THE ISIS PROJECT:

Fault-tolerance in large distributed systems

Final R & D Status Report
Jan 1, 1993

Prof. Kenneth P. Birman
Prof. Keith Marzullo
Department of Computer Science
Cornell University, Ithaca New York
607-255-9199

This work was sponsored by the Defense Advanced Research Projects Agency (DoD), under contract NAG 2-593 issued by the NASA Ames Research Center, Moffett Field, CA.

Personnel

- Academic Staff:
  - Prof. Kenneth P. Birman, Co-Investigator
  - Prof. Keith Marzullo, Co-Investigator
  - Dr. Robert Cooper, Research Associate
  - Dr. Robbert van Renesse, Research Associate
  - Dr. Patrick Stephenson, Post-doctoral Research Associate
- Graduate Students:
  - Lorenzo Alvisi (Marzullo)
  - Navin Budhiraja (Marzullo)
  - Brad Glade (Birman)
  - Guerney Hunt (Birman)
  - Neil Jain (Birman)
  - Michael Kalantar (Birman)
  - Michael Reiter (Birman)
  - Aleta Ricciardi (Birman)
  - Laura Sabel (Marzullo)

The Isis project

This final status report covers activities of the Isis project during the first half of 1992. This is our 8'th progress report under NASA funding, and we assume that the reader has some background regarding the goals and status of our effort, and focus instead on technical accomplishments during the report period and goals for the next six months. Readers unfamiliar with our work could start by reading some of the papers cited below.

During the report period, the Isis effort has achieved a major milestone in its effort to redesign and reimplement the Isis system using Mach and Chorus as target operating system environments. In addition, we completed a number of publications that address issues raised in our prior work; some of these have recently appeared in print, while others are now being considered for publication in a variety of journals and conferences.

With the completion of this milestone, all major goals for the NASA-funded effort have been successfully completed.

The major accomplishments of the final semi-annual report period are:

- We completed the design and prototype implementation of our new "lightweight groups" facility, which will eventually run in Mach or Chorus. This is a major practical advance for the group, which has been working on this problem for the past two years. Although our new system has yet to be integrated with Mach, it does implement the lightweight causal and atomic multicast protocols of our 1991 ACM TOCS paper, support the causal domain model that we introduced recently, and achieves extremely high performance and parallelism even over UNIX. We are extremely encouraged by this development. Predictions of a 10- to 100-fold performance improvement appear to be justified, but until we have this new software running under native Mach it will be difficult to say anything final on the issue.

- We continued work on a new way of presenting Isis groups that will reduce costs by allowing Isis to map multiple application-level process groups to a single Isis process group. The idea here is to amortize membership changes over multiple groups so as to reduce their effective cost. The technique us expected to avoid high overhead in applications that use very large numbers of nearly identical groups.

- Formation of research ties with other laboratories, including the Los

DISPLAY 09/2/2
91N13928*# ISSUE 5 PAGE 704 CATEGORY 62
RPT#: NASA-CR-186625 NAS 1.26:186625 CNT#: NAG2-593 90/07/01 34 PAGES
UNCLASSIFIED DOCUMENT
UTTL: The ISIS project: Fault-tolerance in large distributed systems TLSP: Semiannuel Status Report, 1 Jan. - 1 Jul. 1990
AUTH: A/BIRMAN, KENNETH P.; B/MARZULLO, KEITH
CORP: Cornell Univ., Ithaca, NY. CSS: (Dept. of Computer Science.)
SAP: Avail: CASI HC A03/MF A01
CIO: UNITED STATES
MAJS: /*COMMUNICATION NETWORKS/*COMPUTER SYSTEMS PROGRAMS/*FAULT TOLERANCE/* PROGRAMMING LANGUAGES/*PROTOCOL (COMPUTERS)/*SOFTWARE TOOLS
MINS: / ARCHITECTURE (COMPUTERS)/ INFORMATION SYSTEMS/ REAL TIME OPERATION/ ROBUSTNESS (MATHEMATICS)/ USER MANUALS (COMPUTER PROGRAMS)
ABA: Y.S.
ABS: The semi-annual status report covers activities of the ISIS project during the second half of 1989. The project had several independent objectives: (1) At the level of the ISIS Toolkit, ISIS release V2.0 was completed, containing bypass communication protocols. Performance of the system is greatly enhanced by this change, but the initial software release is limited in some respects. (2) The Meta project focused on the definition of the Lomita programming language for specifying rules that monitor sensors for conditions of interest and triggering appropriate reactions.
MORE
ENTER: =-•PC LINE 24 COL 9
4B• A

Alamos Advanced Computing Laboratory (which focuses on supercomputing), Portugal's INESC research laboratory (known for its work on realtime communication), and with Mach-related research efforts at the Open Software Foundation, Carnegie Mellon, and University of Arizona.

