(2)

FAULT TOLERANT REAL-TIME SYSTEMS

Contract Numbers:
N00014-84-K-0734 and N00014-92-J-1771

YEARLY REPORT

1 October 1991 - 30 September 1992

Prepared for:

Chief of Naval Research
Code 1267/Annual Report
Ballston Tower One
Room 528
800 North Quincy Street
Arlington, Virginia 22217-5660

Prepared By:

John P. Lehoczky, Principal Investigator
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213
(412) 268-8725

12pp

Principal Investigator Names:

J. Lehoczky	(412) 268-8725	jpl@k.cs.cmu.edu
L. Sha	(412) 268-5875	lrs@sei.cmu.edu
D. Siewiorek	(412) 268-2570	dps@a.cs.cmu.edu
J. Strosnider	(412) 268-6927	jks@usa.ece.cmu.edu
H. Tokuda	(412) 268-7672	hxt@k.cs.cmu.edu

PI Institution: Carnegie Mellon University
Contract Title: Fault Tolerant Real-Time Systems
Contract Number: N00014-92-J-1771[1]
Reporting Period: 1 Oct 91 - 30 Sep 92

1 Productivity Measures

- Papers submitted but not yet accepted: 7
- Refereed papers accepted and in press: 7
- Refereed papers published: 15
- Books submitted or published: 0
- Other reports: 5
- Ph.D. dissertations: 0
- Patents filed or granted: 0
- Invited presentations[2]: 15
- Contributed presentations: 5
- Honors, Prizes and Awards received:
  - **John Lehoczky**:
    - Associate Editor, Journal of *Real-Time Systems*,
    - Member of the program committee of the 12th IEEE Real-Time Systems Symposium, the 8th IEEE Real-Time Operating Systems Workshop, the 11th ICDCS, and the 1992 SIGMETRICS and Performance 92 conferences.
    - Member, NIH Special Study Section on Statistics,
  - **Lui Sha**
    - Member NASA Space Station Advisory Committee,
    - Chairman of the Board of Visitors of RICIS, an R&D center established by NASA and NASA JSC at University of Houston at Clearlake.
    - Program chair, 12th IEEE Real-Time Systems Symposium,
    - General chair, 13th IEEE Real-Time Systems Symposium,
    - Program Committee, 2nd International Workshop on Responsive Systems,

[1]This yearly report is prepared for the October 1, 1991 to September 30, 1992 time period. The October 1, 1991 to March 31, 1992 period was supported by N00014-84-K-0734, and the April 1, 1992 to September 30. 1992 period was supported by N00014-92-J-1771.

[2]The invited and contributed presentations exclude all conference and workshop proceedings which are listed under Publications.

- Associate Editor, *Real-Time Systems*
- Associate Editor, *IEEE Computer*

- **Jay Strosnider**
  - Member of the Program Committee, 13th IEEE Real-Time Systems Symposium
- **Hide Tokuda**
  - Chairman of the 8th Real-Time Operating Systems Workshop.

- Graduate students supported: 1
- Post-docs supported: 0
- Minorities supported: 0

Principal Investigator Names:

J. Lehoczky	(412) 268-8725	jpl@k.cs.cmu.edu
L. Sha	(412) 268-5875	lrs@sei.cmu.edu
D. Siewiorek	(412) 268-2570	dps@a.cs.cmu.edu
J. Strosnider	(412) 268-6927	jks@usa.ece.cmu.edu
H. Tokuda	(412) 268-7672	hxt@k.cs.cmu.edu

PI Institution: Carnegie Mellon University
Contract Title: Fault Tolerant Real-Time Systems
Contract Number: N00014-92-J-1771
Reporting Period: 1 Oct 91 - 30 Sep 92

2 Summary of Technical Progress

2.1 Overview

The ART (Advanced Real-Time Technology) Project of Carnegie Mellon University is engaged in wide ranging research on hard real-time systems. The project has as its overall goal the development and demonstration of predictable and fault tolerant hard real-time computer systems. To achieve this goal, research is being conducted in three interrelated areas:

1. The development of a theory of hard real-time resource management including scheduling and synchronization. In addition, this theory is being applied in a wide range of contexts including databases, communication networks and operating systems.
2. The design and construction of an operating system (ARTS 2.0) that supports predictable and fault tolerant real-time computer systems and demonstration and testing in the ARTS testbed to provide proof of concept and to gain understanding into aspects of the theory needing further refinement. A real-time version of the Mach operating system, RT-Mach 2.0, is also being developed as a part of the project.
3. Development of hardware architectures and approaches to fault tolerance that can support predictable hard real-time computer systems.

