

Distributed Algorithms

Content

- 1 A Model of A Distributed Algorithm
 - 1.1 Transition Systems
 - 1.2 Causal Order of Events
- 2 Group Communication Algorithms
 - 2.1 Broadcast Semantics
 - 2.2 Centralize Algorithm for Total Order
 - 2.3 Three-Phase Distributed Algorithm
 - 2.4 Propagation Trees for Multicast
 - 2.5 Classification of Application Level Multicast Algorithms
 - 2.6 Semantics of Fault-Tolerant Group Communication
- 3 Self Stabilization
 - 3.1 Legal Execution Fragments
 - 3.2 Handled Transient Faults
 - 3.3 Dijkstra Self-Stabilizing Mutual Exclusion
 - 3.4 [Dolev] Solution with 2 States
 - 3.5 SS Algorithms for Spanning-Tree Construction
 - 3.5.1 Dolev, Israeli and Moran
 - 3.5.2 Afek and Bremler
 - 3.6 Limitation of Self-Stabilization
- 4 Global Snapshots
 - 4.1 Consistent Global Snapshots
 - 4.2 Stable Property Detection
 - 4.3 Chandy and Lamport Algorithm

5 Distributed Consensus

5.1 Agreement in a Failure Free System

5.2 Consensus Algorithm for Crash Failures

5.3 Consensus Algorithm for Byzantine Failures

5.4 Phase-King Algorithm for Consensus

5.5 Consensus in Asynchronous Message-Passing Systems with Failures

5.5.1 K-Set Consensus

5.5.2 Approximate Consensus

6 Deadlock Detection

6.1 Wait-For Graph

6.2 Deadlock Handling Strategies

6.3 Issues in Deadlock Detection

6.4 Models of Deadlocks

6.5 Knapp's Classification of Distributed Deadlock Detection Algorithms

6.6 Mitchell and Merritt's Algorithm for the Single-Resource Model

6.7 Chandy-Misra-Haas Algorithm for the AND Model

6.8 Chandy-Misra-Haas Algorithm for the OR Model

7 Peer-to-Peer Computing

7.1 Data Indexing and Overlays

7.1.1 Distributed Hash Tables

7.1.2 Chord DHT

7.1.3 Simple Lookup

7.1.4 Scalable Lookup

7.1.5 Managing Churn

7.2 Fairness

- 7.2.1 Game Theory
- 7.2.2 Trust and Reputation Management
- 7.2.3 Aberer and Despotovic Algorithm
- 7.2.4 Gupta et Al. Solution

Bibliography

1. **G.R.Andrews**, *Foundations of Multithreaded, Parallel, and Distributed Programming*, Addison Wesley, Inc., 2000
2. **Gerard Tel**, *Introduction to Distributed Algorithms*, Cambridge University, 1994
3. **M. Ben Ari**, *Principles of concurrent and distributed programming*, Prentice Hall, N.Y. 1990
4. **H.Ball**, *Programming distributed systems*, Prentice Hall, NY 1990
5. **E. W. Dijkstra**, *Self stabilizing systems in spite of distributed control*, Communications of the ACM, 17(11), 1974, 643–644.
6. **S. Dolev**, *Self-Stabilization*, MIT Press, 2000.
7. **J. E. Burns, M. G. Gouda, and R. E. Miller**, *Stabilization and pseudo-stabilization*, Distributed Computing archive, 7(1), 1993.
8. **Lamport**, *Solved Problems, Unsolved Problems and NonProblems in Concurrency*, Proceedings of the Third Annual ACM Symposium on Principles of Distributed Computing (August, 1984) - transcript of the PODC 1983 talk
9. **S. Dolev, A. Israeli, and S. Moran**, *Self-stabilization of dynamic systems assuming only read/write atomicity*, Proceedings of the 9th Annual ACM Symposium on Principles of Distributed Computing, Quebec City, Canada, August 22–24, 1990, 103–117.
10. **Afek, Y. and Bremler, A.** 1997. *Self-stabilizing unidirectional network algorithms by power-supply*. In Proceedings of the Eighth Annual ACM/SIAM Symposium on Discrete Algorithms (New Orleans, Louisiana, United States, January 05 - 07, 1997). Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, Philadelphia, PA, 111-120.
11. Karger, **Lehman, Leighton, Levine, Lewin, Panigrahy** *Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web*
12. **Stoica, Morris, Karger, Kaashoek, Balakrishnan** *Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications*