Application Areas and Preliminary Sources

In all the application areas described below, the papers listed should be regarded only as hints. Students are free (and encouraged) to find other algorithms in these general areas that are of special interest to them, and that seem amenable to formal study.

1 Data management in distributed networks

This application area involves maintaining data coherently in distributed networks. For availability and performance, the data is generally replicated, which means that protocols are needed to manage consistency among the replicas. Interesting properties to study will include basic coherence properties for the data, fault-tolerance properties, and performance bounds.

A simple "strong caching" algorithm was introduced in 6.042, and shown to implement an "atomic object".

Nancy Lynch. 6.042 lecture notes, Fall 2000. Lecture 10. `casestudies/6042-L10.ps.`

The following note presents three basic replicated data algorithms: the strong caching algorithm, a majority voting algorithm, and a version of Lamport's Replicated State Machine algorithm:

Andrej Bogdanov, Laura Dean, and Christine Karlovich. Distributed memory algorithms coded in ioa: Challenge problems for software analysis and synthesis, January 2001. Manuscript. `casestudies/Bogdanov.ps`.

Attiya, et al. present a very asynchronous majority voting algorithm. This can be extended to allow multiple writers, and to handle general read/write quorums:

Hagit Attiya, Amotz Bar-Noy, and Danny Dolev. Sharing memory robustly in message-passing systems. *Journal of the ACM*, 42(1):124–142, January 1995. `casestudies/p124-attiya.pdf`.

More difficult algorithms involve maintaining coherent data in the presence of system reconfiguration:

I. Keidar and D. Dolev. Efficient message ordering in dynamic networks. In *15th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 68–76, May 1996. `casestudies/corel-podc96.ps`.

Alan Fekete, Nancy Lynch, and Alex Shvartsman. Specifying and using a partitionable group communication service. *ACM Transactions on Computer Systems*. To appear. `casestudies/TOCS.ps`.

R. De Prisco, A. Fekete, N. Lynch, and A.A. Shvartsman. A dynamic primary configuration group communication service. In Prasad Jayanti, editor, *Distributed Computing* Proceedings of DISC'99 - 13th International Symposium on Distributed Computing, Bratislava, Slovak Republic, September 1999, volume 1693 of *Lecture Notes in Computer Science*, pages 64–78, Bratislava, Slovak Republic, 1999. Springer-Verlag-Heidelberg. `casestudies/disc99.ps`.

Roberto DePrisco and Nancy Lynch. An algorithm for replicated atomic objects. Submitted for publication. `casestudies/deprisco.ps`.

Byzantine failures are considered in:

Miguel Castro. *Practical Byzantine Fault Tolerance.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, November 2000.

2 Reliable network communication

This application area involves a range of protocols, at different layers, to implement reliable point-to-point or multicast communication in unreliable networks. Some of these protocols provide interesting safety guarantees; we can also study performance properties, under a range of assumptions about the behavior of the underlying network and about the input load. Since there are so many communication protocols around, students have a lot of options in choosing subjects to study formally.

Simple examples include a send-receive channel and a multicast channel, as described, for example, in:

> Nancy Lynch. *Distributed Algorithms.* Morgan Kaufmann Publishers, Inc., San Mateo, CA, March 1996. Chapter 14.

Another simple example is a "totally ordered multicast" channel that delivers messages everywhere, in a consistent order, as specified, for example, in:

> Alan Fekete, Nancy Lynch, and Alex Shvartsman. Specifying and using a partitionable group communication service. *ACM Transactions on Computer Systems.*
> To appear. `casestudies/TOCS.ps`.

An "obligatory" example for formal methods for communication is the Alternating Bit Protocol, described, e.g., in:

> K.A. Bartlett, R.A. Scantlebury, and P.T. Wilkinson. A note on reliable full-duplex transmission over half-duplex links. *Communications of the ACM*, 12(5):260–261, May 1969.

> Nancy Lynch. *Distributed Algorithms.* Morgan Kaufmann Publishers, Inc., San Mateo, CA, March 1996. Chapter 22.

> Leslie Lamport. Specifying concurrent systems with TLA+. Preliminary draft, December 2000. Chapter 13.