The project members are also concerned with the transition of the theory being developed to the user community, especially the U.S. Navy. For this reason, project members have a close relationship with the RMARTS and ZDAK projects in the CMU Software Engineering Institute. Dr. Lui Sha is a member of the ART Project, the RMARTS project and is the head of the ZDAK project and has made great strides in increasing adoption of the rate monotonic scheduling theory in major national projects. In July 1992, Raganathan Rajkumar joined the ART project and the ZDAK project to further strengthen both.

During the October 1, 1991 - September 30, 1992 period, substantial progress was made in each of these broad categories. The progress on resource management and real-time networks is briefly described below. A more detailed collection of briefing materials for the entire project is contained in the yearly *ART Project Review* distributed to ONR representatives.

2.2 Theory of Hard Real-Time Resource Management

The theoretical research is focused on fixed priority scheduling algorithms which include the rate monotonic and deadline monotonic algorithms. This family of algorithms is designed to handle in a predictable fashion many important real-time system characteristics including: a mixture of hard-deadline periodic, hard deadline aperiodic and soft deadline aperiodic tasks, task synchronization and scheduling

distinct resources including processors and communication media. Some extensions to this theory are described next.

2.2.1 Scheduling Tasks with Varying Execution Priorities

This research involves the scheduling of a set of periodic tasks where each task is composed of a set of serially executed subtasks, each of which may have a deadline and may be executed at its own fixed priority level. This model is very flexible and allows one to consider a wide range of practical situations including tasks with non-preemptible sections, task synchronization using priority ceiling emulation, interrupts, operating system overhead, tasks with certain precedence constraints and certain message passing systems. A theoretical analysis of such task sets and the theoretical underpinnings are presented along with a robotics example. The schedulability analysis considers each task individually and reduces it to "canonical form." The other tasks are then divided into one of five categories, and an algorithm for the resulting analysis is presented. Dividing tasks into subtasks, if properly done, offers the possibility of achieving higher levels of task set schedulability. Indeed, it is proved that for task sets of size 2, 100% schedulability can always be achieved. While dividing a task into subtasks is not a recommended procedure, the methods given allow for the analysis of systems where this has been done to achieve schedulability.

2.2.2 Applying Scheduling Theory to Real-Time OS Analysis

We are currently working to include the costs of fixed priority schedulers within the real-time scheduling analysis framework. Several different theoretical implementations of real-time schedulers were considered, including event-driven and timer-driven models, and scheduling models were developed for each. The results of this work show that there is a significant gap between the ideal scheduling analysis, which assumes no cost for preemption, scheduling, or interrupt handling, and the actual implementation of schedulers within an operating system.

We have experimentally verified the scheduling theory described above for the Real-Time (RT) Mach operating system. To do this we first developed a scheduling model for the RT Mach scheduler using one of the basic timer-driven scheduling models. We then characterized the RT Mach scheduler by measuring its costs under worst case scenarios. We then experimentally verified the model by measuring the performance of a task set consisting of real-time threads with tightly controlled timing characteristics. The execution times of the threads were scaled uniformly to their breakdown utilization. The measured performance agreed with the performance predicted by the model to within several percentage points.

