. M. Grinstead and and J. L. Snell, Introduction to Probability. American Mathematical Society, 1997.

****

CS 6360 – Database Design

Textbook: "Fundamentals of database systems (Fifth edition)"
by Elmasri and Navathe:

Chapter/Sections	Topics
1	Introduction
2	Data models, DB architecture, classification
3	Entity relationship model, ER diagrams (exclude 3.8)
4	EER model (exclude 4.6, 4.7)
5	Storage organization
7	Relational model, Relational algebra
8	SQL (exclude 8.7)
10	
15	Algorithms for RDB design
17	(exclude 17.6)
18.1-2	Query processing and optimization
19	Transaction processing (exclude 19.5)

Look on Google for Topical Information

****

CS 6361 – Requirements Engineering

Topics:

- Requirements Engineering: Introduction
  - Why RE? – error propagation in software lifecycle, cost and size of requirements errors, aims and scope
  - What is RE? – What are requirements? Role of requirements and requirements engineers
  - How to do RE? – types of errors, formal vs. semi-formal vs. Informal
- Requirements Engineering Processes
  - Why process? A framework for RE process: Elicitation, specification, validation, RE process and software lifecycle models
- Requirements Elicitation: Part I
  - Why is it difficult? What to elicit? How to elicit?
- Requirements Elicitation: Part II
  - Advanced goal-directed strategy, knowledge acquisition, data/information elicitation techniques
- Scenario Analysis
  - Use cases, episodes, scripts, cycle of natual inquiry, abstract vs. concrete scenario, scenario space
- Requirements Analysis, Modelling and Specification: Review
  - Conceptual modeling perspective of basic RE process, carving the product space
- Object-Oriented Modeling:
  - Intelletual origins, conceptual modeling, UML overview
  - Enterprise Requirements & Functional Requirements: Structural Requirements
  - Agent-oriented approach to enterprise modeling, ERD, i*, JSD, SADT, IDEF
- Functional Requirements: A Formal OO-RML/Telos epistemological primitives, ontological primitives, interval calculus, axiomatization of OO
- Functional Requirements: Behavioral Requirements
  - Decision-oriented behavioral models, State-oriented Behavioral models (Finite State Machines, StateCharts, PetriNets), Function-oriented behavioral models.
- Non-Functional Requirements
  - Types of NFRs, classification schemes, Process-oriented approach, Product-oriented approach, Portability, Reliability, Efficiency, Usability, Security

Main reference: Lecture Notes at
http://www.utdallas.edu/~chung/RE/contents.html

Additional References:

Articles:

Axel van Lamsweerde,"Requirements engineering in the year 00: a research perspective", Proc., Int. Conf. on Software Engineering (ICSE) 2000, pp. 5-19.

Axel Van Lamsweerde", "Goal-Oriented Requirements Engineering: A Guided Tour", Proc., Int. Symposium on Requiremens Engineering (RE'01), pp.249-261.

Sol Greenspan, John Mylopoulos, Alex Borgida, "On Formal Requirements Modeling
Languages: RML Revisited", Proc., Int. Conf. on Software Engineering (ICSE) 1994, pp. 135-147.

Mike Wooldridge and Nick Jennings, "Software Engineering with Agents: Pitfalls and Pratfalls," IEEE Internet Computing, 3(3):20-27, May/June 1999.

A. Tveit, "A Survey of Agent-Oriented Software Engineering", Proc. of the First NTNU CSGS Conference, May, 2001
URL = http://www.jfipa.org/publications/AgentOrientedSoftwareEngineering/

Anton, A.I.; Potts, C. "The use of goals to surface requirements for evolving systems", Proc., Int. Conf. on Software Engineering (ICSE) 1998, pp. 157 -166
J.M. Wing, "A Specifier's Introduction to Formal Methods," IEEE Computer, 23(9):8-24, September 1990.

J.A. Hall, "Seven Myths of Formal Methods," IEEE Software, 7(5):11-19, September 1990.

J.P. Bowen and M.G. Hinchey, "Seven More Myths of Formal Methods," IEEE Software, 12(4):34-41, July 1995.