The connection set-up part of TCP has interesting safety and liveness properties:

> Douglas E. Comer. *Internetworking with TCP/IP: Principles, Protocols, and Architectures*, volume 1. Prentice-Hall, 2000.

> Douglas E. Comer and David L. Stevens. *Internetworking with TCP/IP: Client-Server Programming and Applications*, volume 3. Prentice-Hall, 2001.

> Nancy Lynch. *Distributed Algorithms.* Morgan Kaufmann Publishers, Inc., San Mateo, CA, March 1996. Chapter 22.

Protocols that build and maintain spanning trees form an important part of many communication systems:

> Chris Luhrs. Distributed spanning tree algorithms coded in IOA: Challenge problems for software analysis and synthesis methods ver. 1.3, October 2000. Manuscript. `casestudies/Luhrs-paper1.3.ps`.

> William Adjie-Winoto. A self-configuring resolver architecture for resource discovery and routing in device networks. Master's thesis, Massachusetts Institute of Technology, May 2000.

Now, focusing on multicast, and progressing from protocols with fewer guarantees to protocols with more guarantees:
IP multicast protocols for routing and group membership are described in:

> Stephen E. Deering and David R. Cheriton. Multicast Routing in Datagram Internetworks and Extended LANs. *ACM Transactions on Computer Systems (TOCS)*, 8(2):85–110, May 1990.

Stephen E. Deering. Host Extensions for IP Multicasting. Technical Report (RFC1112), Internet Engineering Task Force (IETF), August 1989. Obsoletes RFCs 988 and 1054.

William C. Fenner. Internet Group Management Protocol, Version 2. Technical Report (RFC2236), Internet Engineering Task Force (IETF), November 1997. Updates RFC 1112.

Brad Cain, Steve Deering, Bill Fenner, Isidor Kouvelas, and Ajit Thyagarajan. Internet Group Management Protocol, Version 3. Internet-Draft, Internet Engineering Task Force (IETF), November 2000. Proposed Update to RFC 2236.

Chuck Semeria and Tom Maufer. Intoduction to IP Multicast Routing. Internet-Draft (Informational), Internet Engineernig Task Force, July 1997. Also, Technical Memo, Networking Solutions Center, 3Com Corporation.

Scalable Reliable Multicast (SRM), Bimodal Multicast, and similar algorithms try hard to deliver multicast messages, but do not make absolute guarantees. It should be interesting to formulate some claims about what they guarantee.

Sally Floyd, Van Jacobson, Steve McCanne, Ching-Gung Liu, and Lixia Zhang. A reliable multicast framework for light-weight sessions and application level framing. *IEEE/ACM Transactions on Networking*, 5(6), December 1997.

Kenneth Birman, Mark Hayden, Oznur Ozkasap, Zhen Xiao, Mihai Budiu, and Yaron Minsky. Bimodal multicast. *ACM Transactions on Computer Systems*, 17(2), May 1999.

Christos Papadopoulos, Guru Parulkar, and George Varghese. An error control scheme for large-scale multicast applications. In *Proc. 17th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE/INFOCOM'98)*, volume 3, pages 1188–1196, March 1998.

Group communication algorithms guarantee strong (view-oriented) consistency and reliability conditions; see, e.g.:

I. Keidar and D. Dolev. Efficient message ordering in dynamic networks. In *15th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 68–76, May 1996. `casestudies/corel-podc96.ps`.

Roberto DePrisco, Alan Fekete, Nancy Lynch, and Alex Shvartsman. A dynamic view-oriented group communication service. In *Proceedings of the 17th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, pages 227–236, Puerto Vallarta, Mexico, June-July 1998. Also, technical memo in progress. `casestudies/DVS.ps`.

3 Algorithms and services for mobile communication systems

Mobile communication leads to a new set of communication algorithms. Some of these deal with basic communication tasks like locating participants and routing messages to them:

Elizabeth M. Royer and Charles E. Perkins. Multicast operations of the ad-hoc on-demand distance vector routing protocol. In *MOBICOM '99, The Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 207–218, Seattle, Washington, USA, August 1999. `casestudies/royerperkins.pdf`.