2.2.3 Jointly Scheduling Periodic and Aperiodic Tasks

New results were derived for the joint scheduling of hard deadline periodic tasks and soft deadline aperiodic tasks. The objective is to find a fixed priority scheduling algorithm which minimizes the response times of the aperiodic tasks subject to meeting all hard deadlines of all aperiodic tasks. Background service and polling servers are two standard methods for this problem. More recently, server algorithms including the sporadic server and the deferrable server have been shown to offer important performance gains. Now, a new algorithm, called the slack-stealing algorithm, has been developed which has been shown to offer significant performance gains over the sporadic and deferrable servers. The algorithm is optimal in the sense that for any given fixed priority order of the periodic tasks and for a first-come-first-served queueing discipline for the aperiodic tasks, the algorithm gives the minimum response time to every aperiodic task. Servicing in shortest remaining processing time order will generate the shortest average response time for the aperiodic tasks. In addition to the optimality properties, this approach offers additional advantages: (1) it allows for the convenient reclaiming of allocated but unused

processing time by the periodic tasks, (2) it allows for the schedulable servicing of hard deadline aperiodic tasks, and (3) it allows for mixtures of hard and soft deadline aperiodic tasks. This algorithm can provide a method for the servicing of faulty tasks, which gives an analytic basis for temporal redundancy for enhancing fault tolerance. The slack-stealing algorithm can be implemented, but we are in the process of developing versions which are easily implemented with low overhead but which capture nearly all of the performance gains of the slack-stealing approach. The methods also apply to dynamic scheduling algorithms such as the earliest deadline algorithm, and we are in the process of extending the work to this case.

2.3 Real-Time Networks

In the past year we continued our work in the area of real-time networks. Our work this year has two main thrusts: (1) Development of a generic scheduling framework for RT Networks, and (2) analysis of Specific Network Architectures. Various aspects of these thrusts are next described in detail.

2.3.1 Generic Scheduling Framework

We have developed the notion of a "scheduling model" for a network. A scheduling model is an abstraction that facilitates reasoning about the timing correctness of a set of messages on the network. The model incorporates the effects of network components that potentially affect message timing behavior. The scheduling model is useful not only in determining the schedulability of a message set, but also in selecting network parameters such as maximum packet size. When a message set is schedulable under several different scheduling situations (in our case CTR and ETR), a figure of merit is necessary to compare the different situations. To address this need we have introduced the "degree of schedulable saturation," S_{max}, which represents the degree to which the system is saturated from a schedulability viewpoint. A smaller S_{max} indicates greater remaining schedulable capacity. Therefore a particular scheduling situation is said to be "better" if it results in a smaller S_{max}. A scheduling algorithm is infeasible if it yields an S_{max} greater than unity. We have also been able to show that this measure of scheduling performance is equivalent to using the maximum capacity high priority sporadic server as a figure of merit.

Traditional notions of schedulability are insufficient in the analysis of networks such as IEEE 802.5 in ETR mode, FDDI, dual link networks, and high-speed switch based networks where the next packet transmission can begin before a particular packet reaches its destination. In these cases it is more useful to consider the notion of "transmission schedulability." A set of messages is said to be transmission schedulable (t-schedulable) if each message can be transmitted before its deadline. Satisfaction of the end-to-end deadline of the message can be found using the relation: End-to-End Delay = Transmission Deadline + Propagation Delay.

2.3.2 Distributed Version of a Schedulable Routing Algorithm

In the previous year we had developed an algorithm to route real-time messages in a packet switched network. However the algorithm had the short-coming that it was centralized and hence was computationally expensive and required network nodes to have a large memory. We have developed a distributed version of the algorithm that reduces the computational and memory requirements at individual network hosts, compared to the previously developed centralized algorithm. The algorithm routes traffic in an arbitrary topology multihop network, with a connection-oriented network service, such that end-to-end timing guarantees can be made. The degree of schedulability saturation of each link in the network is calculated and a path that traverses the least saturated link is selected. This keeps the load in

the network balanced and minimizes the possibility of saturating any link.

2.3.3 Real-Time Communication On Dual-Link Networks

We have developed a theoretical framework to analyze schedulability of dual link MANs such as IEEE 802.6 (DQDB). Showed that current MAN protocols exhibit unpredictable behavior. The fundamental challenge here is to ensure predictable operation in spite of incomplete information in both time and space. To address the unpredictability problem we introduced the concept of system coherence. A system is coherent if the distributed queues in each station in the network are consistent with some observable ordering of requests. We derive the conditions for system coherence and develop a scheduling model that allows calculation of end-to-end message delay between a source and destination.

