

References

- Abadi, M. and P. Rogaway (2002). Reconciling two views of cryptography (the computational soundness of formal encryption). *Journal of Cryptology* 15(2), 103–127.
- Abd-El-Malek, M., G. R. Ganger, G. R. Goodson, M. K. Reiter, and J. J. Wylie (2005). Fault-scalable byzantine fault-tolerant services. In *Symposium on Operating Systems Principles (SOSP 2005)*, pp. 59–74.
- Abraham, I., G. Chockler, I. Keidar, and D. Malkhi (2006). Byzantine disk Paxos: Optimal resilience with Byzantine shared memory. *Distributed Computing* 18(5), 387–408.
- Aguilera, M., W. Chen, and S. Toueg (2000). Failure detection and consensus in the crash–recovery model. *Distributed Computing* 13(2), 99–125.
- Alpern, B. and F. B. Schneider (1985). Defining liveness. *Information Processing Letters* 21(4), 181–185.
- Amir, Y., C. Danilov, and J. Stanton (2000). A low latency, loss tolerant architecture and protocol for wide area group communication. In *Dependable Systems and Networks (DSN 2000, formerly FTCS-30 and DCCA-8)*, pp. 327–336.
- Amir, Y., D. Dolev, S. Kramer, and D. Malki (1992). Transis: A communication sub-system for high availability. In *Fault-Tolerant Computing (FTCS-22)*, pp. 76–84.
- Amir, Y., L. E. Moser, P. M. Melliar-Smith, D. A. Agarwal, and P. Ciarfella (1995). The Totem single-ring ordering and membership protocol. *ACM Transactions on Computer Systems* 13(4), 311–342.
- Attiya, H., A. Bar-Noy, and D. Dolev (1995). Sharing memory robustly in message-passing systems. *Journal of the ACM* 1(42), 124–142.
- Attiya, H. and J. Welch (2004). *Distributed Computing: Fundamentals, Simulations and Advanced Topics* (Second ed.). Wiley.
- Babaoglu, Ö., A. Bartoli, and G. Dini (1997). Enriched view synchrony: A programming paradigm for partitionable asynchronous distributed systems. *IEEE Transactions on Computers* 46(6), 642–658.
- Ban, B. (2002–2010). JGroups, a toolkit for reliable multicast communication. <http://www.jgroups.org>.
- Ben-Or, M. (1983). Another advantage of free choice: Completely asynchronous agreement protocols. In *Principles of Distributed Computing (PODC 1983)*, pp. 27–30.
- Ben-Or, M. and R. El-Yaniv (2003). Resilient-optimal interactive consistency in constant time. *Distributed Computing* 16, 249–262.
- Berman, P. and J. A. Garay (1989). Asymptotically optimal distributed consensus (extended abstract). In G. Ausiello, M. Dezani-Ciancaglini, and S. R. D. Rocca (Eds.), *Automata, Languages and Programming (ICALP 1989)*, Volume 372 of *Lecture Notes in Computer Science*, pp. 80–94.
- Bessani, A. and P. Sousa (2009–2010). SMaRt — High-performance Byzantine-fault-tolerant state machine replication. <http://code.google.com/p/bft-smart/>.
- Bhatti, N., M. Hiltunen, R. Schlichting, and W. Chiu (1998). Coyote: A system for constructing fine-grain configurable communication services. *ACM Transactions on Computer Systems* 16(4), 321–366.

- Birman, K. (1999). A review of experiences with reliable multicast. *Software – Practice and Experience* 29(9), 741–774.
- Birman, K. (2005). *Reliable Distributed Systems*. Springer.
- Birman, K., M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky (1999). Bimodal multicast. *ACM Transactions on Computer Systems* 17(2), 41–88.
- Birman, K. and T. Joseph (1987). Reliable communication in the presence of failures. *ACM Transactions on Computer Systems* 1(5), 47–76.
- Birman, K. and R. van Renesse (1993). *Reliable Distributed Programming with the Isis Toolkit*. IEEE Computer Society Press.
- Birman, K., R. van Renesse, and W. Vogels (2001). Spinglass: Secure and scalable communications tools for mission-critical computing. In *Survivability Conference and Exposition (DISCEX 2001)*.
- Boichat, R., P. Dutta, S. Frølund, and R. Guerraoui (2003a). Deconstructing Paxos. *SIGACT News* 34(1), 47–67.
- Boichat, R., P. Dutta, S. Frølund, and R. Guerraoui (2003b). Reconstructing Paxos. *SIGACT News* 34(2), 42–57.
- Boichat, R. and R. Guerraoui (2005). Reliable and total order broadcast in a crash–recovery model. *Journal of Parallel and Distributed Computing* 65(4), 397–413.
- Bracha, G. (1987). Asynchronous Byzantine agreement protocols. *Information and Computation* 75, 130–143.
- Bracha, G. and S. Toueg (1985). Asynchronous consensus and broadcast protocols. *Journal of the ACM* 32(4), 824–840.
- Brasileiro, F. V., F. Greve, A. Mostéfaoui, and M. Raynal (2001). Consensus in one communication step. In V. E. Malyskhin (Ed.), *Parallel Computing Technologies (PaCT 2001)*, Volume 2127 of *Lecture Notes in Computer Science*, pp. 42–50.
- Cachin, C. (2009). Yet another visit to Paxos. Research Report RZ 3754, IBM Research.
- Cachin, C., K. Kursawe, F. Petzold, and V. Shoup (2001). Secure and efficient asynchronous broadcast protocols (extended abstract). In J. Kilian (Ed.), *Advances in Cryptology – CRYPTO 2001*, Volume 2139 of *Lecture Notes in Computer Science*, pp. 524–541.
- Cachin, C., K. Kursawe, and V. Shoup (2005). Random oracles in Constantinople: Practical asynchronous Byzantine agreement using cryptography. *Journal of Cryptology* 18(3), 219–246.
- Cachin, C. and J. A. Poritz (2002). Secure intrusion-tolerant replication on the Internet. In *Dependable Systems and Networks (DSN 2002)*, pp. 167–176.
- Canetti, R. and T. Rabin (1993). Fast asynchronous Byzantine agreement with optimal resilience. In *Symposium on the Theory of Computing (STOC 1993)*, pp. 42–51.
- Castro, M. and B. Liskov (2002). Practical Byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems* 20(4), 398–461.
- Chandra, T., R. Griesemer, and J. Redstone (2007). Paxos made live – an engineering perspective. In *Principles of Distributed Computing (PODC 2007)*, pp. 398–407.
- Chandra, T., V. Hadzilacos, and S. Toueg (1996). The weakest failure detector for solving consensus. *Journal of the ACM* 43(4), 685–722.
- Chandra, T. and S. Toueg (1996). Unreliable failure detectors for reliable distributed systems. *Journal of the ACM* 43(2), 225–267.
- Chang, J. and N. Maxemchuck (1984). Reliable broadcast protocols. *ACM Transactions on Computer Systems* 2(3), 251–273.
- Charron-Bost, B., F. Pedone, and A. Schiper (Eds.) (2010). *Replication: Theory and Practice*, Volume 5959 of *Lecture Notes in Computer Science*. Springer.
- Cheriton, D. and W. Zwaenepoel (1985). Distributed process groups in the V kernel. *ACM Transactions on Computer Systems* 3(2), 77–107.
- Chockler, G., R. Guerraoui, and I. Keidar (2007). Amnesic distributed storage. In A. Pelc (Ed.), *Distributed Computing (DISC 2007)*, Volume 4731 of *Lecture Notes in Computer Science*, pp. 139–151.
- Chockler, G., I. Keidar, and R. Vitenberg (2001). Group communication specifications: A comprehensive study. *ACM Computing Surveys* 33(4), 427–469.
- Clement, A., M. Kapritsos, S. Lee, Y. Wang, L. Alvisi, M. Dahlin, and T. Riche (2009). UpRight cluster services. In *Symposium on Operating Systems Principles (SOSP 2009)*, pp. 277–290.