- We continued the development and initial implementation of the new security architecture for Isis, reported in the first progress report for 1992. We view this as an extremely important advance, because the previous version of Isis was almost completely trusting of its users. The secure Isis architecture, in contrast, can tolerate arbitrary failures outside an island of physically secure nodes, and supports a highly sophisticated trust, encryption and delegation architecture within an island of secured nodes. Implementation of this architecture is proving to be a cornerstone of our new system, and substantial parts of the code now exist.

- We completed the implementation of a Meta rule manager. This decentralized Isis program can be thought of as a "run-time" environment that dynamically loads Meta rules onto instrumented components as they become active or recover from crashes. The rule manager operates from a description of the instrumented program (much like the schema of an object-oriented database) and allows a user to make simple queries about the status of the instrumented program.

  The lack of such a rule manager has been a major stumbling block to the clients of Meta. We are also continuing to expand its function into a full-fledged runtime system. We plan to add support for interactively debugging active Meta rules and to add graphical tools for monitoring the status of the application.

- We are completed the design of a higher-level language for Meta. This has proven to be more difficult than we thought since the differences between what appear to be reasonable semantics of temporal commands are subtle. The new language is being implemented, and will replace a simpler version of Lomita that only supports Meta-style guarded commands and rules for maintaining the membership of aggregates. This compiler produces object files that are read by the rule manager mentioned in the previous item which in turn activates rules on the instrumented application.

The main drawback with the previous, simple version of Lomita is the lack of control flow structures—for example, recovery when some control rule terminates abnormally. Hence, we are extending the function of the Meta *shell* actuator to allow sensor values to be passed in as environment variables. Combined with a shell command that accesses Meta (also nearing completion), this will allow a programmer to write shell scripts that are invoked by Meta as actuators. Such scripts can both record state for temporal matching and perform complex control functions by using both Unix features and Meta sensors and actuators.

- Almost all of the applications that Meta has been used for outside of Cornell have used the sensor abstractions of Meta much more than the actuator abstractions. We think that part of this is due to the lack of rule support mentioned above, but it is also somewhat due to the lack of a good example that could be distributed with Meta. Hence, we have built such an example application that uses Meta to load balance requests to a set of simulated computation servers. Writing this application has (not unexpectedly) flushed out a set of subtle bugs in Meta and Isis, and so the example application is not ready for distribution at the time this report was written. It currently exhibits simple rules (such as transparent submission to lightly-loaded servers) and we are currently adding more complex control rules (such as dynamic server creation and removal based on average service load).

  We are also rewriting the Isis Resource Manager as a Meta client. Again, this has flushed out a set of problems with Meta (most notably, the lack of support for remote Isis and the lack of support of large aggregates). We expect to have the Isis Resource Manager fully functional as a Meta client by the end of 1992.

- We have made substantial progress in a new experimental effort to understand flow control problems on hardware multicast technologies such as ethernet, FDDI and token ring, and are extending our work to include next-generation technologies such as ATM. The goal of this effort is to develop effective flow-control algorithms for use within the Isis multicast protocols. So far, we have focused on collecting data concerning the behavior of the raw devices themselves, and have obtained fascinating and non-intuitive results concerning packet loss rates in a number of settings. These show that the most significant loss rates are for small packets sent in many-one or many-many situations. Low or

zero loss occurs with large packets and for one-many patterns. This information will be used to develop algorithms that narrow in on the situations in which loss rates are highest, while remaining uninvolved in other situations. Such flow control algorithms are the key element limiting Isis performance on many systems, and development of this new flow control software will be a small but critical activity for us during the coming year.

- We have initiated a new project to explore specialized implementations of Isis for the CM/5 and Intel Touchstone multiprocessors. This work is motivated by the impressive results of Berkeley's Split/C and Active Messages research, demonstrating that asynchronous communication can lead to tremendous performance gains on the most important emerging parallel processors. As we move Isis onto these platforms, we want to build our protocols in ways that exploit the hardware fully and minimize unnecessary work in software – work needed on networks but not on closely coupled machines. We are very excited about this new direction.