2.3.4 Fixed Priority Scheduling With Limited Priority Levels

Priorities provide the basic mechanism that allow scheduling algorithms to discriminate between tasks with different response time and criticality requirements. Networks are usually designed with very few priority levels. Although, previous work has shown that a lack of sufficient priority levels results in a loss of schedulable utilization, the analysis in that work resulted in extremely pessimistic results since it used only sufficient schedulability conditions. We have developed both necessary and sufficient conditions that can be used to analyze the schedulability impact of limited priority levels for any fixed priority algorithm. When the system supports fewer priority levels than a task set's natural priorities, multiple tasks must be grouped into the same system priority. We have developed two metrics to evaluate the scheduling impact of grouping tasks. Finally we have applied our analysis to a case study of a multimedia task set on a network with four priority levels.

2.3.5 Conventional & Early Token Release Scheduling Models For IEEE 802.5

We have developed analytical scheduling models for both the conventional IEEE 802.5 token ring protocol and a recent extension to the original protocol that allows early token release (ETR). We have previously analyzed the conventional token ring protocol from the viewpoint of improving responsiveness of soft deadline aperiodic messages. In contrast, in this work we develop a generic scheduling framework and develop models for both conventional and early token release versions of the protocol. The models can be used to analyze the schedulability of arbitrary periodic message sets. The main contributions of this work are: Scheduling models for both the original protocol and ETR protocol; that capture necessary and sufficient schedulability conditions; comparison for maximum achievable utilizations for the two protocols; comparison between the original protocol and ETR from a schedulability viewpoint. We also demonstrate the utility of our scheduling models to select network operating parameters such as of maximum packet size, and to quantify effects of parameters such as the number of stations, and network size on schedulability.

2.3.6 Responsive Aperiodic Services in High-Speed Networks

High-speed switch based networks need to support highly responsive aperiodic messages for functions such as connection setup/tear-down, link-state updates, topology acquisition and other signaling functions. Scheduling these message in the background causes them to have very long delays which is undesirable. On the other hand giving these messages a high priority may cause timing guarantees of existing aperiodic connections to be violated. In this work, we propose techniques to minimize delay for these signaling messages without jeopardizing guarantees made to existing connections. We demonstrate that our techniques significantly improve signaling messages' responsiveness compared to transmitting them in background mode. Finally, we propose simple hardware implementations for our techniques that can be embedded in the output queues of fast packet switches.

Principal Investigator Names:

J. Lehoczky	(412) 268-8725	jpl@k.cs.cmu.edu
L. Sha	(412) 268-5875	lrs@sei.cmu.edu
D. Siewiorek	(412) 268-2570	dps@a.cs.cmu.edu
J. Strosnider	(412) 268-6927	jks@usa.ece.cmu.edu
H. Tokuda	(412) 268-7672	hxt@k.cs.cmu.edu

PI Institution: Carnegie Mellon University
Contract Title: Fault Tolerant Real-Time Systems
Contract Number: N00014-92-J-1771
Reporting Period: 1 Oct 91 - 30 Sep 92

