REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
		Technical Report 7/1/96-9/30/96

4. TITLE AND SUBTITLE
Applications of the Theory of Distributed and Real Time Systems to the Development of Large-Scale Timing Based Systems

5. FUNDING NUMBERS
C -
F19628-95-0118

6. AUTHOR(S)
Nancy Lynch

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02138

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Department of the Airforce
Electronic Systems Center (AFMC)
Hanscom Air Force Base, MA 01731

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
N/A

12a. DISTRIBUTION / AVAILABILITY STATEMENT
No limits on disclosure.

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

19961031 001

14. SUBJECT TERMS	15. NUMBER OF PAGES
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	**PR**	- Project
G	- Grant	**TA**	- Task
PE	- Program Element	**WU**	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

- **DOD** - See DoDD 5230.24, "Distribution Statements on Technical Documents."
- **DOE** - See authorities.
- **NASA** - See Handbook NHB 2200.2.
- **NTIS** - Leave blank.

Block 12b. Distribution Code.

- **DOD** - Leave blank.
- **DOE** - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
- **NASA** - Leave blank.
- **NTIS** - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

Applications of the Theory of Distributed and Real-Time Systems to the Development of Large-Scale Timing-Based Systems

Quarterly Technical Report
Reporting Period: 7/1/96-9/30/96

Nancy A. Lynch
Theory of Distributed Systems
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139
lynch@theory.lcs.mit.edu

Sponsored by
Advanced Research Projects Agency/CSTO
Applications of the theory of distributed and real time systems to the development of large scale timing-based systems.
Issued by ESC/ENS under Contract #F19628-95-C-0118
ARPA Order No. D014

Members of MIT's Theory of Distributed Systems group continued their work on modelling, designing, verifying and analyzing distributed and real-time systems. The focus is on the study of "building-blocks" for the construction of reliable and efficient systems. Our works falls into three general categories: modelling and verification tools, algorithms and impossibility results, and applications.

I. Modelling and verification tools

- Garland and Lynch continued their work on the design of a Larch interface language for I/O automata. There is now a working parser, which enables the use of Larch to verify invariants and simulation mappings for algorithms written in this language. Various members of the group are testing the language by writing some of their distributed algorithms using it.
  See URL `http://larch.lcs.mit.edu:8001/~garland/ioaLanguage.html`.

- Luchangco worked with Garland, Lynch, and Petrov to fill in some gaps and otherwise polish our Larch proof of the concurrent timestamp system of Dolev and Shavit. The final version will appear in the proceedings of FORTE'96 (as the lead-off paper of the conference).
  See URL `http://theory.lcs.mit.edu/tds/CTSS.html`.

- Lynch, Segala, Vaandrager, and Weinberg continued their work on a full version of their paper on the hybrid I/O automaton (HIOA) model; this is a mathematical model for reasoning about hybrid (continuous/discrete) systems. This quarter, we experimented with several technical generalizations of our original set of axioms.
  For information about the HIOA model, see
  URL `http://theory.lcs.mit.edu/tds/hybrid-model.html`.

- Segala and Lynch continued their development of basic theory to support compositional reasoning about probabilistic systems. This quarter, the focus was on modular techniques for performance analysis.
  See URL `http://theory.lcs.mit.edu/tds/AH.html`.

- Vaziri and Jensen began examining and developing techniques for the integration of model checking and theorem proving methods for verification of concurrent systems.

II. Algorithms and impossibility results

- Della Libera and Shavit completed their work on reactive diffracting trees, a new version of the diffracting tree synchronization primitive that grows and shrinks according to the load on the data structure. They completed a formal correctness proof, and obtained a collection

of experimental results showing that reactive diffracting trees scale well to large numbers of processors. Della Libera's M.S. thesis was completed in July.
See URL `http://theory.lcs.mit.edu/~gio/research.html`.

- Shavit and Zemach started working on a highly concurrent priority queue design based on their earlier "combining forest" data structure. They are currently performing empirical evaluations of the design using the Proteus simulator. Their next step will be design modifications and empirical tests using the Alewife machine here at MIT. Shavit and Zemach also completed their "Diffracting Trees" manuscript, which will appear in ACM TOCS within a month or two. They also worked with Upfal of IBM Almaden on a journal version of their SPAA 96 paper providing a mathematical model for analyzing diffracting tree performance. Finally, they worked on developing a new "wait-free" sorting algorithm, that is, one that will take logarithmic parallel time and will run (though slightly less effectively) even if many processes fail. See URL `http://theory.lcs.mit.edu/~asaph/`.

- Shvartsman completed the manuscript, *A Theory of Fault-Tolerant Parallel Computation.* This monograph synthesizes the latest results for parallel computation in the presence of failures, restarts and delays; these were previously described only in research papers. The monograph deals with several models of processor failures and restarts, it identifies the key problems, and presents algorithms, general simulations and lower bounds for fault-tolerant computation. The manuscript is to be published by Kluwer Academic Press.
See URL `http://theory.lcs.mit.edu/~alex/mono2.html`.

- Touitou and Shvartsman performed a comprehensive suite of simulations, validating the earlier theoretical results of Lynch, Shavit, Shvartsman and Touitiou showing that many important classes of the highly concurrent data structures used for counting and load balancing exhibit nearly linearizable behavior.
See URL `http://theory.lcs.mit.edu/~alex/count2.html`.

- De Prisco and Shvartsman are also developing new and efficient algorithms for the *Do-All* problem of performing n tasks using p message passing processors under the constraint of mainting message and work efficiency. Another manuscript is in preparation.

