System Models of Distributed Systems - 3

Shahul Hamead H,
AP/CSE - SSN

Failure Model

- Omission Failure
- Arbitrary Failure
- Timing Failure

- Masking Failure
- Reliability of one-to-one communication

Omission Failure

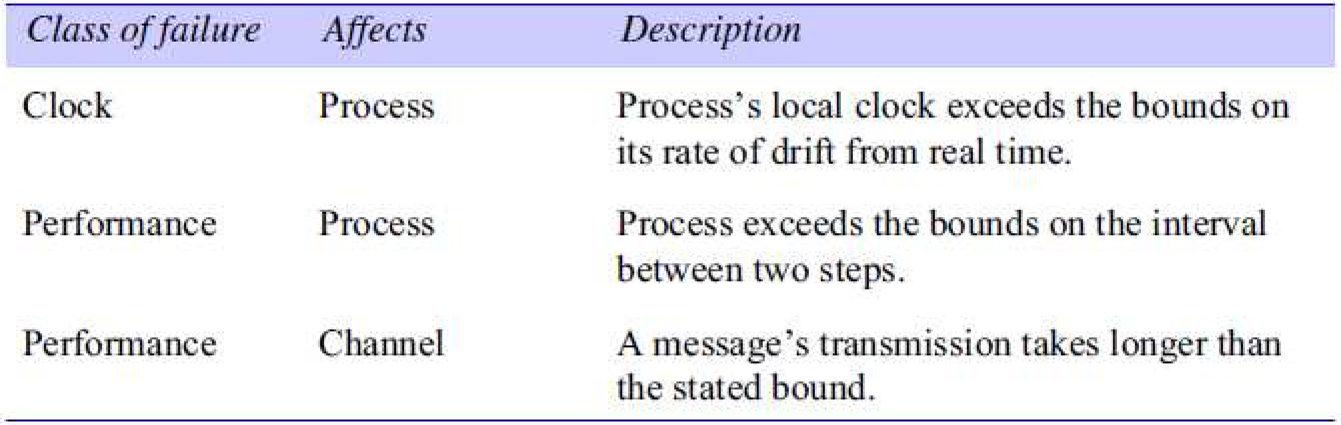
- Process failure – fail/stop, crash
- Communication Failure – Send omission/ Receive omission/ Channel omission

Omission and arbitrary failures

Class of failure	*Affects*	*Description*
Fail-stop	Process	Process halts and remains halted. Other processes may detect this state.
Crash	Process	Process halts and remains halted. Other processes may not be able to detect this state.
Omission	Channel	A message inserted in an outgoing message buffer never arrives at the other end's incoming message buffer.
Send-omission	Process	A process completes a *send* operation but the message is not put in its outgoing message buffer.
Receive-omission	Process	A message is put in a process's incoming message buffer, but that process does not receive it.
Arbitrary (Byzantine)	Process or channel	Process/channel exhibits arbitrary behaviour: it may send/transmit arbitrary messages at arbitrary times or commit omissions; a process may stop or take an incorrect step.

Timing Failure

- Synchronous Systems
- Process/ Communication



Class of failure	*Affects*	*Description*
Clock	Process	Process's local clock exceeds the bounds on its rate of drift from real time.
Performance	Process	Process exceeds the bounds on the interval between two steps.
Performance	Channel	A message's transmission takes longer than the stated bound.

Masking Failures

- “A service *masks a failure either by hiding* it altogether or by converting it into a more acceptable type of failure”
- Time out -> Retransmit

Reliability of One-to-One Communication

- Validity – “Any message in the outgoing message buffer is eventually delivered to the incoming message buffer”
- Integrity – “The message received is identical to one sent, and no messages are delivered twice”

Arbitrary Failures

- Byzantine Failure
- Non Tolerant ones

Security Model

- Protecting Objects
- Securing Process and their Interactions
- Threats to Process
- Threats to Channel
- Defeating Security Threats – Cryptography/ Authentication/ Secure Channels (Privacy, time stamps)

Other Threats

- Denial of Service
- Mobile Code

- Use of Threat Model