- Cooper, E. (1984a). Replicated procedure call. In *Principles of Distributed Computing (PODC 1984)*, pp. 220–232.
- Cooper, E. C. (1984b). Circus: A replicated procedure call facility. In *Reliability in Distributed Software and Database Systems (SRDS 1984)*, pp. 11–24.
- Coulouris, G., J. Dollimore, and T. Kindberg (2005). *Distributed Systems: Concepts and Design* (4th ed.). Addison-Wesley/Pearson Education.
- De Prisco, R., B. Lampson, and N. Lynch (2000). Revisiting the PAXOS algorithm. *Theoretical Computer Science* 243, 35–91.
- Delporte-Gallet, C., H. Fauconnier, and R. Guerraoui (2002). Failure detection lower bounds on registers and consensus. In D. Malkhi (Ed.), *Distributed Computing (DISC 2002)*, Volume 2508 of *Lecture Notes in Computer Science*, pp. 237–251.
- Delporte-Gallet, C., H. Fauconnier, and R. Guerraoui (2010). Tight failure detection bounds on atomic object implementations. *Journal of the ACM* 57(4).
- Delporte-Gallet, C., H. Fauconnier, R. Guerraoui, V. Hadzilacos, P. Kouznetsov, and S. Toueg (2004). The weakest failure detectors to solve certain fundamental problems in distributed computing. In *Principles of Distributed Computing (PODC 2004)*, pp. 338–346.
- Diffie, W. and M. E. Hellman (1976). New directions in cryptography. *IEEE Transactions on Information Theory* 22(6), 644–654.
- Dolev, D. and H. R. Strong (1983). Authenticated algorithms for Byzantine agreement. *SIAM Journal on Computing* 12(4), 656–666.
- Dolev, D. and A. C. Yao (1983). On the security of public key protocols. *IEEE Transactions on Information Theory* 29(2), 198–208.
- Doudou, A., B. Garbinato, and R. Guerraoui (2005). Tolerating arbitrary failures with state machine replication. In H. B. Diab and A. Y. Zomaya (Eds.), *Dependable Computing Systems: Paradigms, Performance Issues, and Applications*. Wiley.
- Dutta, P. and R. Guerraoui (2005). The inherent price of indulgence. *Distributed Computing* 18(1), 85–98.
- Dwork, C., N. Lynch, and L. Stockmeyer (1988). Consensus in the presence of partial synchrony. *Journal of the ACM* 35(2), 288–323.
- Eugster, P., R. Guerraoui, S. Handurukande, P. Kouznetsov, and A.-M. Kermarrec (2003). Lightweight probabilistic broadcast. *ACM Transactions on Computer Systems* 21(4), 341–374.
- Eugster, P., R. Guerraoui, and P. Kouznetsov (2004). Delta-reliable broadcast: A probabilistic measure of broadcast reliability. In *International Conference on Distributed Computing Systems (ICDCS 2004)*, pp. 636–643.
- Ezhilchelvan, P., A. Mostefaoui, and M. Raynal (2001). Randomized multivalued consensus. In *International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2001)*, pp. 195–200.
- Fekete, A., N. Lynch, and A. Shvartsman (2001). Specifying and using a partitionable group communication service. *ACM Transactions on Computer Systems* 19(2), 171–216.
- Felber, P. and R. Guerraoui (2000). Programming with object groups in CORBA. *IEEE Concurrency* 8(1), 48–58.
- Fidge, C. (1988). Timestamps in message-passing systems that preserve the partial ordering. In *11th Australian Computer Science Conference*.
- Fischer, M., N. Lynch, and M. Paterson (1985). Impossibility of distributed consensus with one faulty process. *Journal of the ACM* 32(2), 374–382.
- Friedman, R. and R. van Renesse (1996). Strong and weak virtual synchrony in Horus. In *Symposium on Reliable and Distributed Systems (SRDS 1996)*, pp. 140–149.
- Garbinato, B., R. Guerraoui, and K. Mazouni (1995). Implementation of the GARF replicated objects platform. *Distributed Systems Engineering* 2(1), 14–27.
- Garbinato, B., F. Pedone, and R. Schmidt (2004). An adaptive algorithm for efficient message diffusion in unreliable environments. In *Dependable Systems and Networks (DSN 2004)*, pp. 507–516.
- Gifford, D. K. (1979). Weighted voting for replicated data. In *Symposium on Operating Systems Principles (SOSP 1979)*, pp. 150–162.