Books:

A. M. Davis, Software Requirements: Objects, Functions, & States Prentice Hall: Englewood Cliffs, 1993.

P. Loucopoulos and V. Karakostas, System Requirements Engineering, McGraw-Hill, 1995.

M. Jackson and T. DeMarco, Software Requirements and Specifications, Addison-Wesley, 1995.

R. H. Thayer and M. Dortman, Software Requirements Engineering: 2nd edition, IEEE Computer Society Press, 1998.

I. Sommerville and P. Sawyer, Requirements Engineering - A Good Practice Guide, Wiley, 1997.

D. Gause and G. Weinberg, Exploring Requirements, Dorset House, 1989.

J. Martin and J. Odell, Object-Oriented Methods: A Foundation, Prentice-Hall, 1995.

J. Rumbaugh, I. Jacobson and G. Booch, The Unified Modeling Language Reference
Manual, Addison-Wesley, 1998.

L. Chung, B. Nixon, E. Yu and J. Mylopoulos, Non-Functional Requirements in Software Engineering, Kluwer Academic Publishing, 2000.

****

CS 6362 – Software Architecture and Design

Topics
Introduction to Software Architecture Classical
Module Interconnection
Languages Abstract DataTypes
and Objects Module
Decomposition Issues
Data Flow
Repositorie
s Events
Process Control
JavaBeans
Client Server
Middleware : CORBA, OLE/DCOM, J2EE/J2ME, .Net Patterns
Main reference: Lecture Notes at
http://www.utdallas.edu/~chung/SA/contents.html

Articles:

Mary Shaw, Paul Clements, "The Golden Age of Software Architecture," *IEEE Software*, Vol.23, no. 2, pp. 31-39, Mar./Apr.2006

Paul Clements, Mary Shaw, ""The Golden Age of Software Architcture" Revisited, " IEEE Software, vol. 26, no. 4, pp. 70-72, July/August, 2009.

Hofmeister, C., Kruchten, P., Nord, R., Obbink, H., Ran, A., & America, P. (2207). A General Model of Software Architecture Design derived from Five Industrial Approaches, Journal of Systems & Software, 80(1), 106-126.

Hassan Gomma, "Advances in Software Deisgn Methods for Concurrent, Real-Time and Distrcubuted Application," in proceedings The Third International Conference on Software Engineering Advances, 2008, pp. 451-456.

Kendall Scott, The Unified Process Explained. ISBN 0-201-74204-7, 2002, best practices in Architecture and Design.

Advanced Design Patterns.Re-use
M-A. Laverdiere; A. Mourad; A. Hanna; M. Debbabi; "Security Design Patterns: Survey and Evaluation", in Canadian Conference on Electrical and Computer Engineering, May 2006, pp. 1605 – 1608

David Kalinsky, "design Patterns for High Availability", March 13, 2003
URL: http://www.eetimes.com/story/OEG20020729S0030

Ashraf Armoush, Falk Salewski, Stefan Kowaleski, "Design Pattern Representation for Saftey-Critical Embedded Systems", JSEA

Glen B. Alleman, "Exception Handling in CORBA Environments, The Late Introduction of Distributed Exception Handling in JAVA™, CORBA-Based COTS Application Domains", 2000.

Representation

Nenad Medvidovic. Modeling software architectures in unified modeling language. *ACM Transactions on Software Engineering and Methodology*, 11(1):2-57, January 2002.

Nenad Medvidovic and Richard N. Taylor. A classification and comparision framework for software architecture description language. *IEEE Transactions on Software Engineering*, 26(1):70-93, January 2000.

P. Clements, Comparing the SEI's Views and Beyond Approach for Documenting Software Aechitecutres with ANSI-IEEE 1471-2000

Grady Booch, James Rumbaugh, and Ivar Jacobson, Unified Moedling Language User Guide, (2nd Edition), Advanced UML topics, Sections 3, 5, and 6.