With this final progress report, it is interesting to observe that Isis seems also at the end of a period of initial transition. The original version of Isis is no longer a subject of active research at Cornell, and the initial version of the Meta system is also finished. With the successful handoff of these systems to ISIS Distributed Systems (and the widespread release of public, source-form distributions), technology transition for this version of ISIS is well established, and Cornell is now free to focus on the development of the next generation of this technology.

Users of the technology include Sematech, Hughes (EOS), GE/Motorola (Iridium), the military (HiperD), the financial community (New York Stock Exchange, World Bank, many banks and brokerages), CERN, Los Alamos, FermiLab, GTE, SouthWestern Bell Telephone, and many other large and small companies, both for commercial and for research purposes. DARPA and Nasa, though support for Isis, have created a new technology that is clearly having an enduring impact on the way that distributed systems are developed in the United States and worldwide.

On the research side, Isis and Meta have launched a major wave of activity in the O/S community. Hundreds of papers have been written by dozens of research groups on variations of the Isis approach. The technology can only improve from this type of activity, and there can be no clearer proof that the approach is valid and viable.

The redesign of Isis has been structured around a much simplified core of protocols and system management routines (a sort of "micro-kernel"). This core is flexible enough to support all existing Isis functionality, as well as real-time applications, secure applications, a version of the Isis toolkit optimized for parallel processing environments, and support for object-oriented and modular programming languages, like C++ and ADA. We are building this new software layer so that it can run directly over the Mach and Chorus kernels, while continuing to support a UNIX-level interface similar to our current toolkit interface.

A final comment relates to our continued and enlarging discussions with industry. We are now actively pursuing standardization of the Isis approach to group computing with Unix Systems Laboratories, Unix International, the Open Software Foundation, the Texas Instruments/DARPA OODB project, Electronic Joint Venture, and other standards organizations. This is having significant impact, as demonstrated by the decision of OSF to integrate Isis into OSF 1/AD and the recent announcements by Unix International and USL concerning the key role that reliable process group technologies will play in their future products. We are increasingly joined by industry strategists in recognizing Isis and META as enablers for a whole new generation of highly reliable, large-scale, self-managing distributed software. We believe that DARPA and NASA can point to this emerging trend as a demonstration of the huge impact that government research activities can have on industry, given sufficient time, sufficient investment, and consistently positive results to point to.

Publications

On the following pages, we reproduce a full list of publications by the effort, including some very recent ones as well as earlier work. A good general review of the project is TR 1216, soon to be published by the Communications of the ACM. We are beginning work on a book that will collect the most important papers into a single volume. Where a paper has appeared in a journal, the TR version will often be out of date, since copyright restrictions prevent us from distributing versions of these papers for which copyright has been assigned to the journal. These revisions are often important, so where a journal version is cited, we urge the potential reader to consult the journal itself and not to rely on the older TR copy stated on the list.

PUBLICATIONS LIST

ISIS Activity

92-xxxx* Light-Weight Process Groups. Bradford B. Glade, Kenneth P. Birman, Robert C. B. Cooper and Robbert van Renesse. *Proceedings of the Open-Forum '92 Technical Conference*, November 1992, 323-336.

92-xxxx* A RISC Approach to Process Groups. Robbert van Renesse, Robert Cooper, Bradford Glade, Patrick Stephenson. *Proceedings of the 5th ACM SIGOPS Workshop*, Rennes, France, September 21-23, 1992.

92-1289* Using the ISIS Resource Manager for Distributed, Fault-Tolerant Com-. puting. Timothy Clark and Kenneth Birman. June 1992

92-1287* How to Securely Replicate Services. Michael Reiter and Kenneth Birman. June 1992. Submitted to ACM Transactions on Programming Languages and Systems.

92-xxxx Reliable Multicast between Microkernels. Robbert van Renesse, Ken Birman, Robert Cooper, Brad Glade, and Pat Stephenson. *Proceedings of the USENIX Workshop on Micro-Kernels and Other Kernel Architectures*, Seattle, Washington, April 27-28, 1990, 269-283.

92-1269* Integrating Security in a Group Oriented Distributed System (replaces 1239). Michael Reiter, Kenneth Birman, and Li Gong. February 1992. Also in the *Proceedings of the 1992 IEEE Symposium on Research in Security and Privacy*, Oakland, CA, (May 1992), 18-32.