- Golding, R. and D. Long (1992). Design choices for weak-consistency group communication. Technical Report UCSC-CRL-92-45, University of California Santa Cruz.
- Goldreich, O. (2001–2004). *Foundations of Cryptography*, Volume I & II. Cambridge University Press.
- Gray, C. and D. Cheriton (1989). Leases: An efficient fault-tolerant mechanism for distributed file cache consistency. In *Symposium on Operating Systems Principles (SOSP 1989)*, pp. 202–210.
- Gray, J. (1978). Notes on database operating systems. In *Operating Systems: An Advanced Course*, Volume 60 of *Lecture Notes in Computer Science*, pp. 393–481.
- Guerraoui, R. (2000). Indulgent algorithms. In *Principles of Distributed Computing (PODC 2000)*, pp. 289–297.
- Guerraoui, R. (2002). Non-blocking atomic commit in asynchronous distributed systems with failure detectors. *Distributed Computing* 15(1), 17–25.
- Guerraoui, R., P. Eugster, P. Felber, B. Garbinato, and K. Mazouni (2000). Experiences with object group systems. *Software – Practice and Experience* 30(12), 1375–1404.
- Guerraoui, R., N. Knežević, V. Quéma, and M. Vukolić (2010). The next 700 BFT protocols. In *European Conference on Computer Systems (EuroSys 2010)*, pp. 363–376.
- Guerraoui, R. and R. Levy (2004). Robust emulations of a shared memory in a crash–recovery model. In *International Conference on Distributed Computing Systems (ICDCS 2004)*, pp. 400–407.
- Guerraoui, R., R. Oliveria, and A. Schiper (1998). Stubborn communication channels. Technical Report LSR-REPORT-1998-009, Ecole Polytechnique Fédérale de Lausanne (EPFL).
- Guerraoui, R. and M. Raynal (2004). The information structure of indulgent consensus. *IEEE Transactions on Computers* 53(4), 453–466.
- Guerraoui, R. and A. Schiper (2001). Genuine atomic multicast in asynchronous distributed systems. *Theoretical Computer Science* 254, 297–316.
- Gupta, I., A.-M. Kermarrec, and A. Ganesh (2006). Efficient and adaptive epidemic-style protocols for reliable and scalable multicast. *IEEE Transactions on Parallel and Distributed Systems* 17(7), 593–605.
- Hadzilacos, V. (1984). *Issues of Fault Tolerance in Concurrent Computations*. Ph. D. thesis, Harvard University.
- Hadzilacos, V. and S. Toueg (1993). Fault-tolerant broadcasts and related problems. In S. J. Mullender (Ed.), *Distributed Systems*. New York: ACM Press & Addison-Wesley. Expanded version appears as Technical Report TR94-1425, Department of Computer Science, Cornell University, 1994.
- Hayden, M. (1998). *The Ensemble System*. Ph. D. thesis, Cornell University, Computer Science Department.
- Herlihy, M. and J. Wing (1990). Linearizability: A correctness condition for concurrent objects. *ACM Transactions on Programming Languages and Systems* 3(12), 463–492.
- Israeli, A. and M. Li (1993). Bounded timestamps. *Distributed Computing* 4(6), 205–209.
- Jelasiy, M., R. Guerraoui, A.-M. Kermarrec, and M. van Steen (2004). The peer sampling service: Experimental evaluation of unstructured gossip-based implementations. In H.-A. Jacobsen (Ed.), *Middleware 2004*, Volume 3231 of *Lecture Notes in Computer Science*, pp. 79–98.
- Kaashoek, F. and A. Tanenbaum (1991). Group communication in the Amoeba distributed operating system. In *International Conference on Distributed Computing Systems (ICDCS 1991)*, pp. 222–230.
- Kaashoek, F., A. Tanenbaum, S. Hummel, and H. Bal (1989). An efficient reliable broadcast protocol. *Operating Systems Review* 4(23), 5–19.
- Kermarrec, A.-M., L. Massoulié, and A. J. Ganesh (2003). Probabilistic reliable dissemination in large-scale systems. *IEEE Transactions on Parallel and Distributed Systems* 14(3), 248–258.
- Koldehofe, B. (2003). Buffer management in probabilistic peer-to-peer communication protocols. In *Symposium on Reliable Distributed Systems (SRDS 2003)*, pp. 76–85.
- Kotla, R., L. Alvisi, M. Dahlin, A. Clement, and E. L. Wong (2009). Zyzzyva: Speculative Byzantine fault tolerance. *ACM Transactions on Computer Systems* 27(4), 7:1–7:39.

- Kouznetsov, P., R. Guerraoui, S. Handurukande, and A.-M. Kermarrec (2001). Reducing noise in gossip-based reliable broadcast. In *Symposium on Reliable Distributed Systems (SRDS 2001)*, pp. 186–189.
- Kshemkalyani, A. D. and M. Singhal (2008). *Distributed Computing: Principles, Algorithms, and Systems*. Cambridge University Press.
- Ladin, R., B. Liskov, and L. Shrira (1990). Lazy replication: Exploiting the semantics of distributed services. In *Principles of Distributed Computing (PODC 1990)*, pp. 43–57.
- Lamport, L. (1977). Concurrent reading and writing. *Communications of the ACM* 11(20), 806–811.
- Lamport, L. (1978). Time, clocks and the ordering of events in a distributed system. *Communications of the ACM* 21(7), 558–565.
- Lamport, L. (1986a). On interprocess communication. Part I: Basic formalism. *Distributed Computing* 2(1), 75–85.
- Lamport, L. (1986b). On interprocess communication. Part II: Algorithms. *Distributed Computing* 2(1), 86–101.
- Lamport, L. (1998). The part-time parliament. *ACM Transactions on Computer Systems* 16(2), 133–169. Initially appeared as Technical Report 49, DEC Systems Research Center, 1989.
- Lamport, L., R. Shostak, and M. Pease (1982). The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems* 4(3), 382–401.
- Lampson, B. (2001). The ABCD’s of Paxos. In *Principles of Distributed Computing (PODC 2001)*.
- Lesley, N. and A. Fekete (2003). Providing view synchrony for group communication services. *Acta Informatica* 40(3), 159–210.
- Li, H. C., A. Clement, A. S. Aiyer, and L. Alvisi (2007). The Paxos register. In *Symposium on Reliable Distributed Systems (SRDS 2007)*, pp. 114–126.
- Lin, M.-J. and K. Marzullo (1999). Directional gossip: Gossip in a wide area network. In J. Hlavicka, E. Maehle, and A. Pataricza (Eds.), *European Dependable Computing Conference (EDCC-3)*, Volume 1667 of *Lecture Notes in Computer Science*, pp. 364–379.
- Liskov, B. (2010). From viewstamped replication to Byzantine fault tolerance. In B. Charron-Bost, F. Pedone, and A. Schiper (Eds.), *Replication: Theory and Practice*, Volume 5959 of *Lecture Notes in Computer Science*, pp. 121–149. Springer.
- Lynch, N. (1996). *Distributed Algorithms*. Morgan Kaufmann.
- Lynch, N. and A. Shvartsman (1997). Robust emulation of shared memory using dynamic quorum acknowledged broadcasts. In *Fault-Tolerant Computing Systems (FTCS 1997)*, pp. 272–281.
- Lynch, N. and A. Shvartsman (2002). RAMBO: A reconfigurable atomic memory service for dynamic networks. In D. Malkhi (Ed.), *Distributed Computing (DISC 2002)*, Volume 2508 of *Lecture Notes in Computer Science*, pp. 173–190.
- Lynch, N. A. (1989). A hundred impossibility proofs for distributed computing. In *Principles of Distributed Computing (PODC 1989)*, pp. 1–28.
- Malkhi, D. and M. K. Reiter (1998). Byzantine quorum systems. *Distributed Computing* 11(4), 203–213.
- Martin, J.-P. and L. Alvisi (2006). Fast Byzantine consensus. *IEEE Transactions on Dependable and Secure Computing* 3(3), 202–215.
- Martin, J.-P., L. Alvisi, and M. Dahlin (2002). Minimal Byzantine storage. In D. Malkhi (Ed.), *Distributed Computing (DISC 2002)*, Volume 2508 of *Lecture Notes in Computer Science*, pp. 311–325.
- Menezes, A. J., P. C. van Oorschot, and S. A. Vanstone (1997). *Handbook of Applied Cryptography*. CRC Press.
- Milosevic, Z., M. Hutle, and A. Schiper (2009). Unifying Byzantine consensus algorithms with weak interactive consistency. In T. F. Abdelzaher, M. Raynal, and N. Santoro (Eds.), *Principles of Distributed Systems (OPDIS 2009)*, Volume 5923 of *Lecture Notes in Computer Science*, pp. 300–314.
- Miranda, H., A. Pinto, and L. Rodrigues (2001). Appia, a flexible protocol kernel supporting multiple coordinated channels. In *International Conference on Distributed Computing Systems (ICDCS 2001)*, pp. 707–710.