Brad Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the ACM/IEEE Sixth Annual International Conference on Mobile Computing and Networking (Mobicom 2000)*, Boston, MA, 2000.

P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM99)*, 1999.

David B. Johnson. Routing in ad hoc networks of mobile hosts. In *IBM Mobile Computing Workshop*, 1994.

Josh Broch, David Johnson, and David Maltz. The Dynamic Source Routing protocol for mobile ad hoc networks. Internet draft (work in progress), Internet Engineering Task Force (IETF), October 1999. `www.ietf.org/internet-drafts/draft-ietf-manet-dsr-03.txt`.

Charles E. Perkins and Pravin Bhagwat. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. In *SIGCOMM'94 - ACM Conference on Communications, Architectures, Protocols, and Applications*, pages 234–244, London, England, August/September 1994.

Charles Perkins, Elizabeth Royer, and Samir R. Das. Ad hoc On demand Distance Vector (AODV) routing. Internet draft (work in progress), October 1999. `www.ietf.org/internet-drafts/draft-ietf-manet-aodv-04.txt`.

Vincent D. Park and M. Scott Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of the Sixteenth IEEE Conference on Computer Communications (INFOCOM'97)*, pages 1405–1413, Kobe, Japan, April 1997.

Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris. A scalable location service for geographic ad hoc routing. In *ACM Mobicom 2000*, Boston, MA, 2000.

Other problems, such as leader election and resource allocation, can also be addressed in this new setting:

Jennifer Walter, Jennifer L. Welch, and Nitin Vaidya. A mutual exclusion algorithm for ad hoc mobile networks. To appear in a special issue of *Telecommunication Systems Journal*, 2001. Also, In *Second International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M workshop)*, Dallas, Texas, October, 1998). `casestudies/balt.ps`.

Kostas P. Hatzis, George P. Pentaris, Paul G. Spirakis, Vasilis T. Tampakas, and Richard B. Tan. Fundamental control algorithms in mobile networks. In *Eleventh Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 251–260, Saint-Malo, France, June 1999.

Navneet Malpani, Jennifer L. Welch, and Nitin Vaidya. Leader election algorithms for mobile ad hoc networks. In *Fourth International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)*, Boston, MA, August 2000. `casestudies/dialm00.ps`.

The following deals with group-oriented communication in mobile networks:

Navneet Malpani, Nitin Vaidya, and Jennifer Welch. Distributed token circulation in mobile ad hoc networks. Submitted for publication. `casestudies/Mobicom01-sub.ps`.

The Oxygen project may also provide some interesting case studies for design of mobile applications.

An interesting aspect of this case study will be to try to express some guarantees about what the algorithms do in cases where the participants are constantly moving, perhaps with some bounds on their speed.

4 Automated transportation systems

This case study involves hybrid systems, which combine discrete and continuous behavior. Some preliminary hybrid system modeling examples include a simple delay buffer that just replicates its (discrete and continuous) input, but time d later; a simple audio control protocol:

Doeko Bosscher, Indra Polak, and Frits Vaandrager. Verification of an audio control protocol. In H. Langmaack, W.-P. de Roever, and J. Vytopil, editors, *Proceedings of the Third International School and Symposium on Formal Techniques in Real Time and Fault Tolerant Systems. (FTRTFT'94)*, volume 863 of *Lecture Notes in Computer Science*, pages 170–192, Lübeck, Germany, September 1994. Springer-Verlag. Full version available as Report CS-R9445, CWI, Amsterdam, July 1994.

and an aircraft altitude switch:

> Nancy G. Leveson. Sample intent specification: Altitude switch. Massachusetts Institute of Technology, Safeware Engineering Corporation. `casestudies/ASW-leveson.ps`.
>
> Nancy G. Leveson. Completeness in Formal Specification Language Design for Process Control Systems. *Proceeedings of Formal Methods in Software Practice*, August 2000. To appear.