92-xxxx The Cost of Order in Asynchronous Systems. Aleta Ricciardi, Ken Birman, and Pat Stephenson. *Springer-Verlag Lecture Notes in Computer Science*, 1992.

91-1240* Maintaining Consistency in Distributed Systems. Kenneth Birman. October 1991. Submitted to Journal of Parallel and Distributed Computing.

91-1233* Practical Utility of Knowledge-Based Analyses: Optimizations and Optimality for an Implementation of Asynchronous, Fail-Stop Processes. Aleta M. Ricciardi. September 1991. To appear in the *Proceedings of the Fourth Conference on the Theoretical Aspects of Reasoning About Knowledge*, (1992).

91-1216* The Process Group Approach to Reliable Distributed Computing. Kenneth P. Birman. July 1991. To appear in the *Communications of the ACM.*

PUBLICATIONS LIST Continued

91-1192* Lightweight Causal and Atomic Group Multicast. Kenneth Birman, Andre Schiper and Pat Stephenson. February 1991. *ACM Transactions on Computer Systems*, **9**, 3, (August 1991), 272-314.

91-1188* Using Process Groups to Implement Failure Detection in Asynchronous Environments. Aleta Ricciardi and Kenneth Birman. *ACM Symposium on Principles of Distributed Computing*, Montreal, Quebec, Canada, (August 19-21, 1991), 341-353. *Principles of Distributed Computing* (1991), 341-353.

90-1165* Designing Application Software in Wide Area Network Settings. Mesaac Makpangou, Kenneth Birman. October 1990.

90-1138* The ISIS Project: Real Experience with a Fault Tolerant Programming System. Kenneth Birman and Robert Cooper. *Operating Systems Review*, (April 1991), 103-107. *ACM/SIGOPS European Workshop on Fault-Tolerance Techniques in Operating Systems*, Bologna, Italy, 1990.

90-1105R* Fast Causal Multicast. *Operating Systems Review* (April 1991), 75-79. Kenneth Birman, Andre Schiper and Pat Stephenson. March 1990. Revised February 1991. New TR 91-1192. New title: Lightweight Causal and Atomic Group Multicast. To appear in *ACM TOCS*.

89-1067 Log-Based Recovery in Asynchronous Distributed Systems. Kenneth Kane. December 1989.

89-1042 Deceit: A Flexible Distributed File System. Alex Siegel, Kenneth Birman and Keith Marzullo. November 1989.

89-xxxx Supporting Large Scale Applications on Networks of Workstations. Robert Cooper and Kenneth Birman. April 1989. *Second Workshop on Workstation Operating Systems*, Asilomar, CA, September 1989.

89-1014 How Robust are Distributed Systems. Kenneth Birman. June 1989. *An Advanced Course on Distributed Systems - lecture notes from Artic '88*, Addison-Wesley, 1989. To be published.

89-1001 The Role of Order in Distributed Programs. Kenneth Birman and Keith Marzullo. March 1989. Submitted for publication.

88-949 Causally Consistent Recovery of Partially Replicated Logs. Kenneth Birman and Kenneth Kane. November 1988. Submitted for publication.

PUBLICATIONS LIST Continued

88-928 The Use of Efficient Broadcast Protocols in Asynchronous Distributed Systems. Frank Schmuck. (Ph.D. Thesis). August 1988.

88-918 Reliable Broadcast Protocols. Chapter 14 in *Distributed Systems*, Thomas Joseph and Kenneth Birman. June 1988. Sape J. Mullender, ed., Addison-Wesley/ACM Press Series (1989).

88-917 Exploiting Replication. Chapter 15 in *Distributed Systems*, Thomas Joseph and Kenneth Birman. June 1988. Sape J. Mullender, ed., Addison-Wesley/ACM Press Series (1989).

87-849 ISIS - A Distributed Programming Environment, Version 2.1 - User's Guide and Reference Manual. Kenneth Birman, Thomas Joseph and Frank Schmuck. July 1987.

87-811 Exploiting Virtual Synchrony in Distributed Systems. Kenneth Birman and Thomas Joseph. February 1987. *11th ACM Symposium on Operating Systems Principles*, December 1987. Also appearing in *Operating Systems Review*, **22**, 1, (December 1987), 123-138.

86-781 Efficient Concurrency Control for Libraries of Typed Objects. T. Raeuchle. (Ph.D. Thesis). September 1986.

86-772 Programming with Shared Bulletin Boards in Asynchronous Distributed Systems. Kenneth Birman, Thomas Joseph and Pat Stephenson. August 1986. (Revised December 1986).