- Miranda, H., A. Pinto, L. Rodrigues, N. Carvalho, J. Mocito, and L. Rosa (2001–2009). Appia communication framework. <http://appia.di.fc.ul.pt/> and <http://sourceforge.net/projects/appia/>.
- Mishra, S., L. Peterson, and R. Schlichting (1993). Experience with modularity in Consul. *Software – Practice and Experience* 23(10), 1059–1075.
- Moser, L. E., P. M. Melliar-Smith, and D. A. Agarwal (1995). The Totem system. In *Fault-Tolerant Computing Systems (FTCS 1995)*, pp. 61–66.
- Naor, M. and A. Wool (1998). The load, capacity and availability of quorum systems. *SIAM Journal on Computing* 27(2), 423–447.
- Neiger, G. and S. Toueg (1993). Simulating synchronized clocks and common knowledge in distributed systems. *Journal of the ACM* 2(40), 334–367.
- Oki, B. M. and B. Liskov (1988). Viewstamped replication: A new primary copy method to support highly-available distributed systems. In *Principles of Distributed Computing (PODC 1988)*, pp. 8–17.
- Parrington, G., S. Shrivastava, S. Wheeler, and M. Little (1995). The design and implementation of Arjuna. *Computing Systems* 8(3), 255–308.
- Pease, M., R. Shostak, and L. Lamport (1980). Reaching agreement in the presence of faults. *Journal of the ACM* 27(2), 228–234.
- Pereira, J., L. Rodrigues, and R. Oliveira (2003). Semantically reliable multicast: Definition, implementation, and performance evaluation. *IEEE Transactions on Computers* 52(2), 150–165.
- Peterson, G. (1983). Concurrent reading while writing. *ACM Transactions on Programming Languages and Systems* 5(1), 46–55.
- Peterson, L., N. Bucholz, and R. Schlichting (1989). Preserving and using context information in interprocess communication. *ACM Transactions on Computer Systems* 7(3), 217–246.
- Powell, D. (Ed.) (1991). *Delta Four: A Generic Architecture for Dependable Distributed Computing*. Springer.
- Powell, D. (1994). Distributed fault tolerance: Lessons from Delta-4. *IEEE Micro* 14(1), 36–47.
- Rabin, M. O. (1983). Randomized Byzantine generals. In *Foundations of Computer Science (FOCS 1983)*, pp. 403–409.
- Raynal, M., A. Schiper, and S. Toueg (1991). The causal ordering abstraction and a simple way to implement it. *Information Processing Letters* 39(6), 343–350.
- Reiter, M. K. (1994). Secure agreement protocols: Reliable and atomic group multicast in Rampart. In *Computer and Communications Security (CCS 1994)*, pp. 68–80.
- Rivest, R. L., A. Shamir, and L. Adleman (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM* 21(2), 120–126.
- Rodrigues, L., H. Fonseca, and P. Veríssimo (1996). Totally ordered multicast in large-scale systems. In *International Conference on Distributed Computing Systems (ICDCS 1996)*, pp. 503–510.
- Rodrigues, L., R. Guerraoui, and A. Schiper (1998). Scalable atomic multicast. In *International Conference on Computer Communications and Networks (ICCCN 1998)*, pp. 840–847.
- Rodrigues, L., S. Handurukande, J. Pereira, R. Guerraoui, and A.-M. Kermerrec (2003). Adaptive gossip-based broadcast. In *Dependable Systems and Networks (DSN 2003)*, pp. 47–56.
- Rodrigues, L. and M. Raynal (2003). Atomic broadcast in asynchronous crash–recovery distributed systems and its use in quorum-based replication. *IEEE Transactions on Knowledge and Data Engineering* 15(5), 1206–1217.
- Rodrigues, L. and P. Veríssimo (1992). xAMP: A multi-primitive group communications service. In *Symposium on Reliable Distributed Systems (SRDS 1992)*, pp. 112–121.
- Rufino, J., P. Veríssimo, G. Arroz, C. Almeida, and L. Rodrigues (1998). Fault-tolerant broadcasts in CAN. In *Fault-Tolerant Computing (FTCS 1998)*, pp. 150–159.
- Saltzer, J. H., D. P. Reed, and D. D. Clark (1984). End-to-end arguments in system design. *ACM Transactions on Computer Systems* 2(4), 277–288.
- Schneider, F., D. Gries, and R. Schlichting (1984). Fault-tolerant broadcasts. *Science of Computer Programming* 4(1), 1–15.
- Schneider, F. B. (1990). Implementing fault-tolerant services using the state machine approach: A tutorial. *ACM Computing Surveys* 22(4), 299–319.

- Schwarz, R. and F. Mattern (1994). Detecting causal relationships in distributed computations: In search of the holy grail. *Distributed Computing* 7(3), 149–174.
- Shamir, A. (1979). How to share a secret. *Communications of the ACM* 22(11), 612–613.
- Shao, C., E. Pierce, and J. Welch (2003). Multi-writer consistency conditions for shared memory objects. In F. E. Fich (Ed.), *Distributed Computing (DISC 2003)*, Volume 2848 of *Lecture Notes in Computer Science*, pp. 106–120.
- Skene, D. (1981). A decentralized termination protocol. In *IEEE Symposium on Reliability in Distributed Software and Database Systems*.
- Song, Y. J. and R. van Renesse (2008). Bosco: One-step Byzantine asynchronous consensus amnesic distributed storage. In G. Taubenfeld (Ed.), *Distributed Computing (DISC 2008)*, Volume 5218 of *Lecture Notes in Computer Science*, pp. 438–450.
- Spread Concepts LLC (2001–2010). The Spread toolkit. <http://www.spread.org>.
- Srikanth, T. K. and S. Toueg (1987). Simulating authenticated broadcasts to derive simple fault-tolerant algorithms. *Distributed Computing* 2, 80–94.
- Tanenbaum, A. and M. v. Steen (2002). *Distributed Systems: Principles and Paradigms*. Prentice Hall, Englewood Cliffs, NJ, USA.
- Tel, G. (2000). *Introduction to Distributed Algorithms* (2nd ed.). Cambridge University Press, Cambridge.
- Thomas, R. H. (1979). A majority consensus approach to concurrency control for multiple copy databases. *ACM Transactions on Database Systems* 4(2), 180–209.
- Toueg, S. (1984). Randomized Byzantine agreements. In *Principles of Distributed Computing (PODC 1984)*, pp. 163–178.
- van Renesse, R., K. Birman, and S. Maffei (1996). Horus: A flexible group communication system. *Communications of the ACM* 4(39), 76–83.
- Veríssimo, P. and L. Rodrigues (2001). *Distributed Systems for System Architects*. Kluwer Academic Publishers, Dordrecht (Hingham, MA).
- Veríssimo, P., L. Rodrigues, and M. Baptista (1989). AMP: A highly parallel atomic multicast protocol. In *Communications Architectures & Protocols (SIGCOMM '89)*, pp. 83–93.
- Vidyasankar, K. (1988). Converting Lamport's regular register to atomic register. *Information Processing Letters* 28(6), 287–290.
- Vidyasankar, K. (1990). Concurrent reading while writing revisited. *Distributed Computing* 2(4), 81–85.
- Vitányi, P. M. B. and B. Awerbuch (1986). Atomic shared register access by asynchronous hardware. In *Foundations of Computer Science (FOCS 1986)*, pp. 233–243.
- Voulgaris, S., M. Jelasity, and M. van Steen (2003). A robust and scalable peer-to-peer gossiping protocol. In G. Moro, C. Sartori, and M. Singh (Eds.), *Agents and Peer-to-Peer Computing (AP2PC 2003)*, Volume 2872 of *Lecture Notes in Computer Science*.
- Wensley, J., L. Lamport, J. Goldberg, M. Green, K. Levitt, P. Melliar-Smith, R. Shostak, and C. Weinstock (1978). SIFT: Design and analysis of a fault-tolerant computer for aircraft control. *Proceedings of the IEEE* 10(66), 1240–1255.
- Xiao, Z., K. Birman, and R. van Renesse (2002). Optimizing buffer management for reliable multicast. In *Dependable Systems and Networks (DSN 2002)*, pp. 155–166.

