Fault Tolerance in Distributed Systems

4.5 Fault Tolerance

- Fault tolerance is the ability of a system to keep working even if some parts fail.

Importance of Fault Tolerance

- Systems fail when they don’t perform as expected, which can range from minor issues, like a grocery store running out of stock, to critical failures, like errors in an air traffic control system.

- As systems play more important roles in safety-related situations, it becomes crucial to design them to prevent failures.

4.5.1 Component Faults

- Systems can fail due to faults in any part of a computer, such as:
- - Processors
- - Memory
- - Input/Output devices
- - Cables
- - Software

Causes of Faults

- Faults are malfunctions that happen due to:
- - Design or programming errors
- - Physical damage or aging
- - Environmental factors

- Faults don’t always cause immediate failures, but some do eventually lead to system breakdowns.

Types of Faults

- 1.Transient Faults: Occur temporarily and are resolved on retrying.
- Example: A temporary network glitch.

- 2. Intermittent Faults: Appear and disappear randomly, making diagnosis difficult.
- Example: Loose connection causing intermittent issues.

- 3. Permanent Faults: Remain until the component is fixed or replaced.
- - Example: Burnt-out chip or software bug.

Goal of Fault-Tolerant Design

- The goal is to ensure that the overall system keeps running even if individual components fail.
- This approach differs from simply making each component reliable while accepting system failure if one part fails.

4.5.2 System Failures

- In critical distributed systems, the system must survive even if some processors fail.

- Distributed systems often have many components, increasing the chance of a component failing.

Types of Processor Faults

- 1. Fail-Silent Faults:
- - The faulty processor stops working and does not respond further.

- 2. Byzantine Faults:
- The faulty processor continues working but gives incorrect responses.
- It may conspire with other faulty components to appear functional.

Challenges of Byzantine Faults

- Byzantine faults are challenging because faulty components may give incorrect responses that seem correct.

- This requires tracking and verifying responses that may look right on the surface.