Mugurel T. Ionita, Henk Obbink and Dieter Hammer, Scenario-Based Architecture Evaluation Methods: An overview, Internationl Conference on Software Engineering 2002 (ICSE'02), Orlando, Florida.

Hans-Peter Hoffmann, UML 2.0-Based Systems Engineering Using a Model-Driven Development Approach, 2005, available at http://www.stsc.hill.af.mil/crosstalk/2005/11/0511Hoffman.pdf

Michel, M.M. Gala-Edeen, G.H., "Detecting inconsistencies between software architecture views", in proceedings International Conference on Computer Engineering & Systems, 2009.
ICCES 2009. pp. 429-434

Pengcheng Zhang, Henry Muccini and Bixin Li, "A classification and comparision of model checking software architecture techniques", 2009.

Jeannette M. Wing and Mandanna Vaziri-Farahani, "A case study in model checking software systems", Science of Computer Programming 28 (1997) pp. 273-299.

Books:

Mary Shaw and David Garlan, Software Architecture: Perspectives on an Emerging Discipline, Prentice Hall, 1996.

L. Bass, P. Clements & R. Kazman, Software Architecture in Practice, Addison Wesley, 1998.

A. W. Brown (Editor), Component-Based Software Engineering, IEEE Computer Society, 1996.

Eric Gamma, Richard Helm, Ralph Johnson and John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Eric Gamma, Richard Helm, Ralph Johnson and John Vlissides, Addison-Wesley, 1994.

Wolfgang Pree, Design Patterns for Object-Oriented Software Development, Addison-Wesley Longman, 1995.

I. Singh, B. Stearns, M. Johnson, The Enterprise Team, Designing Enterprise Applications with the J2EE Platform, 2/E, Addison Wesley & Benjamin Cummings, 2002.

Randy Otte, Paul Patrick and Mark Roy, Understanding CORBA: The Common Object Request Broker Architecture, Prentice Hall, 1996.
Robert Orfali, Dan Harkey and Jeri Edwards, The Essential Client/Server Architecture: Survivor's Guide, John Wiley & Sons, 1995.

****

CS 6363 – Design and Analysis of Computer Algorithms

The exam will test knowledge of:
1. major techniques for algorithm design (as listed below);
2. methods to prove algorithm correctness and to analyze its running time;
3. Basic knowledge of NP-Completeness.

NOTE: You should know more than just the algorithms; you are responsible for proving correctness, including all necessary supporting lemmas, and are responsible for proving the correctness of any statements about the asymptotic running times. In addition, you should know the stated subject matter well enough to enable you to provide solutions for closely related questions.

Most topics (and knowledge) required are in the CS6363 textbook:

Introduction to algorithms, Second edition, Cormen, Leiserson, Rivest and

Stein.

General topics:
Introduction, recurrences and Master Theorem (Theorem 4.1, the proof is not required)

Divide-and-Conquer algorithms
- Linear time median selection algorithm (Section 9.3, pp. 189-192)
- Closest pair of points in the plane (Section 33.4, pp. 957-961)
- Permutation networks (Problem 27-3, page 722)
- Multiplication of large integers (Section 7.1, page 219-223, and Problems 7.2, 7.3, page 250, of "Fundamentals of Algorithms, by Brassard and Bratley, Prentice Hall Publ.)

Note: students should be able to design divide-and-conquer algorithms for various problems beside those mentioned above

Dynamic Programming
- Matrix Chain Order (Section15.2, pp. 331-338)
- Longest Common Subsequence Algorithm (Section 15.4, pp. pp. 350-355.)
- All pairs shortest paths (Section 25.2, pp. 629-634)
- 0/1-knapsack problem (Problem 16.2-2, page 384)

Greedy Method
- Huffman's code algorithm (Section 16.3, pp. 385-392)
- Minimum spanning tree (Chapter 23)
- Single Source Shortest Paths (e.g. Dijkstra's algorithm) (Chapter 24, up to page 601)

Maximum flow (Chapter 26, up to page 668)

Graph algorithms (Chapter 22)

NP-Completeness (Chapter 34, specifically 3SAT, VERTEX COVER, INDEPENDENT SET, CLIQUE, 3COLOR, HAMILTON CIRCUIT (both directed and undirected), as well as definitions and properties of polynomial time reducibilities.)

