A List of Topics

1. Introduction

• How to define a distributed system
• Advantages of a distributed system
• Examples of a distributed system

2. Understanding models

• Communication models (e.g., message passing, knowledge based, shared memory)
• Weak vs. strong models
• Network topologies
• Synchronous vs. Asynchronous models
• Complexity models (e.g., run-time, message, round)

3. Syntax and semantics

• Guarded actions and nondeterminism
• Atomic operations and fairness

4. Program correctness

• Correctness criteria: safety and liveness properties
• Correctness proofs: predicate logic, proof by contradiction and induction

5. Clock synchronization

• Causality
• Logic and vector clocks
• Types of synchronization
• Synchronization algorithms: Berkeley, Lamport and Melliar-Smith, Cristian, and Network Time Protocol

6. Mutual exclusion

• Basic requirements
• Centralized solutions
• Distributed solutions: Lamport, Ricart & Agrawala, Maekawa, Suzuki-Kasami, and Raymond

7. Distributed snapshot

• Consistent cut and snapshot
• Chandy-Lamport Algorithm
• Applications of snapshot

8. Global state collection

• All-to-all broadcast
• Termination detection: Dijkstra-Scholten algorithm
• Deadlock detection
9. Faults and fault-tolerance
   - Types of failure and tolerance
   - Failure detection
   - Tolerating crash failure
   - Tolerating omission failure: Stenning, sliding window, alternating bit
   - Distributed consensus: Byzantine general’s problem and its solution

10. Consistency and replication
    - Consistency model: data centric and client centric
    - Replica management

11. Coordination algorithms
    - Leader election: bully, Chang-Roberts, Franklin, randomized
    - Synchronizers: ABD and Awerbuch ($\alpha$, $\beta$ and $\gamma$ synchronizers)

12. Group communication
    - Multicast: atomic and IP multicasts
    - Ordered multicast: total, causal and reliable ordered multicasts.