
Supplementary Reading List

1. Other distributed algorithms textbooks

- [1] Hagit Attiya and Jennifer Welch. *Distributed Computing: Fundamentals, Simulations, and Advanced Topics*. John Wiley and Sons, Inc., 2004. Second Edition.
- [2] Maurice Herlihy and Nir Shavit. *The Art of Multiprocessor Programming*. Elsevier, Inc., 2008.
Note: This book is expected to be published (by Elsevier) mid-semester. However, the authors have, in the meantime, made a pdf of a book chapter available for download at <http://courses.csail.mit.edu/6.852/08/papers/lists-book-chapter.pdf>.
Publisher info at http://www.elsevier.com/wps/find/bookdescription.cws_home/714091/.
- [3] Shlomi Dolev. *Self-Stabilization*. MIT Press, 2000.

2. Dijkstra Prize papers

In each of the years 2000-2007, a prize has been awarded to a research paper that has had a strong impact on research in the area of distributed algorithms. The prize was originally called the “PODC Influential Paper Award”. After the death of Edsger Dijkstra, one of the pioneers of the field, in August, 2002, the prize was renamed the “Dijkstra Prize”.

We will study the key contributions of all eight of these papers during this semester. In case you want to read the original papers for yourselves, here is a list:

- [4] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558-565, July, 1978.
<http://research.microsoft.com/users/lamport/pubs/time-clocks.pdf>.
- [5] M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM*, 32(2):374-382, April, 1985.
<http://groups.csail.mit.edu/tds/papers/Lynch/jacm85.pdf>.
- [6] E. W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Communications of the ACM*, 17(11):643-644, November, 1974.
<http://courses.csail.mit.edu/6.852/08/papers/p643-Dijkstra.pdf>.
- [7] M. Herlihy. Wait-free synchronization. *ACM Transactions on Programming Languages and Systems*, 13(1):124-149, January, 1991.
<http://courses.csail.mit.edu/6.852/08/papers/p124-herlihy.pdf>.
- [8] R. G. Gallager, P. A. Humblet, and P. M. Spira. A distributed algorithm for minimum-weight spanning trees. *ACM Transactions on Programming Language Systems*, vol. 5, pp. 66-77, 1983.
<http://courses.csail.mit.edu/6.852/08/papers/p66-gallager.pdf>.
- [9] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228-234, April, 1980.
<http://research.microsoft.com/users/lamport/pubs/reaching.pdf>.
- [10] J. M. Mellor-Crummey and M. L. Scott. Algorithms for scalable synchronization on shared-memory multiprocessors. *ACM Transactions on Computer Systems*, 9(1):21-65, February, 1991.
<http://courses.csail.mit.edu/6.852/08/papers/p21-mellor-crummey.pdf>.
- [11] C. Dwork, N. Lynch, and L. Stockmeyer. Consensus in the presence of partial synchrony. *Journal of the ACM*, 35(2):288-323, April 1988.
<http://groups.csail.mit.edu/tds/papers/Lynch/jacm88.pdf>.

3. Synchronous networks

- [12] Marcos Kawazoe Aguilera and Sam Toueg. A simple bivalency proof that t -resilient consensus requires $t + 1$ rounds. *Information Processing Letters*, 71(3-4):155-158, August 1999.
<http://courses.csail.mit.edu/6.852/08/papers/IPL-AguileraToueg.pdf>.
- [13] I. Keidar and S. Rajsbaum. A Simple Proof of the Uniform Consensus Synchronous Lower Bound. In *Information Processing Letters (IPL)*, 85(1), pages 47-52, January 2003.
<http://www.ee.technion.ac.il/~idish/Abstracts/uniform-bound.html>.

4. Asynchronous networks

- [14] Friedemann Mattern. Virtual time and global states of distributed systems. In Michel Cosnard et al., editors, *Parallel and Distributed Algorithms: Proceedings of the International Workshop on Parallel and Distributed Algorithms* (Chateau de Bonas, Gers, France, October, 1988), pages 215-226. North Holland, 1989. (Reprinted in: Z. Yang, T.A. Marsland (Eds.), “Global States and Time in Distributed Systems”, IEEE, 1994, pp. 123-133.)
<http://courses.csail.mit.edu/6.852/08/papers/VirtTimeGlobStatesFull.pdf>.
- [15] Colin Fidge. Logical time in distributed computing systems. *IEEE Computer*, 24(8):28-33, August 1991.
<http://courses.csail.mit.edu/6.852/08/papers/ieeecomputer-fidge.pdf>.
- [16] Soma Chaudhuri. More Choices Allow More Faults: Set Consensus Problems in Totally Asynchronous Systems. *Information and Computation* 105 (1), pp. 132-158, July 1993.
<http://courses.csail.mit.edu/6.852/08/papers/Chaudhuri-iandc.pdf>.

5. Asynchronous shared memory

5.1 Mutual exclusion

The following paper and thesis chapter present a new, fundamental lower bound for the time required to achieve mutual exclusion.

- [17] Rui Fan and Nancy Lynch. An $\Omega(n \log n)$ Lower Bound on the Cost of Mutual Exclusion. *Proceedings of the Twenty-Fifth Annual Symposium on Principles of Distributed Computing (PODC'06)*, Denver, Colorado, July 2006. Best Student Paper Award.
<http://groups.csail.mit.edu/tds/papers/Fan/podc06.pdf>.
- [18] Rui Fan. *Lower Bounds in Distributed Computing*. PhD Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, February 2008. Chapter 4, Mutual Exclusion.
<http://people.csail.mit.edu/rfan/rfan.thesis08.pdf>.

5.2 Wait-free computability and the wait-free consensus hierarchy

This paper popularized the notion of wait-free computability, and also introduced the wait-free consensus hierarchy.

- [19] Maurice Herlihy. Wait-free synchronization. *ACM Transactions on Programming Languages and Systems*, 13(1):124-149, January 1991.
<http://courses.csail.mit.edu/6.852/08/papers/p124-herlihy.pdf>.
The paper presents an interesting observation about the wait-free consensus hierarchy.
- [20] Prasad Jayanti. Robust wait-free hierarchies. *Journal of the ACM*, 44(4): 592-614, 1997.
<http://courses.csail.mit.edu/6.852/08/papers/p592-jayanti.pdf>.

5.3 Wait-free vs. f -fault-tolerant data objects

- [21] T.D. Chandra, V. Hadzilacos, P. Jayanti and S. Toueg. Wait-freedom vs. t -resiliency and the robustness of the h_m^r hierarchy. *Proceedings of the 13th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, pages 334-343, Los Angeles, CA, August 1994.
<http://courses.csail.mit.edu/6.852/08/papers/p334-chandra.pdf>.
- [22] T.D. Chandra, V. Hadzilacos, P. Jayanti, and S. Toueg. Generalized irreducibility of consensus and the equivalence of t -resilient and wait-free implementations of consensus. *SIAM Journal on Computing*, 34(2):333-357, 2004.
<http://courses.csail.mit.edu/6.852/08/papers/CHJT.pdf>.
- [23] Elizabeth Borowsky, Eli Gafni, Nancy Lynch, and Sergio Rajsbaum. The BG distributed simulation algorithm. *Distributed Computing*, 14:127-146, 2001.
<http://courses.csail.mit.edu/6.852/08/papers/bg-dist-sim.pdf>.
- [24] Paul Attie, Rachid Guerraoui, Petr Kouznetsov, Nancy Lynch, and Sergio Rajsbaum. The Impossibility of boosting distributed service resilience. Submitted for journal publication, January 2007.
<http://courses.csail.mit.edu/6.852/08/papers/boosting-jacm-submit.pdf>.

5.4 Failure detectors, consensus, and set consensus

The idea of a “failure detector” was introduced in the following paper, which also shows how certain failure detectors can be used to solve consensus.

- [25] Tushar Deepak Chandra and Sam Toueg. Unreliable failure detectors for reliable distributed systems. *Journal of the ACM*, 43(2):225–267, March 1996.
<http://courses.csail.mit.edu/6.852/08/papers/CT96-JACM.pdf>.

Lamport’s “Paxos” paper solves consensus, essentially assuming an underlying failure detector service:

- [26] L. Lamport. The part-time parliament. *ACM Transactions on Computer Systems*, 16(2):133–169, May 1998. Also Research Report 49, Digital Equipment Corporation Systems Research Center, Palo Alto, CA, September 1989.
<http://research.microsoft.com/users/lamport/pubs/lamport-paxos.pdf>.

The following paper proves that a certain failure detector is provably “weakest” for solving consensus:

- [27] Tushar Deepak Chandra, Vassos Hadzilacos, and Sam Toueg. The weakest failure detector for solving consensus. *Journal of the ACM*, 43(4):685–722, July 1996.
<http://courses.csail.mit.edu/6.852/08/papers/jacm96-cht.pdf>.

The following new paper studies failure detectors for set consensus:

- [28] Rachid Guerraoui, Maurice Herlihy, Petr Kouznetsov, Nancy Lynch, and Calvin Newport. On the weakest failure detector ever. *Proceedings of the Twenty-Sixth Annual ACM Symposium on the Principles of Distributed Computing (PODC)*, Portland, Oregon, August 2007.
<http://courses.csail.mit.edu/6.852/08/papers/p235-guerraoui.pdf>.

6. Multiprocessor programming

The brand-new Herlihy-Shavit textbook, cited above, covers the principles of multiprocessor programming quite thoroughly.

This paper introduced the MCS queue lock, which is fast, scalable and fair in a wide variety of multiprocessor systems.