Linear programming: (Chapter 29, pp. 770-789 and pp. 804-807.)

****

CS 6364 – Artificial Intelligence

Text:
S. Russell & P. Norvig. *Artificial Intelligence, A Modern Approach,* Second Edition 2002.

Problem solving by search:
Uninformed (Blind) Search and Heuristic (Informed) Search

Problem formulation; Uninformed search strategies: Depth-First Search, Breadth-First Search, Ununiform-Cost Search, Iterative-Deepening.
Informed Search strategies: Greeedy Best-First Search, A*, IDA*.
Heuristic Functions: heuristic domination, inventing admissible heuristics.

Adversary Search (Game Trees)

How to design computer programs that play games intelligently. The MIN/MAX and the ALPHA/BETA-Pruning algorithms, their complexity and efficient implementations.

Knowledge Representation

Propositional logic. Syntax, semantics and inference in prepositional logic as well as reasoning patterns. First Order Logic: syntax and semantics. Resolution in FOL.

Probabilistic reasoning

Modeling uncertainties with probabilities. Inference using Full Joint Distributions. Bayes' Rule. Naïve-Bayesian Reasoning.

Bayesian Networks / Belief Networks

Representation of knowledge in uncertain domains.Semanticsof
Bayesian Networks. Exact inference in Baysian Networks: inference by enumeration; PolyTree Bayesian networks..

****

CS 6367 – Software Testing, Validation, and Verification

Textbook: Software Testing by Paul Jorgensen, 2nd Edition, CRC.
Part 1: Requirements-Based Testing, Inspections
Introduction, Approaches to Reliability, Requirements-based Testing strategies (Equivalence Partitioning, Boundary value Analysis, Cause-Effect graphing), Valid and Reliable testing strategies and the Fundamental Theorem of testing, the Partition Testing Model, Random/Statistical testing. Software Inspections and related approaches.

Textbook: Ch 1, 3, 5-8
Myers: The Art of Software Testing, Wiley.
Goodenough + Gerhart, "Toward a Theory of Test Data Selection", IEEE Trans. on Software Engineering, June 1975.
Hamlet+Taylor, "Partition Testing Does Not Inspire Confidence", IEEE TSE, Dec. 1990.
Wheeler, Brykczynski, Meeson, "Software Inspection: An Industry Best Practice", IEEE Computer Society Press.

Part 2: Program Proofs
Predicate calculus, validity, theoretical limitations, deduction systems, the Resolution method. Verification of Programs (Flowchart Programs, Inductive Assertions, Termination, Programs with Arrays, extensions).

Manna: Mathematical Theory of Computation, McGraw-Hill.
Chapter 2: Predicate Calculus
Chapter 3: Verification of Programs

Part 3: Structural, Fault-Based Testing Strategies
Structural Testing, Statement, Branch, Predicate, Base-Path, Path Testing, Variations of Path Testing, Data-Flow Testing, Domain Testing, Mutation Analysis, other methods. Evaluations of testing strategies, inclusion, test set size. Integration testing; Object-oriented Testing

Textbook: Ch. 9-11, 13, 16-20
DeMillo, Lipton, Sayward, "Hints on Test Data Selection: Help for the Practicing Programmer", IEEE Computer, April 1978.
Musa, "Operational Profiles in Software Reliability Engineering", IEEE Software, March 1993.
Ntafos, "A Comparison of Some Structural Testing Strategies", IEEE TSE, June 1988.
White, Cohen, "A domain strategy for Computer Program Testing", IEEE-TSE, May 1980.
Part 4: Reliability Estimation
Failure rate estimation from test outcomes, error-seeding, reliability growth models.
Notes on Reserve in Library
References:
Lyu: Handbook of Software Reliability Engineering, IEEE Computer Society Press, Mc Graw Hill.
Musa: Software Reliability Engineering, McGraw-Hill.
****