3 Publications and Presentations

3.1 Published or In Press

- Mercer, C.W. and Tokuda, H., "An evaluation of priority consistency in protocol architectures," *Proceedings of the IEEE 16th Conference on Local Computer Networks*," October 1991.
- Tokuda, H. and Nakajima, T., "Evaluation of real-time synchronization in Real-Time Mach," *Proceedings of the 2nd USENIX Mach Symposium*, October 1991.
- Mercer, C.W. and Tokuda, H., "Priority consistency in protocol architectures," *Proceedings of the 2nd International Workshop on Network Operating System Support for Digital Audio and Video*, November 1991.
- Tokuda, H. and Nakajima, T., "Evaluation of Real-Time Synchronization in Real-Time Mach", *Proceedings of 2nd Mach Symposium*, November 1991.
- Gonzalez Harbour, M., Klein, M. H., Lehoczky, J. P., "Fixed Priority Scheduling of Periodic Tasks with Varying Execution Priority", *Proceedings of the 12th IEEE Workshop on Real-Time Systems Symposium*, December 1991.
- Ramos-Thuel, S. and Strosnider, J.K., "The Transient Server Approach to Scheduling Time-Critical Recover Operations", *Proceedings of the 12th IEEE Workshop on Real-Time Systems Symposium*, December 1991.
- Chou, S. T.-C., and Tokuda, H., "Real-time communication with deadline based scheduling," *Proceedings of the IEEE 12th Workshop on Real-Time Software and Operating Systems*, May 1992.
- Paul, C. J., Holloway, L.E., Strosnider, J.K. and Krogh, B. H., "An intelligent, reactive scheduling system," *IEEE Controls Magazine*, June, 1992.
- Tobe, Y., Chou, S. T.-C., and Tokuda, H., "QOS control for continuous media communications," *Proceedings of INET 92*, June 1992.
- Tokuda, H., Tobe, Y., Chou, S.T.-C., and Moura, J.M.F., "Continuous media communication with dynamic QOS Control using ARTS with an FDDI network," *Proceedings of ACM SIGCOMM 92*, August 1992.
- Nakajima, T. and Tokuda, H., "Implementation of scheduling policies in Real-Time Mach," *Proceedings of the 3rd International Workshop on Object-Oriented Operating Systems*, September 1992.
- Nakajima, T. and Tokuda, H., "Real-time synchronization in Real-Time Mach," *Proceedings of the National Conference of Japan Society for Software Science and Technology*,

September 1992.

- Tokuda, H., Nakajima, T. and Oikawa, S., "Towards a new operating system architecture: microkernel vs. reflective architecture," *Proceedings of the National Conference of Japan Society for Software Science and Technology*, September 1992.
- Oikawa, S., Tokuda, H. and Nakajima, T, "Design and implementation of real-time user-level thread," *Proceedings of the National Conference of Japan Society for Software Science and Technology*, September 1992.
- Tokuda, H., "A real-time thread model for continuous media applications," *Proceedings of the National Conference of Japan Society for Software Science and Technology*, September 1992.
- Lehoczky, J.P. and Ramos-Thuel, S., "An optimal algorithm for scheduling soft-aperiodic tasks in fixed-priority preemptive systems," to appear *Proceedings of the 13th IEEE Real-Time Systems Symposium*, December 1992.
- Sathaye, S.S., Sha, L. and Strosnider, J.K., "Scheduling real-time communications on dual link networks," to appear *Proceedings of the 13th IEEE Real-Time Systems Symposium*, December 1992.
- Mercer, C.W. and Tokuda, H., "Preemptibility in real-time operating systems," to appear in *Proceedings of the 13th IEEE Real-Time Systems Symposium*, December 1992.
- Strosnider, J.K., Lehoczky, J.P. and Sha, L., "Deferrable server algorithm for improved asynchronous response times," to appear *IEEE Transactions on Computers.*
- Sasinowski, J.E. and Strosnider, J.K., "A dynamic programming algorithm for cache/memory partitioning for real-time systems," to appear *IEEE Transactions on Computers.*
- Sha, L. and Rajkumar, R., "Generalized rate monotonic scheduling," to appear in *Encyclopedia of Software Engineering*, (ed. J. Marchiniak), John Wiley and Sons.
- Lehoczky, J.P., "Real-time resource management," to appear in *Encyclopedia of Software Engineering*, (ed. J. Marchiniak), John Wiley and Sons.

3.2 Submitted for Publication and Other Reports

- Harbour, Michael Gonzalez, Klein, Mark H., Lehoczky, John P., "Timing analysis for fixed priority scheduling of hard real-time systems," submitted for publication.
- Katcher, D., Arakawa, H. and Strosnider, J.K., "Engineering and analysis of fixed priority schedulers," submitted for publication.
- Sathaye, S.S., Yee, A., Bianchini, R.P., and Strosnider, J.K., "A distributed routing algorithm for real-time traffic in multi-hop networks," submitted for publication.
- Sathaye, S.S., Sha, L. and Strosnider, J.K., "Analysis of reservation based dual link networks for real-time applications," submitted for publication.
- Sathaye, S.S. and Strosnider, J.K., "A scheduling analysis of the IEEE 802.5 token ring with and without early token release," submitted for publication.
- Mraz, R. and Strosnider, J.K., "Generation and the use of dynamic program traces for RISC pipeline analysis," submitted for publication.**
- Arakawa, H., Katcher, D.I, Strosnider, J.K. and Tokuda, H., "Modeling and validation of the Real-Time Mach scheduler," submitted for publication.**
- Chou, S.T.-C., Nakajima, T. and Tokuda, H., "RTS: A real-time server in real-Time Mach,"

Draft, August 1992.