- [29] J. M. Mellor-Crummey and M. L. Scott. Algorithms for scalable synchronization on shared-memory multiprocessors. *ACM Transactions on Computer Systems*, 9(1):21-65, February, 1991
<http://courses.csail.mit.edu/6.852/08/papers/p21-mellor-crummey.pdf>.
This paper presents a variant on queue locks for systems in which processors are clustered, and memory access is much more expensive across clusters than within a cluster.
- [30] Victor Luchangco, Daniel Nussbaum and Nir Shavit. A Hierarchical CLH Queue Lock. *Proceedings of the European Conference on Parallel Computing (EuroPar 2006)*, Dresden, Germany, pages 801-810, 2006.
<http://courses.csail.mit.edu/6.852/08/papers/CLH.pdf>.
This paper presents scalable algorithms for nonzero indicators, which are counter-like data structures whose read operation indicates only whether the value is nonzero.
- [31] Faith Ellen, Yossi Lev, Victor Luchangco and Mark Moir. SNZI: Scalable NonZero Indicators. *Annual ACM Symposium on Principles of Distributed Computing*, pages 13-22, 2007.
<http://courses.csail.mit.edu/6.852/08/papers/p13-ellen.pdf>.
This paper presents the first software transactional memory with reasonable performance characteristics for real systems.
- [32] Maurice Herlihy, Victor Luchangco, Mark Moir, and William N. Scherer III. Software Transactional Memory for Dynamic-Sized Data Structures. *Annual ACM Symposium on Principles of Distributed Computing*, pages 92-101, 2003.
<http://www.cs.brown.edu/~mph/HerlihyLM03b/main.pdf>.
The following paper presents an efficient lock-based transactional memory:
- [33] David Dice, Ori Shalev and Nir Shavit. Transactional Locking II. *International Conference on Distributed Computing*, 2006.
http://www.cs.tau.ac.il/~Eshanir/nir-pubs-web/Papers/Transactional_Locking.pdf.

7. Self-stabilization

Dijkstra's breakthrough paper originated the idea of distributed algorithm self-stabilization:

- [34] Edsger W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Communications of the ACM*, 17(11):643-644, November 1974.
<http://courses.csail.mit.edu/6.852/08/papers/p643-Dijkstra.pdf>.
The Dolev book, cited above, contains everything you might want to know about basic self-stabilizing distributed algorithms.

8. Partially synchronous systems

The Attiya-Welch book, cited above, contains a chapter on basic clock synchronization algorithms.

The following paper and thesis contain a lower bound on "gradient" clock synchronization. The thesis chapter is more up-to-date than the journal paper.

- [35] Rui Fan and Nancy Lynch. Gradient Clock Synchronization. *Distributed Computing*, 18(4):255-266, March, 2006.
<http://groups.csail.mit.edu/tds/papers/Fan/FanLynch-gradient.pdf>.
- [36] Rui Fan. *Lower Bounds in Distributed Computing*. PhD Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, February 2008. Chapter 2, Gradient Clock Lower Bound.
<http://people.csail.mit.edu/rfan/rfan.thesis08.pdf>.

The following monograph contains basic formal modeling and proof methods for timing-based systems. It provides the mathematical foundation for the Tempo toolset that we will use in this course.

- [37] Dilsun K. Kaynar, Nancy Lynch, Roberto Segala, and Frits Vaandrager. The Theory of Timed I/O Automata. *Synthesis Lectures on Computer Science*, Morgan Claypool Publishers, 2006. Revised and shortened version of Technical Report MIT-LCS-TR-917a (from 2004).
<http://groups.csail.mit.edu/tds/papers/Kirli/mainfinal.pdf>.

9. Dynamic distributed algorithms

The following papers describe some algorithms for mobile ad hoc networks (MANETs). The first paper implements atomic objects in MANETs, and the other three develop a kind of abstraction layer for programming MANETs. The second paper shows how to implement atomic objects using this abstraction layer.

- [38] Seth Gilbert, Nancy Lynch, and Alex Shvartsman. RAMBO: A Robust, Reconfigurable Atomic Memory service for Dynamic Networks. Submitted for journal publication.
<http://groups.csail.mit.edu/tds/papers/Gilbert/rambo-journal2.pdf>.
- [39] Shlomi Dolev, Seth Gilbert, Nancy A. Lynch, Alex A. Shvartsman, and Jennifer L. Welch. GeoQuorums: Implementing Atomic Memory in Mobile Ad Hoc Networks. *Distributed Computing, Special Issue DISC03*, 18(2):125-155, 2005. <http://groups.csail.mit.edu/tds/papers/Gilbert/TR-900a.ps>.
- [40] Shlomi Dolev, Seth Gilbert, Limor Lahiani, Nancy Lynch, and Tina Nolte. Timed Virtual Stationary Automata for Mobile Networks. Technical Report MIT-LCS-TR-979a, MIT CSAIL, Cambridge, MA, August 2005.
<http://groups.csail.mit.edu/tds/papers/Nolte/TR979a.ps>.
- [41] Matthew Brown, Seth Gilbert, Nancy Lynch, Calvin Newport, Tina Nolte, and Michael Spindel. The Virtual Node Layer: A Programming Abstraction for Wireless Sensor Networks. *Proceedings of the the International Workshop on Wireless Sensor Network Architecture (WWSNA)*, Cambridge, MA, April, 2007.
<http://groups.csail.mit.edu/tds/papers/Gilbert/BrownEtAl.pdf>.