86-753 Communication Support for Reliable Distributed Computing. Kenneth Birman and Thomas Joseph. *Lecture Notes in Computer Science* (1990), 124-137. *Proc. Asilomar Workshop on Fault Tolerant Distributed Computing*, March 1986.

86-744 ISIS: A System for Fault-Tolerance in Distributed Systems. Kenneth Birman. April 1986.

85-712 Low Cost Management of Replicated Data. Thomas Joseph. (Ph.D. Thesis). November 1985.

PUBLICATIONS LIST Continued

85-694 Reliable Communication in the Presence of Failures. Kenneth Birman and Thomas Joseph. July 1985. (Revised August 1986). *ACM Transactions on Computer Systems*, **5**, 1, (February 1987), 47-76.

85-668 Replication and Fault-Tolerance in the ISIS System. Kenneth Birman. March 1985 (Revised September 1985). *10th ACM Symposium on Operating Systems Principles*, (December 1985), 79-86. *Operating Systems Review*, **19**, 5, (December 1985).

84-644 Low-Cost Management of Replicated Data in Fault-Tolerant Distributed Systems. Kenneth Birman and Thomas Joseph. October 1984. *ACM Transactions on Computer Systems*, **4**, 1, (February 1986), 54-70.

84-642 An Overview of the ISIS Project. Kenneth Birman, A. El Abbadi, W. C. Dietrich, Thomas Joseph and T. Raeuchle. October 1984. *IEEE Distributed Processing Technical Committee Newsletter.* January 1985.

84-594 Implementing Fault-Tolerant Distributed Objects. Kenneth Birman, Thomas Joseph, T. Raeuchle, and A. El Abbadi. *4th Symposium on Reliability in Distributed Systems and Database Systems,* Silver Springs, MD, October 1984. *IEEE Transactions on Software Engineering*, **SE-11**, 6, (June 1985), 502-508.

83-552 ISIS: An Environment for Constructing Fault-Tolerant Distributed Systems. Kenneth Birman, D. Skeen, A. El Abbadi, W. C. Dietrich and T. Raeuchle. May 1983.

META Activity

92-1277* Using Consistent Subcuts for Detecting Stable Properties (replaces TR 91-1205). Keith Marzullo, Laura Sabel. April 1992.

91-1252* Fault-Tolerant Management of Distributed Applications Using the Reactive System Architecture. PhD Thesis. Mark Wood. December 1991.

91-1238* Putting Time into Proof Outlines. Fred B. Schneider, Bard Bloom and Keith Marzullo. September 1991.

91-1221* Detection of Global State Predicates. Keith Marzullo and Gil Neiger. November 1991. To appear in *Proceedings of the Fifth Workshop on Distributed Algorithms and Graphs*, (Springer-Verlag), Delphi, Greece, October 1991.

PUBLICATIONS LIST Continued

91-1205* Using Consistent Subcuts for Detecting Stable Properties. Keith Marzullo and Laura Sabel. May 1991. To appear in *Proceedings of the Fifth Workshop on Distributed Algorithms and Graphs*, (Springer-Verlag), Delphi, Greece, October 1991).

91-1200* Consistent Detection of Global Predicates. Robert Cooper and Keith Marzullo. April 1991. *ACM/ONR Workshop on Parallel and Distributed Debugging*, 163-173 (1991).

91-1193* Tools for Constructing Distributed Reactive Systems. Keith Marzullo and Mark Wood. February 1991.

91-1190* Masking Failures of Multidimensional Sensors. Paul Chew and Keith Marzullo. February 1991. *Proceedings of the Tenth Symposium on Reliable Distributed Systems*, Pisa, Italy, (October 1991), 32-41.

91-1187* Tools for Monitoring and Controlling Distributed Applications. Keith Marzullo and Mark Wood. February 1991. *IEEE Computer*,**24**, 8 (August 1991), 42-51.

90-1156* Tolerating Failures of Continuous-Valued Sensors. Keith Marzullo. September 1990. *ACM Transactions on Computer Systems*, **8**, 4, (1990), 284-304.

90-1155* Making Real-Time Reactive Systems Reliable. Keith Marzullo and Mark Wood. September 1990. *Proceedings of the Fourth ACM SIGOPS European Workshop*, (1990), 1-4.