CS 6371 – Advanced Programming Languages

Topics:
Programming with Functions; Lambda Calculus and ML programming;
Logic programming; Unification and backtracking; Search tree; Programming in Prolog;Abstract Syntax; Definite Clause Grammars; Grammar Classifications;
Sets, functions, domains;
Domain Theory: Primitive and Compound Domains; Denotational Definition of Programming Languages; Semantics of Imperative Languages; Recursive Functions;
Monotonicity, Continuity, and Fix-points;
Introduction to semantics of Logic Programming Languages,
Verification of Programs, Partial Evaluation; Interpretation and Automatic Compilation;
Axiomatic Semantics: Hoare's Axiomatization of partial correctness
References:
Denotational Semantics by D.A. Schmidt.
Elements of ML Programming, Jeffrey D. Ullman, ML97 Edition
The Art of Prolog, L. Sterling and E. Shapiro. MIT Press, 1997.
Also see the following web page for more details:
http://www.utdallas.edu/~gupta/courses/apl/

****

CS 6375 – Machine Learning (Syllabus updated Oct 2006)

Topics: Decision Tree Learning, Artificial Neural Networks, Evaluating Hypotheses, Bayesian Learning, Computational Learning Theory, Instance-Based Learning, Markov Decision Processes, Reinforcement Learning, Support Vector Machines, Bagging, Boosting, Hidden Markov
Models, and Clustering.

References:
Artificial Intelligence (second edition) by Stuart Russell and Peter Norvig, Prentice Hall, 2003.
Machine Learning by Tom Mitchell, McGraw Hill, 1997.

****

CS 6378 – Advanced Operating Systems (Material in red with strikethrough is no longer in the syllabus)

Textbook: ``Advanced Concepts in Operating Systems'' by Mukesh Singhal and Niranjan G. Shivaratri.

Introduction Chapters

1, 2. Theoretical Foundations

Chapter 4: Section 4.1-4.5

Chapter 5: Entire chapter.

Clock Synchronization: Cristian's method, Berkeley algorithm, Networktime protocol, Srikant & Toueg's algorithm (from papers listed below)

Papers:

K. Birman, A. Schiper, and P. Stephenson
Lightweight Causal and Atomic Broadcast
ACM Transactions on Computer Systems, 9(3):272--314, 1991.

K. M. Chandy and L. Lamport. Distributed Snapshots : Determining Global States of Distributed Systems ACM Transactions on Computer Systems, 3(1):63--75, February 1985.

L. Lamport. Time, Clocks and the Ordering of Events in a Distributed System Communications of the ACM, 21(7):558--565, July 1978.

C. Fidge, "Logical time in distributed computing systems," IEEE Computer, vol. 24, pp. 28-33, Aug. 1991.

Cristian, F. "Probabilistic Clock Synchronization," Distributed Computing, Volume 3, 1989, pp 146-158.

Gusella R and Zatti, S. "The accuracy of clock synchronization achieved by TEMPO in Berkeley UNIX 4.3BSD." IEEE Transactions on Software Engineering, Volume 15, 1989, pp 847-853.

Mills, D. "Improved algorithms for synchronizing computer network clocks," IEEE Transations Networks, June, 1995, pp 245-254.

Srikanth, T. K. and Toueg, S. 1987. Optimal clock synchronization. *Journal of the. ACM, Volume* 34, # 3 (Jul. 1987), 626-645

Distributed Mutual Exclusion

Chapter 6: Entire chapter.

Papers:

M. Maekawa
A sqrt{N} Algorithm for Mutual Exclusion in Decentralized Systems
ACM Transactions on Computer Systems, pages 145--159, May 1985.

Deadlocks

Chapter 3: Entire chapter.

Agreement Protocols

Chapter 8: Sections 8.1-8.4

Papers:

L. Lamport, R. Shostak, and M. Pease. The Byzantine Generals Problem
ACM Transactions on Programming Languages and Systems, July 1982.