List of Modules

1.1	Job handler	13
1.2	Job transformation and processing abstraction	15
2.1	Fair-loss point-to-point links	34
2.2	Stubborn point-to-point links	35
2.3	Perfect point-to-point links	37
2.4	Logged perfect point-to-point links	40
2.5	Authenticated perfect point-to-point links	42
2.6	Perfect failure detector	50
2.7	Leader election	52
2.8	Eventually perfect failure detector	54
2.9	Eventual leader detector	56
2.10	Byzantine eventual leader detector	61
2.11	FIFO-order perfect point-to-point links	68
3.1	Best-effort broadcast	75
3.2	(Regular) Reliable broadcast	77
3.3	Uniform reliable broadcast	82
3.4	Stubborn best-effort broadcast	86
3.5	Logged best-effort broadcast	88
3.6	Logged uniform reliable broadcast	90
3.7	Probabilistic broadcast	94
3.8	FIFO-order (reliable) broadcast	102
3.9	Causal-order (reliable) broadcast	104
3.10	Causal-order uniform (reliable) broadcast	104
3.11	Byzantine consistent broadcast	112
3.12	Byzantine reliable broadcast	117
3.13	Byzantine consistent channel	121
3.14	Byzantine reliable channel	121
3.15	Logged reliable broadcast	129
4.1	$(1, N)$ regular register	143
4.2	$(1, N)$ atomic register	149
4.3	(N, N) atomic register	161
4.4	$(1, N)$ logged regular register	171
4.5	$(1, N)$ Byzantine safe register	176
4.6	$(1, N)$ Byzantine regular register	180
4.7	$(1, N)$ Byzantine atomic register	189

5.1	(Regular) Consensus	205
5.2	Uniform consensus	211
5.3	Epoch-change	218
5.4	Epoch consensus	221
5.5	Logged uniform consensus	229
5.6	Logged epoch-change	230
5.7	Logged epoch consensus	232
5.8	Randomized consensus	236
5.9	Common coin	237
5.10	Weak Byzantine consensus	245
5.11	(Strong) Byzantine consensus	246
5.12	Byzantine epoch-change	247
5.13	Byzantine epoch consensus	249
5.14	Conditional collect	250
5.15	Byzantine randomized consensus	262
5.16	Validated Byzantine consensus	273
6.1	Regular total-order broadcast	283
6.2	Uniform total-order broadcast	284
6.3	Byzantine total-order broadcast	289
6.4	Uniform terminating reliable broadcast	294
6.5	Uniform fast consensus	297
6.6	Fast Byzantine consensus	301
6.7	Non-blocking atomic commit	305
6.8	Group membership	308
6.9	View-synchronous communication	313
6.10	Uniform view-synchronous communication	314
6.11	Logged uniform total-order broadcast	326
6.12	Replicated state machine	330

List of Algorithms

1.1	Synchronous Job Handler	14
1.2	Asynchronous Job Handler	14
1.3	Job-Transformation by Buffering	16
2.1	Retransmit Forever	36
2.2	Eliminate Duplicates	38
2.3	Log Delivered	41
2.4	Authenticate and Filter	42
2.5	Exclude on Timeout	51
2.6	Monarchical Leader Election	53
2.7	Increasing Timeout	55
2.8	Monarchical Eventual Leader Detection	57
2.9	Elect Lower Epoch	58
2.10	Rotating Byzantine Leader Detection	62
2.11	Sequence Number	69
3.1	Basic Broadcast	76
3.2	Lazy Reliable Broadcast	78
3.3	Eager Reliable Broadcast	80
3.4	All-Ack Uniform Reliable Broadcast	83
3.5	Majority-Ack Uniform Reliable Broadcast	85
3.6	Basic Stubborn Broadcast	87
3.7	Logged Basic Broadcast	89
3.8	Logged Majority-Ack Uniform Reliable Broadcast	91
3.9	Eager Probabilistic Broadcast	95
3.10	Lazy Probabilistic Broadcast (part 1, data dissemination)	98
3.11	Lazy Probabilistic Broadcast (part 2, recovery)	99
3.12	Broadcast with Sequence Number	102
3.13	No-Waiting Causal Broadcast	105
3.14	Garbage-Collection of Causal Past (extends Algorithm 3.13)	107
3.15	Waiting Causal Broadcast	108
3.16	Authenticated Echo Broadcast	113
3.17	Signed Echo Broadcast	115
3.18	Authenticated Double-Echo Broadcast	118
3.19	Byzantine Consistent Channel	122
3.20	Byzantine Reliable Channel	123
3.21	Simple Optimization of Lazy Reliable Broadcast	126

3.22	Ideal Uniform Reliable Broadcast	128
3.23	Logged Eager Reliable Broadcast	130
3.24	No-Waiting Causal Broadcast using FIFO Broadcast	132
4.1	Read-One Write-All	144
4.2	Majority Voting Regular Register	147
4.3	From $(1, N)$ Regular to $(1, 1)$ Atomic Registers	152
4.4	From $(1, 1)$ Atomic to $(1, N)$ Atomic Registers	154
4.5	Read-Impose Write-All	157
4.6	Read-Impose Write-Majority (part 1, read)	158
4.7	Read-Impose Write-Majority (part 2, write and write-back)	159
4.8	From $(1, N)$ Atomic to (N, N) Atomic Registers	163
4.9	Read-Impose Write-Consult-All	166
4.10	Read-Impose Write-Consult-Majority (part 1, read and consult)	168
4.11	Read-Impose Write-Consult-Majority (part 2, write and write-back)	169
4.12	Logged Majority Voting (part 1, write)	174
4.13	Logged Majority Voting (part 2, read)	175
4.14	Byzantine Masking Quorum	178
4.15	Authenticated-Data Byzantine Quorum	181
4.16	Double-Write Byzantine Quorum (part 1, write)	184
4.17	Double-Write Byzantine Quorum (part 2, read)	185
4.18	Byzantine Quorum with Listeners (part 1, write)	190
4.19	Byzantine Quorum with Listeners (part 2, read)	191
4.20	Modification of Majority Voting	196
5.1	Flooding Consensus	206
5.2	Hierarchical Consensus	209
5.3	Flooding Uniform Consensus	213
5.4	Hierarchical Uniform Consensus	214
5.5	Leader-Based Epoch-Change	219
5.6	Read/Write Epoch Consensus	223
5.7	Leader-Driven Consensus	225
5.8	Logged Leader-Based Epoch-Change	231
5.9	Logged Read/Write Epoch Consensus	233
5.10	Logged Leader-Driven Consensus (part 1)	234
5.11	Logged Leader-Driven Consensus (part 2)	235
5.12	Randomized Binary Consensus (phase 1)	239
5.13	Randomized Binary Consensus (phase 2)	240
5.14	Randomized Consensus with Large Domain	243
5.15	Byzantine Leader-Based Epoch-Change	248
5.16	Signed Conditional Collect	251
5.17	Byzantine Read/Write Epoch Consensus (part 1, read phase)	252
5.18	Byzantine Read/Write Epoch Consensus (part 2, write phase)	253
5.19	Byzantine Leader-Driven Consensus	259
5.20	Byzantine Randomized Binary Consensus (phase 1)	263
5.21	Byzantine Randomized Binary Consensus (phase 2)	264
5.22	Rotating Coordinator (part 1)	270