90-1136* Tools for Distributed Application Management. Keith Marzullo, Robert Cooper, Mark Wood and Kenneth Birman. *IEEE Computer*, **24**, 8, (August 1991), 42-51.

89-997 Implementing Fault-Tolerant Sensors. Keith Marzullo. May 1989. Submitted for publication.

ISIS/META Activity

90-1103 ISIS and Meta Projects: Progress Report. Kenneth Birman, Robert Cooper and Keith Marzullo. February 1990.

89-xxx The ISIS Distributed Programming Toolkit and The Meta Distributed Operating System. Ken Birman and Keith Marzullo. *SUN Technology* **2**, 1, (Summer 1989).

PUBLICATIONS LIST Continued

Other Distributed Systems Activity

92-1317* Nonblocking and Orphan-Free Message Logging Protocols. Lorenzo Alvisi, Bruce Hoppe, Keith Marzullo. December 1992.

92-1299* Optimal Primary-Backup Protocols. Navin Budhiraja, Keith Marzullo, Fred B. Schneider, and Sam Toueg. August 1992. To appear in the Sixth Workshop on Distributed Algorithms, Haifa, Israel, November 1992.

92-1298* Fault-Tolerant Wait-Free Shared Objects. Prasad Jayanti, Tushar Deepak Chandra, and Sam Toueg. August 1992. (A revision of TR 92-1281, April 1992). A summary of these results will appear in FOCS92.

92-1293* The Weakest Failure Detector for Solving Consensus. Tushar Deepak Chandra, Vassos Hadzilacos, and Sam Toueg. July 1992. A shorter version appeared in the Principals of Distributed Computing, August 1992 in Vancouver.

92-xxxx Distributed Programming with Asynchronous Ordered Channels in Distributed ML. Robert Cooper and Clifford Krumvieda. To appear in the *Proceedings of the ACM SIGPLAN Workshop on ML and it Applications*, June 1992.

92-xxxx Expressing Fault-Tolerant and Consistency-Preserving Programs in Distributed ML. Clifford D. Krumvieda. To appear in the of the *ACM SIGPLAN Workshop on ML and its Applications*, June 1992.

92-1281* Fault-Tolerant Wait-Free Shared Objects. Prasad Jayanti, Tushar D.. Chandra and Sam Toueg. April 1992

92-1265* Primary-Backup Protocols: Lower Bounds and Optimal Implementations. Navin Budhiraja, Keith Marzullo, Fred B. Schneider and Sam Toueg. January 1992. A shorter version appeared in DCCA-3, Mondello, Italy, September 1992.

PUBLICATIONS LIST Continued

91-1257* Design Alternatives for Process Group Membership & Multicast (replaces TR 91-1185). Kenneth Birman, Robert Cooper, and Barry Gleeson. December 1991. Submitted to *IEEE Transactions on Parallel and Distributed Systems.*

91-1249* Tools and Techniques for Adding Fault Tolerance to Distributed and Parallel Programs. Ozalp Babaoglu. December 1991.

91-xxxx Lower Bounds for Primary-Backup Implementations of Bofo Services. Navin Budhiraja, Keith Marzullo, Fred B. Schneider and Sam Toueg. *Proceedings ONR 2nd Annual Workshop on Ultradependable Multicomputers and Electronic Systems*, Washington, DC, (November 1991), 81-86.

91-xxxx DML: Packaging High-Level Distributed Abstractions in SML. Clifford D. Krumvieda. September 1991. *Proceedings of the Third International Workshop on Standard ML*, Robert Harper (ed.), Department of Computer Science, Carnegie Mellon University, September 26-27, 1991.

91-1225* Unreliable Failure Detectors for Asynchronous Systems. Tushar Deepak chandra and Sam Toueg. August 1991.

91-1217* Derivation of Sequential, Real-Time, Process-Control Programs. Navin Budhiraja, Keith Marzullo and Fred B. Schneider. July 1991. In *Foundations of Real-Time Computing: Formal Specifications and Methods*, Kluwer Academic Publishers 1991, pp. 39-54. Navin Budhiraja, Keith Marzullo and Fred B. Schneider. July 1991.

90-1141* MTP: An Atomic Multicast Transport Protocol. Alan O. Freier and Keith Marzullo. July 1990.

89-996 Concurrency Control for Transactions with Priorities. Keith Marzullo. May 1989.

Technical reports marked with a '*' can be copied from ftp.cs.cornell.edu using anonymous, binary ftp. The reports are in the "pub" subdirectory.