Distributed File Systems

Chapter 9: Sections 9.1 – 9.4

RAID

Papers:

Peter M. Chen, Edward K. Lee, Garth A. Gibson, Randy H. Katz, and David A. Patterson. "RAID: High-Performance, Reliable Secondary Storage." ACM Computing Surveys, Volume 26, Number 2, June 1994, Pages 145-185.

~~Distributed Scheduling (topic deleted on Oct 2006)~~

~~Chapter 11: Section 11.1 – 11.9. Read the case studies to get an idea of how the concepts are implemented.~~

~~Papers:~~

~~D. L. Eager, E.D. Lazowska, and J. Zahorjan. Adaptive Load Sharing in Homogeneous Distributed Systems IEEE Transactions on Software Engineering, SE12(5):662--675, May 1986.~~

Recovery
Chapter 12: Sections 12.1 -12.9.
Papers:
R. Koo and S. Toueg. Checkpointing and Rollback-Recovery for Distributed Systems. IEEE Transactions on Software Engineering, SE-13(1):23—January 1987

Fault Tolerance
Chapter 13: Sections 13.1 - 13.11.
Papers:
S.B. Davidson, H. Garcia-Molina, and D. Skeen. Consistency in Partitioned Networks. ACM Computing Surveys, 17(3):341--370, September 1985.

****

CS 6390 – Advanced Computer Networks

General topics:

1) Transport and Routing (including multicasting) protocols,
2) Quality of Service,
3) Mobile IP/Wireless Data
4) IPV6
5) MPLS
6) Peer-to-peer applications
7) Voice over IP

Reading List

1. Reference book (Computer Networks by Peterson and Davie).
2. Design Philosophy of the DARPA Internet Protocols, D. Clark, Proc. of ACM SIGCOMM '88.
3. An Architecture for Wide-Area Multicast Routing, S. Deering, D. Estrin, D. Farinacci, V.
Jacobson, C.-G. Liu, and L. Wei, Proc. of ACM SIGCOMM'94.
4. Multicast Routing in Datagram Internetworks and Extended LANS, S. Deering and D.
Cheriton, ACM Transactions on Computer Systems, Vol 8 No 2, May 1990, pp. 85-110.
5. The Stable Paths Problem and Interdomain Routing, T. Griffin, B. Shepherd, and G.
Wilfong, IEEE/ACM Transactions on Networking, Vol 10 No 2, April 2002.
6. Mobile IP, C. Perkins, IEEE Communications Magazine, Vol 35, No. 5, May 1997.
7. Mobility Support in IPV6, C. Perkins, D. Johnson, ACM Mobicom 1996.
8. IP Multicast Channels: EXPRESS Support for Large-scale Single-source Applications, H.
Holbrook and D. Cheriton, SIGCOM 1999.
9. Congestion Avoidance and Control , V. Jacobson and M. Karels, Proc. ACM SIGCOMM '88.
10. Random Early Detection Gateways for Congestion Avoidance, S. Floyd and V. Jacobson,
IEEE/ACM Transactions on Networking, Vol. 1, No. 4, pp. 397-413, August 1993.
11. Equation-Based Congestion Control for Unicast Applications, S. Floyd, M. Handley, J. Padhye,
and J. Widmer, Proc. of ACM SIGCOMM '00, Aug. 2000.
12. Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications, I Stoica, R
Morris, D Liben-Nowell, D R. Karger, M. F Kaashoek, F Dabek, H Balakrishnan, ACM SIGCOMM 2001.
13. SOS: Secure Overlay Services, A. Keromytis, V. Misra, and D. Rubenstein (Columbia University), ACM SIGCOMM, Pittsburgh, PA, USA, August 2002.
14. SIFF: A Stateless Internet Flow Filter to Mitigate DDoS Flooding Attacks, Abraham Yaar, Adrian
Perrig, Dawn Xiaodong Song, IEEE Symposium on Security and Privacy 2004

NOTE: You can find most of these papers at:
http://www.utdallas.edu/%7Eksarac/courses/Papers/
The paper may also be found in the IEEE/IEE Xplore database and in the ACM Digital library, available
from UTD's library webpage
http://www.utdallas.edu/library/collections/journals.htm