- Lynch and Rajsbaum generalized their results on the Borowsky-Gafni simulation algorithm for fault-tolerant systems. The new results extend the work to the case where the underlying memory model consists of simple read/write registers rather than powerful atomic snapshot objects. A journal paper is in preparation.
See URL `http://theory.lcs.mit.edu/tds/borowsky.html`.

III. Applications

A. Distributed system building blocks

- Fekete, Kaashoek, and Lynch prepared and submitted a journal paper based on their earlier work on "Implementing Sequentially Consistent Shared Objects Using Broadcast and Point-to-Point Communication". This version contains a more comprehensive treatment of multicast systems.
  See URL `http://theory.lcs.mit.edu/tds/orca.html`.

- Shvartsman and Oleg Cheiner, an M.Eng. student, implemented a prototype distributed algorithm based on the eventually serializable data service of Fekete, Gupta, Luchangco, Lynch and Shvartsman [1]. This prototype will be used as a testbed for exploring optimizations of the algorithm. A Web-client for the distributed algorithm is being developed using cgi.
  See URL `http://theory.lcs.mit.edu/tds/proto.html`.

- Lynch and Shvartsman formulated a specification of a general purpose processor group-oriented communication primitive. They used the primitive to obtain new results and to extend previous results for distributed algorithms that use replicated read/write memory. A manuscript documenting this work is being revised, in preparation for conference submission.

- Fekete, Lynch and Shvartsman are developing specifications for group communication primitives such as those used in the Isis, Transis, Horus and Psynch systems. For example, they have developed specifications for a virtually synchronous group membership service and for a totally ordered broadcast service. They are developing sample (abstract) applications of these services, proving these correct, and analyzing their complexity and fault-tolerance.

- Vaziri completed the proof of correctness for a controller algorithm for the RAID Level 5 system. This work has helped to clarify work of Courtright and Gibson at CMU. For example, it has uncovered an error in the RAID Level 6 design, and also another situation where more constraints on concurrency than necessary were used. Her M.S. thesis was completed in August.
  See URL `http://theory.lcs.mit.edu/~vaziri/raid.html`.

- De Prisco has continued his work on modelling, improving, and verifying the practical Paxos algorithm for fault-tolerant distributed consensus. Work this quarter involved developing, verifying and analyzing all the subsidiary algorithms, plus developing an application of the Paxos algorithm to replicated data management. Work remains in completing some aspects of the timing model, plus carrying out the complexity and fault-tolerance analysis.
  See URL `http:theory.lcs.mit.edu/~robdep/research.html`.

- Luchangco has begun research on memory consistency models, trying to understand the various memory specifications and program restrictions that have been proposed, and how they compare with each other. For instance, he is seeking theorems that show that certain

classes of programs can be shown to run correctly on certain types of weakly consistent memory.

B. Transit

- Weinberg and Lynch wrote a paper based on their work on modelling vehicle deceleration maneuvers. The paper was accepted to RTSS '96.
  See URL `http://theory.lcs.mit.edu/~hbw/decel.html`.

- Lynch and Dolginova worked with Branicky on modular safety analysis for the platoon join maneuver of the California PATH intelligent highway project. The system is modelled using HIOAs, and its properties are proved using a combination of standard methods for reasoning about continuous systems (mathematical analysis), and HIOA techniques (levels of abstractions, invariants, simulation mappings). So far, analysis has been completed for an ideal case with no delays or uncertainties, a more complicated case with sensor and acceleration/breaking delays, and the complex, but more realistic case which incorporates both kinds of delays, and sensor uncertainties. Future work will involve analyzing multicar collisions. Two papers have been written: one for the Hybrid Systems workshop in Ithaca, Oct. '96 (accepted), and one for the workshop in Grenoble, in Mar. '97
  See URL `http://theory.lcs.mit.edu/~katya/progress2.html`.

- Livadas has continued his work on the formal modelling of vehicle protection (VP) subsystems, as used in the Raytheon Personal Rapid Transit project. Recently, the model has been augmented to allow the VPs to retract protective actions previously issued. The new model allows simple composition of protectors that depend on each other's correct operation. Correctness proofs are nearly complete for protectors preventing overspeed and collisions on a single straight track, and are in preliminary stages for the more general case involving several tracks interconnected by Y shaped merges and diverges.
  See URL `http://theory.lcs.mit.edu/~clivadas/research.html`.

C. Communication

- Several months ago, Smith discovered that T/TCP did not implement TCP – in fact, it can deliver duplicate data. Discussions with protocol designers suggested that this behavior was not so bad, so Smith developed a weaker specification that captures the guarantees that T/TCP actually makes. He is working on showing that T/TCP satisfies this weaker specification. Smith also proved an impossibility result – that in the absence of certain timing guarantees, it is impossible for any protocol to satisfy the strong specification that

TCP satisfies and still provide the efficiency that T/TCP claimed. A paper on his proof of TCP is to appear in the proceedings of FORTE '96.
See URL `http://theory.lcs.mit.edu/~mass/comm.html` and
URL `http://theory.lcs.mit.edu/~mass/papers.html`.

D. Probabilistic Systems

- Lynch and Segala working intensively during the summer, attempting to complete the work of Pogosyants and Segala on modelling and proof of the (randomized) Aspnes-Herlihy consensus protocol. Now the modelling is completed, and there is a good draft of the entire proof, including both safety and probabilistic performance properties. More polishing is still needed. See URL `http://theory.lcs.mit.edu/tds/AH.html`.

References

[1] Alan Fekete, David Gupta, Victor Luchangco, Nancy Lynch, and Alex Shvartsman. Eventually-serializable data services. In *Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing*, pages 300–309, Philadelphia, PA, May 1996.