- 5.23 Rotating Coordinator (part 2) 271
- 5.24 From Anchored Validity to Strong Validity 274
- 5.25 Echo Conditional Collect 275
- 6.1 Consensus-Based Total-Order Broadcast 285
- 6.2 Rotating Sender Byzantine Broadcast 290
- 6.3 Consensus-Based Uniform Terminating Reliable Broadcast 295
- 6.4 From Uniform Consensus to Uniform Fast Consensus 298
- 6.5 From Byzantine Consensus to Fast Byzantine Consensus 302
- 6.6 Consensus-Based Nonblocking Atomic Commit 306
- 6.7 Consensus-Based Group Membership 309
- 6.8 TRB-Based View-Synchronous Communication (part 1) 315
- 6.9 TRB-Based View-Synchronous Communication (part 2) 316
- 6.10 Consensus-Based Uniform View-Synchronous Comm. (part 1) 320
- 6.11 Consensus-Based Uniform View-Synchronous Comm. (part 2) 321
- 6.12 Logged Uniform Total-Order Broadcast 327
- 6.13 Replicated State Machine using Total-Order Broadcast 331
- 6.14 Direct Consensus-Based View-Synchronous Comm. (part 1) 335
- 6.15 Direct Consensus-Based View-Synchronous Comm. (part 2) 336

Index

- $O(\cdot)$, 66
- $\#(\cdot)$, 62
- \square , 245
- \triangle , 293
- append*, 105
- authenticate*, 30
- authentic*, 186
- binds*, 254
- byzhighestval*, 177
- certifiedvalue*, 254
- head*, 291
- highestval*, 148
- highest*, 155
- leader*, 61
- maxrank*, 53
- min*, 207
- quorumhighest*, 254
- random*, 95
- rank*, 20
- remove*, 105
- retrieve*, 26
- selectedmax*, 186
- select*, 59
- sign*, 31
- sort*, 285
- sound*, 253
- starttimer*, 35
- store*, 26
- unbound*, 254
- verifyauth*, 30
- verifysig*, 31
- access
 - sequential, 138
 - serial, 139
- accuracy (failure detector), 49, 54
- accuracy (leader election), 52, 56
- agreement (leader election), 56
- algorithm, 8
 - deterministic, 22
 - distributed, 16, 20
 - execution, 20
 - fail-arbitrary, 17, 112, 114, 117, 122, 123, 177, 180, 183, 189, 250, 252, 301
 - fail-noisy, 17, 216, 218, 225, 270
 - fail-noisy-arbitrary, 246, 259, 288
 - fail-recovery, 17, 86, 89, 90, 173, 230, 232, 234, 326
 - fail-silent, 17, 76, 80, 84, 104, 146, 158, 169, 222, 284, 298, 330
 - fail-stop, 17, 52, 78, 82, 144, 156, 165, 205, 208, 212, 213, 293, 304, 309, 314, 319, 335
 - randomized, 17, 22, 94, 98
 - randomized fail-arbitrary, 261
 - randomized fail-silent, 238, 242
- algorithm name
 - All-Ack Uniform Reliable Broadcast, 82
 - Authenticate and Filter, 42
 - Authenticated Double-Echo Broadcast, 117
 - Authenticated Echo Broadcast, 112
 - Authenticated-Data Byzantine Quorum, 180
 - Basic Broadcast, 76
 - Basic Stubborn Broadcast, 86
 - Broadcast with Sequence Number, 101
 - Byzantine Consistent Channel, 122
 - Byzantine Leader-Based Epoch-Change, 246
 - Byzantine Leader-Driven Consensus, 259
 - Byzantine Masking Quorum, 177
 - Byzantine Quorum with Listeners, 189
 - Byzantine Randomized Binary Consensus, 261

- Byzantine Read/Write Epoch Consensus, 251
 - Byzantine Reliable Channel, 123
 - Consensus-Based Group Membership, 309
 - Consensus-Based Nonblocking Atomic Commit, 304
 - Consensus-Based Total-Order Broadcast, 284
 - Consensus-Based Uniform Terminating Reliable Broadcast, 293
 - Consensus-Based Uniform View-Synchronous Communication, 319
 - Direct Consensus-Based View-Synchronous Communication, 335
 - Double-Write Byzantine Quorum, 182
 - Eager Probabilistic Broadcast, 94
 - Eager Reliable Broadcast, 79
 - Echo Conditional Collect, 274
 - Elect Lower Epoch, 57
 - Eliminate Duplicates, 37
 - Exclude on Timeout, 50
 - Flooding Consensus, 205
 - Flooding Uniform Consensus, 212
 - From $(1, 1)$ Atomic to $(1, N)$ Atomic Registers, 153
 - From $(1, N)$ Atomic to (N, N) Atomic Registers, 161
 - From $(1, N)$ Regular to $(1, 1)$ Atomic Registers, 151
 - From Anchored Validity to Strong Validity, 273
 - From Byzantine Consensus to Fast Byzantine Consensus, 300
 - From Uniform Consensus to Uniform Fast Consensus, 297
 - Garbage-Collection of Causal Past, 106
 - Hierarchical Consensus, 208
 - Hierarchical Uniform Consensus, 213
 - Increasing Timeout, 54
 - Lazy Probabilistic Broadcast, 97
 - Lazy Reliable Broadcast, 78
 - Leader-Based Epoch-Change, 218
 - Leader-Driven Consensus, 225
 - Log Delivered, 40
 - Logged Basic Broadcast, 89
 - Logged Eager Reliable Broadcast, 129
 - Logged Leader-Based Epoch-Change, 230
 - Logged Leader-Driven Consensus, 234
 - Logged Majority Voting, 172
 - Logged Majority-Ack Uniform Reliable Broadcast, 90
 - Logged Read/Write Epoch Consensus, 232
 - Logged Uniform Total-Order Broadcast, 326
 - Majority Voting Regular Register, 146
 - Majority-Ack Uniform Reliable Broadcast, 84
 - Monarchical Eventual Leader Detection, 57
 - Monarchical Leader Election, 52
 - No-Waiting Causal Broadcast, 104
 - Randomized Binary Consensus, 238
 - Randomized Consensus with Large Domain, 242
 - Read-Impose Write-All $(1, N)$ Atomic Register, 156
 - Read-Impose Write-Consult-All (N, N) Atomic Register, 165
 - Read-Impose Write-Consult-Majority (N, N) Atomic Register, 167
 - Read-Impose Write-Majority $(1, N)$ Atomic Register, 157
 - Read-One Write-All Regular Register, 144
 - Read/Write Epoch Consensus, 222
 - Replicated State Machine, 330
 - Retransmit Forever, 35
 - Rotating Byzantine Leader Detection, 61
 - Rotating Coordinator, 270
 - Rotating Sender Byzantine Broadcast, 288
 - Sequence Number, 68
 - Signed Conditional Collect, 250
 - Signed Echo Broadcast, 114
 - TRB-Based View-Synchronous Communication, 314
 - Waiting Causal Broadcast, 108
 - anchored validity (consensus), 267
- Big-O Notation, 66
- broadcast, 73
- atomic, 282
 - best-effort, 75
 - Byzantine consistent, 111
 - Byzantine reliable, 117