In general, the kinds of properties one can prove for hybrid systems include "safety" properties in a real-world sense, time bounds, and properties saying that values of certain real-valued variables stay within some limits or track some trajectories. Automated transportation provides good examples for modeling and analysis. Some models for simple vehicle maneuvers are given in:

> H. B. Weinberg and Nancy Lynch. Correctness of vehicle control systems: A case study. In *17th IEEE Real-Time Systems Symposium*, pages 62–72, Washington, D. C., December 1996. `casestudies/RTSS96.ps`.
>
> Carolos Livadas and Nancy A. Lynch. Formal verification of safety-critical hybrid systems. In S. Sastry and T.A. Henzinger, editors, *Hybrid Systems: Computation and Control* (First International Workshop, HSCC'98, Berkeley, CA, USA, April, 1998), number 1386 in Lecture Notes in Computer Science, pages 253–272. Springer Verlag, 1998. `casestudies/HS98.ps`.
>
> Ekaterina Dolginova and Nancy Lynch. Safety verification for automated platoon maneuvers: A case study. In Oded Maler, editor, *Hybrid and Real-Time Systems* (International Workshop, HART'97, Grenoble, France, March 1997), volume 1201 of *Lecture Notes in Computer Science*, pages 154–170. Springer-Verlag, 1997. `casestudies/platoons.ps`.
>
> John Lygeros and Nancy Lynch. Strings of vehicles: Modeling and safety conditions. In S. Sastry and T.A. Henzinger, editors, *Hybrid Systems: Computation and Control* (First International Workshop, HSCC'98, Berkeley, CA, USA, April, 1998), number 1386 in Lecture Notes in Computer Science, pages 273–288. Springer Verlag, 1998. `casestudies/strings.ps`.

Some more complicated examples arise in air-traffic management:

> Carolos Livadas and Nancy A. Lynch. Formal verification of safety-critical hybrid systems. In S. Sastry and T.A. Henzinger, editors, *Hybrid Systems: Computation and Control* (First International Workshop, HSCC'98, Berkeley, CA, USA, April, 1998), number 1386 in Lecture Notes in Computer Science, pages 253–272. Springer Verlag, 1998. `casestudies/HS98.ps`.
>
> Daniel Jackson and John Chapin. Redesigning air-traffic control: A case study in software design. *IEEE Software*, May/June 2000. `sdg.lcs.mit.edu/~dnj/publications.html#6894`.

5 Object-oriented programming

This case study has a different character from the others. It isn't mostly about modeling and analyzing a collection of algorithms. Rather, the problem is to develop techniques for modeling program objects and their interactions, in object-oriented programs (think Java). The techniques should allow for reasoning about the behavior of the program, not just about object interfaces. Specific examples that arise in the course of developing these general methods can be used as case studies for the math methods and tools of this course.

Traditional forms of data abstraction treat the case of self-contained concrete objects implementing self-contained abstract objects. See, e.g.,

> Barbara Liskov and John Guttag. *Program Development in Java: Abstraction, Specification, and Object-Oriented Design*. Addison-Wesley, 2001.

These two levels, and the connection between them, can be modeled using state machines. See, e.g.

> Nancy Lynch. 6.042 lecture notes, Fall 2000. Lecture 10. `casestudies/6042-L10.ps`.

However, the traditional frameworks are not adequate for representing complications such as: objects that are not self-contained, but interact with (or "contain") other objects; mutability/immutability; and data sharing.

For starters, we might consider an abstract "assemblage" of finite sequences (or queues) that share tails. Since this sharing is visible to programs that use the queues in the assemblage, it must be represented in any abstract model of the assemblage. Such an assemblage can be modeled as a state machine. A variety of implementations are possible (e.g., using sharing of representations, or using separate representations with consistency constraints); these can be modeled and shown to implement the specification. Note that this assemblage is different (abstractly) from an assemblage of queues that don't share tails; different implementations would work for this kind of abstraction.

We might consider containment relations, e.g., between "container" objects and the objects they "contain". What does it mean for one container object to implement another? We need models for mutable and immutable collections of mutable and immutable objects, and results about which implement which others. State machines should be able to help.

More elaborate patterns of object interactions can be found, for example, in:

> Erich Gamma, Richard Helm, Ralph Johnson, and Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software.* Professional Computing. Addison-Wesley, 1994.