

Types of Failure

Parallel & Distributed Computing
Prof. Maryam Saif

Failure vs Fault vs Error

- **Fault:** The cause of a problem (e.g., hardware defect, software bug)
- **Error:** Incorrect internal state caused by a fault
- **Failure:** When the system deviates from its expected behavio

Classification of Failure

- Distributed systems typically classify failures into:
 - **Omission failures**
 - **Timing failures**
 - **Arbitrary (Byzantine) failures**

Omission Failure

- An **omission failure** occurs when a system component **fails to send or receive a message** that it is supposed to handle.
- These failures are common in network-based systems.
- A **send omission failure** occurs when a process **fails to send a message** that it was expected to send.
- **Causes:**
 - Buffer overflow
 - Software bug
 - Network interface failure
- **Impact:**
 - Receiving process waits indefinitely
 - Possible deadlocks
- **Example:**
 - A client sends a request to a server, but the server crashes before sending the response.
- **Detection and Handling:**
 - Acknowledgements
 - Retransmission mechanisms
 - Timeouts

Omission Failure

- A **receive omission failure** occurs when a process **fails to receive a message** that was actually sent.
- **Causes:**
 - Network packet loss
 - Receiver buffer overflow
 - Receiver crash
 - Network congestion
- **Impact:**
 - Loss of synchronization
 - Inconsistent system state
 - Incorrect computations
- **Example:**
 - A server sends a response, but the client never receives it due to packet loss.
- **Detection and Handling:**
 - Message acknowledgements
 - Retransmission protocols
 - Reliable communication protocols (TCP)
 - Duplicate message handling

Timing Failure

- A **timing failure** occurs when a system responds **too early, too late, or outside specified time constraints**.
 - These failures are critical in **real-time distributed systems**.
 - **Causes:**
 - Network latency
 - Clock drift
 - Overloaded processors
 - Poor scheduling
 - **Impact:**
 - Missed deadlines
 - System instability
 - Violation of real-time constraints
-
- **Types of Timing Violations:**
 - Late response
 - Early response
 - Timeout expiration
 - **Example:**
 - In an air traffic control system, delayed radar data can lead to unsafe decisions.
 - **Detection and Handling:**
 - Timeouts
 - Clock synchronization (NTP, logical clocks)
 - Real-time scheduling algorithms
 - Deadline monitoring

Arbitrary Failure

- An **arbitrary failure**, also known as a **Byzantine failure**, occurs when a component behaves **inconsistently or maliciously**, producing **unpredictable or incorrect results**. This is the most severe and difficult type of failure.
- **Characteristics:**
 - Incorrect messages
 - Conflicting responses
 - Random behavior
 - Malicious attacks
- **Causes:**
 - Software bugs
 - Data corruption
 - Security breaches
 - Malicious nodes
- **Impact:**
 - Data inconsistency
 - Loss of trust
 - System-wide failures
- **Example:**
 - A compromised server sends different values to different replicas in a distributed database.
- **Detection and Handling:**
 - Byzantine Fault Tolerance (BFT) algorithms
 - Majority voting
 - Cryptographic techniques
 - Redundancy
 - Secure authentication

Comparison

Failure Type	Description	Predictability	Severity
Omission	Message not sent/received	Medium	Moderate
Timing	Response outside time bounds	Medium	High (real-time systems)
Arbitrary	Unpredictable behavior	Low	Very High

Relationship to Fault Tolerance

Failure Type	Fault Tolerance Technique
Omission	Retransmission, acknowledgements
Timing	Timeouts, synchronized clocks
Arbitrary	Byzantine fault tolerance