- Nakajima, T. and Tokuda, H., "Network protocol server on Real-Time Mach," Draft, August, 1992.
- Tokuda, H. and Nakajima, T. "Design and implementation of real-time scheduler in Real-Time Mach," Draft, September 1992.

3.3 Masters Theses

- Tobe, Y., "Quality reservation in video communications - Subband coding and session reservation," Department of Electrical and Computer Engineering, Carnegie Mellon University, December 1991.
- Arakawa, H., "Schedulability analysis of the Real-Time Mach scheduler," Department of Electrical and Computer Engineering, Carnegie Mellon University, August 1992.

Principal Investigator Names:

J. Lehoczky	(412) 268-8725	jpl@k.cs.cmu.edu
L. Sha	(412) 268-5875	lrs@sei.cmu.edu
D. Siewiorek	(412) 268-2570	dps@a.cs.cmu.edu
J. Strosnider	(412) 268-6927	jks@usa.ece.cmu.edu
H. Tokuda	(412) 268-7672	hxt@k.cs.cmu.edu

PI Institution: Carnegie Mellon University
Contract Title: Fault Tolerant Real-Time Systems
Contract Number: N00014-92-J-1771
Reporting Period: 1 Oct 91 - 30 Sep 92

4 Transitions and DoD Interactions

ART project personnel frequently interact with DoD representatives, especially Lui Sha in his dual role as a member of the ART project and of the RMARTS and ZDAK projects of the CMU SEI. Dr. Sha is deeply involved with transitioning rate monotonic scheduling theory to industry and government. His efforts include:

- Member, NASA Space Station Advisory Committee,
- Interaction with the Navy NGCR,
- Named Chairman of the Board of Visitors of RICIS, an R&D center established by NASA and NASA JSC at University of Houston at Clearlake
- Coordinated the real-time version of POSIX,
- Worked with IEEE 802.6 standards group to develop a real-time capability,

In addition, Hide Tokuda, as developer of ARTS (and Real-Time Mach), interacts regularly with NOSC, IBM and University of Virginia to coordinate the development of testbeds at all four sites and experimentation with ARTS.

As a part of our software fault tolerance effort supported by N00014-92-J-1524, we have interacted with MITRE Corporation to investigate the use of analytic redundancy for airborne radar tracking systems.

Finally, the rate monotonic scheduling theory is increasingly being adopted by major projects. These projects include:

- Navy BSY-1 and BSY-2,
- NASA Space Station Freedom (for system integration),
- European Space Station (recommended its use for its Hard Real-Time OS project).

Principal Investigator Names:

J. Lehoczky	(412) 268-8725	jpl@k.cs.cmu.edu
L. Sha	(412) 268-5875	lrs@sei.cmu.edu
D. Siewiorek	(412) 268-2570	dps@a.cs.cmu.edu
J. Strosnider	(412) 268-6927	jks@usa.ece.cmu.edu
H. Tokuda	(412) 268-7672	hxt@k.cs.cmu.edu

PI Institution: Carnegie Mellon University
Contract Title: Fault Tolerant Real-Time Systems
Contract Number: N00014-92-J-1771
Reporting Period: 1 Oct 91 - 30 Sep 92

5 Software and Hardware Prototypes

The ARTS operating system continues to be developed under the direction of Hide Tokuda and has been distributed to NOSC, IBM, University of Virginia and Cornell. The operating system has an associated tool set, Scheduler 1-2-3 (to determine the schedulability of a task set) and ARM (Advanced Real-Time Monitor). Neither ARTS nor the associated tool set have been commercialized. Hide Tokuda is also developing a real-time version of the Mach operating system, RT-Mach 2.0.