- Byzantine total-order, 288
- causal-order, 103
- FIFO-order, 101
- logged best-effort, 88
- logged uniform reliable, 90
- reliable, 77
- terminating reliable, 292
- total-order, 282
- uniform reliable, 81
- view-synchronous, 311
- Byzantine, 29
- Byzantine consistent channel, 120
- Byzantine consensus, 244
- Byzantine Generals, 338
- Byzantine leader detector, 60
- Byzantine reliable channel, 120

- causal order, 100
- causality, 45
- channel, 111
- common coin, 237
- communication step, 21
- completeness (failure detector), 49, 54
- completeness (operation), 141
- computation step, 21
- concurrent operations, 142
- conditional collect, 249
- consensus, 203
 - Byzantine, 244
 - Byzantine randomized, 261
 - epoch, 220
 - fast, 296
 - fast Byzantine, 300
 - logged uniform, 228
 - randomized, 236
 - regular, 204
 - strong Byzantine, 245
 - uniform, 211, 245
 - uniform fast, 297
 - validated Byzantine, 267
 - weak Byzantine, 244
- correct process, 21
- coverage, 47
- crash-recovery, 26

- digital signature, 31

- eavesdrop, 28
- epoch consensus, 220
 - Byzantine, 249
 - logged, 230
- epoch-change, 217
 - Byzantine, 246
 - logged, 229
- event, 9
 - indication, 11
 - request, 11
 - $\diamond\mathcal{P}$ -Restore, 54
 - $\diamond\mathcal{P}$ -Suspect, 54
 - Ω -Trust, 56
 - \mathcal{P} -Crash, 50
 - *al*-Deliver, 42
 - *al*-Send, 42
 - *bcb*-Broadcast, 112
 - *bcb*-Deliver, 112
 - *bcch*-Broadcast, 121
 - *bcch*-Deliver, 121
 - *bc*-Decide, 246
 - *bc*-Propose, 246
 - *beb*-Broadcast, 75
 - *beb*-Deliver, 75
 - *bec*-StartEpoch, 247
 - *bep*-Abort, 249
 - *bep*-Aborted, 249
 - *bep*-Decide, 249
 - *bep*-Propose, 249
 - *bld*-Complain, 61
 - *bld*-Trust, 61
 - *bonar*-Read, 189
 - *bonar*-ReadReturn, 189
 - *bonar*-Write, 189
 - *bonar*-WriteReturn, 189
 - *bonrr*-Read, 180
 - *bonrr*-ReadReturn, 180
 - *bonrr*-Write, 180
 - *bonrr*-WriteReturn, 180
 - *bonsr*-Read, 176
 - *bonsr*-ReadReturn, 176
 - *bonsr*-Write, 176
 - *bonsr*-WriteReturn, 176
 - *brb*-Broadcast, 117
 - *brb*-Deliver, 117
 - *brch*-Broadcast, 121
 - *brch*-Deliver, 121
 - *brc*-Decide, 262
 - *brc*-Propose, 262
 - *btob*-Broadcast, 289
 - *btob*-Deliver, 289
 - *cc*-Collected, 250

- *cc*-Input, 250
- *coin*-Output, 237
- *coin*-Release, 237
- *crb*-Broadcast, 104
- *crb*-Deliver, 104
- *curb*-Broadcast, 104
- *curb*-Deliver, 104
- *c*-Decide, 205
- *c*-Propose, 205
- *ec*-StartEpoch, 218
- *ep*-Abort, 221
- *ep*-Aborted, 221
- *ep*-Decide, 221
- *ep*-Propose, 221
- *fbc*-Decide, 301
- *fbc*-Propose, 301
- *fl*-Deliver, 34
- *fl*-Send, 34
- *fpl*-Deliver, 68
- *fpl*-Send, 68
- *frb*-Broadcast, 102
- *frb*-Deliver, 102
- *gm*-View, 308
- *jh*-Confirm, 13
- *jh*-Submit, 13
- *lbeb*-Broadcast, 88
- *lbeb*-Deliver, 88
- *lec*-StartEpoch, 230
- *lep*-Abort, 232
- *lep*-Aborted, 232
- *lep*-Decide, 232
- *lep*-Propose, 232
- *le*-Leader, 52
- *lonrr*-Read, 171
- *lonrr*-ReadReturn, 171
- *lonrr*-Write, 171
- *lonrr*-WriteReturn, 171
- *lpl*-Deliver, 40
- *lpl*-Send, 40
- *lrb*-Broadcast, 129
- *lrb*-Deliver, 129
- *luc*-Decide, 229
- *luc*-Propose, 229
- *lurb*-Broadcast, 90
- *lurb*-Deliver, 90
- *lutob*-Broadcast, 326
- *lutob*-Deliver, 326
- *nbac*-Decide, 305
- *nbac*-Propose, 305
- *nnar*-Read, 161
- *nnar*-ReadReturn, 161
- *nnar*-Write, 161
- *nnar*-WriteReturn, 161
- *onar*-Read, 149
- *onar*-ReadReturn, 149
- *onar*-Write, 149
- *onar*-WriteReturn, 149
- *onrr*-Read, 143
- *onrr*-ReadReturn, 143
- *onrr*-Write, 143
- *onrr*-WriteReturn, 143
- *pb*-Broadcast, 94
- *pb*-Deliver, 94
- *pl*-Deliver, 37
- *pl*-Send, 37
- *rb*-Broadcast, 77
- *rb*-Deliver, 77
- *rc*-Decide, 236
- *rc*-Propose, 236
- *rsm*-Execute, 330
- *rsm*-Output, 330
- *sbeb*-Broadcast, 86
- *sbeb*-Deliver, 86
- *sl*-Deliver, 35
- *sl*-Send, 35
- *th*-Confirm, 15
- *th*-Error, 15
- *th*-Submit, 15
- *tob*-Broadcast, 283
- *tob*-Deliver, 283
- *uc*-Decide, 211
- *uc*-Propose, 211
- *ufc*-Decide, 297
- *ufc*-Propose, 297
- *urb*-Broadcast, 82
- *urb*-Deliver, 82
- *utob*-Broadcast, 284
- *utob*-Deliver, 284
- *utrb*-Broadcast, 294
- *utrb*-Deliver, 294
- *uvs*-Block, 314
- *uvs*-BlockOk, 314
- *uvs*-Broadcast, 314
- *uvs*-Deliver, 314
- *uvs*-View, 314
- *vbc*-Decide, 273
- *vbc*-Propose, 273
- *vs*-Block, 313

