

Distributed Systems

COMP90015 2019 Semester 1
Tutorial 3

Fundamental Models

Q1. Briefly explain the purpose of the following fundamental models and explain two important considerations for each:

- Interaction Model.
- Failure Model.
- Security Model.

Interaction Model

Models the interaction between processes of a distributed system - e.g. interaction between clients and servers or peers.

- Performance of communication channels.
 - Latency
 - Bandwidth
 - Jitter
- Computer clocks and timing events
 - Why the timestamps between two processes can vary?
 - Initial time setting being different
 - Differences in clock drift rates

Failure Model

Classifies the failures of processes and communication channels in a distributed system

- Omission failures
 - Omission failures refers to cases where a process or a communication channel fails to perform what is expected to do
- Arbitrary failures
 - Refers to any type of failure that can occur in a system
- Timing failures
 - These failures occur when time limits set on process execution time, message delivery time and clock rate drift.
 - More relevant to synchronous systems

Security Model

Identifies the possible threats to processes and communication channels, as well as protecting encapsulated objects against unauthorized access.

- Encryption.
- Authentication.
- Secure Channel.

Security Model - How to Perform Encryption

- Secret Key.
- Public/Private key pair.
- Establishing a secure channel (SSL).

Asynchronous Protocol vs Synchronous Protocol

Q2. Explain the difference between a synchronous protocol and an asynchronous protocol.

Asynchronous Protocol vs Synchronous Protocol

Q2. Explain the difference between a synchronous protocol and an asynchronous protocol.

- Synchronous communication blocks on both send and receive operations.
 - When a send is issued the sending process is blocked until the receive is issued.
 - Whenever the receive is issued the process blocks until a message arrives.
- In Asynchronous communication the send is nonblocking.
 - The sending process returns as soon as the message is copied to a local buffer and the transmission of the message proceeds in parallel.
 - Receive operation can be blocking or non-blocking (non-blocking receives are not normally supported in today's systems).

Questions?