- *vs-BlockOk*, 313
- *vs-Broadcast*, 313
- *vs-Deliver*, 313
- *vs-View*, 313
- *wbc-Decide*, 245
- *wbc-Propose*, 245
- *Init*, 13, 26
- *Recovery*, 26
- eventual leader detector, 56
- failure, 24
 - detection, 49
 - link, 33
 - process, 24
 - suspicion, 54
- failure detector, 48
- failure detector
 - eventually perfect, 53
 - perfect, 49
- fast consensus, 296
- fast Byzantine consensus, 300
- fault, 24
 - arbitrary, 29
 - crash, 24
 - crash-recovery, 26
 - eavesdropping, 28
 - omission, 26
- FIFO order, 10, 100
- finite-write termination, 185
- gossip, 95
- graceful degradation, 66
- group membership, 307
 - monotone, 308
- group view, 308
- hash function, 30
- heartbeat, 46, 50, 59
- indication, 11
- instance, 13
- job handler, 13
- layer, 9
- leader detector, 56
 - Byzantine, 60
- leader-election, 51
- lease, 46
- linearization, 160
- link, 32
 - authenticated, 41
 - fair-loss, 34
 - logged perfect, 39
 - perfect, 37
 - stubborn, 35
- liveness, 22
- log, 26
- log-decide, 229
- log-deliver, 39, 88
- logical clock, 44
- logical time, 44
- MAC, 30
- membership, 307
- memory, 137
- message
 - deliver, 12, 21
 - receive, 12, 21
 - send, 12, 21
- message-authentication code, 30
- model, 63
 - fail-arbitrary, 64
 - fail-noisy, 63
 - fail-noisy-arbitrary, 64
 - fail-recovery, 63
 - fail-silent, 63
 - fail-stop, 63
 - randomized, 64
- module, 13
 - $(1, N)$ -AtomicRegister, 149
 - $(1, N)$ -ByzantineAtomicRegister, 189
 - $(1, N)$ -ByzantineRegularRegister, 180
 - $(1, N)$ -ByzantineSafeRegister, 176
 - $(1, N)$ -LoggedRegularRegister, 171
 - $(1, N)$ -RegularRegister, 143
 - (N, N) AtomicRegister, 161
 - AuthPerfectPointToPointLinks, 42
 - BestEffortBroadcast, 75
 - ByzantineConsensus, 246
 - ByzantineConsistentBroadcast, 112
 - ByzantineConsistentBroadcastChannel, 121
 - ByzantineEpochChange, 247
 - ByzantineEpochConsensus, 249
 - ByzantineLeaderDetector, 61
 - ByzantineRandomizedConsensus, 262
 - ByzantineReliableBroadcast, 117
 - ByzantineReliableBroadcastChannel, 121

- ByzantineTotalOrderBroadcast, 289
 - CausalOrderReliableBroadcast, 104
 - CausalOrderUniformReliableBroadcast, 104
 - CommonCoin, 237
 - ConditionalCollect, 250
 - Consensus, 205
 - EpochChange, 218
 - EpochConsensus, 221
 - EventualLeaderDetector, 56
 - EventuallyPerfectFailureDetector, 54
 - FairLossPointToPointLinks, 34
 - Fast Byzantine Consensus, 301
 - FIFOPerfectPointToPointLinks, 68
 - FIFOReliableBroadcast, 102
 - GroupMembership, 308
 - JobHandler, 13
 - LeaderElection, 52
 - LoggedBestEffortBroadcast, 88
 - LoggedEpochChange, 230
 - LoggedEpochConsensus, 232
 - LoggedPerfectPointToPointLinks, 40
 - LoggedReliableBroadcast, 129
 - LoggedUniformConsensus, 229
 - LoggedUniformReliableBroadcast, 90
 - LoggedUniformTotalOrderBroadcast, 326
 - NonBlockingAtomicCommit, 305
 - PerfectFailureDetector, 50
 - PerfectPointToPointLinks, 37
 - ProbabilisticBroadcast, 94
 - RandomizedConsensus, 236
 - ReliableBroadcast, 77
 - ReplicatedStateMachine, 330
 - StubbornBestEffortBroadcast, 86
 - StubbornPointToPointLinks, 35
 - TotalOrderBroadcast, 283
 - TransformationHandler, 15
 - Uniform Fast Consensus, 297
 - UniformConsensus, 211
 - UniformReliableBroadcast, 82
 - UniformTerminatingReliableBroadcast, 294
 - UniformTotalOrderBroadcast, 284
 - UniformViewSynchronousCommunication, 314
 - ValidatedByzantineConsensus, 273
 - ViewSynchronousCommunication, 313
 - WeakByzantineConsensus, 245
- module identifier, 13
 - nonblocking atomic commit, 304
 - order
 - causal, 100
 - first-in first-out (FIFO), 67, 100
 - partial, 142
 - total, 142, 282
 - performance, 65
 - precedence, 141
 - process, 20
 - arbitrary-fault, 29
 - Byzantine, 29
 - crash-recovery, 26
 - crash-stop, 24
 - protocol, 16
 - publish-subscribe, 4
 - quorum, 65, 84, 146, 158, 168, 173, 222, 238, 271
 - Byzantine, 65, 112, 115, 119, 180, 185, 189, 253, 262
 - Byzantine masking, 177
 - randomized Byzantine consensus, 261
 - randomized consensus, 236
 - rank, 20
 - register, 137
 - atomic, 138, 149, 189
 - (1, N), 156
 - (1, 1), 151, 153
 - (1, N), 149, 151, 153, 158, 162
 - (N , N), 160, 162, 165
 - Byzantine atomic, 189
 - Byzantine regular, 179
 - Byzantine safe, 176
 - logged regular, 170
 - regular, 138, 142, 170, 179
 - (1, N), 142, 144, 146, 151, 170, 173
 - safe, 138, 176
 - request, 11
 - resilience, 25
 - retrieve, 26
 - safety, 22
 - self, 20
 - sequential operations, 142
 - SIFT, 17, 71, 134

- state machine, 282
- store, 26
- strong Byzantine consensus, 245
- system
 - asynchronous, 44
 - partially synchronous, 47
 - synchronous, 45
- terminating reliable broadcast, 292
- timeout, 35, 49
- total order, 282
- uniformity, 81
- validated Byzantine consensus, 267
- validity (consensus), 204, 244
- view, 308
- view-synchronous communication, 311
 - uniform, 312
- wait-free, 140
- weak Byzantine consensus, 244