02_Modeling Distributed Systems

1. Preliminaries

1.1. Network, processors and processes

- For DAs, a DS is modeled as a set of **processors or processes** (equivalent)
- Processes do local computations, send messages, and receive messages
- Processes are connected by communication **channels** (**links**)
- Links are assumed to be **unidirectional**
- Network assumed to be **connected**

1.2. States and configurations

- **State of a process:** the set of values of all its variables
- **Subsets of states:** initial states and terminal states
- **State of a channel:** messages sent along the channel that have not yet been received
- **configuration** of a DS is a set of simultaneous states of all its processes and channels

1.3. Transitions, events, executions

Transition: state change of a DS

- Transitions caused by **events** in processes
- Types of **events:**

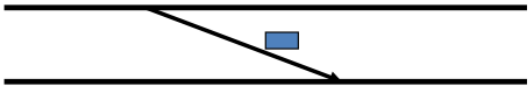
- nternal events
- message-send events
- message-receive events

An **execution** of a DA in an asynchronous system is an alternating sequence of configurations and events with all processes and channels in C1 in an initial state

1.4. Communication

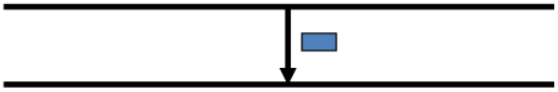
1.4.1. Asynchronous Communication (AC)

- **non-blocking** send, receive message is later than send message
- receive may be blocking or not (interrupt)



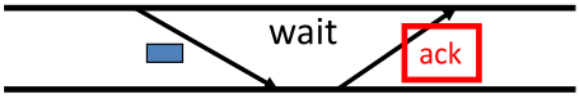
1.4.2. Synchronous Communication (SC)

- send and receive message simultaneously (both **blocking**)
- never messages in transit

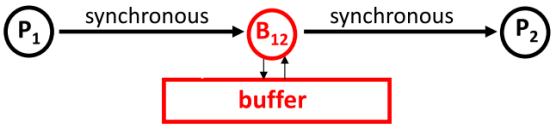


1.4.3. Simulated Relations

- A system with SC **can be simulated** by a system with AC
  - require **explicit acknowledgements** to force the sender only to continue when the message has been received



- A system with AC **can be simulated** by a system with SC
  - introduce for every link a process **with a buffer** that may delay messages



1.5. Properties of Links

1. Messages may (not) be lost
2. Messages may (not) be damaged

3. FIFO
4. Buffers are (un)bounded
5. Message delays are finite
6. Message delays are bounded

Properties 3.-6. **do not make sense** for synchronous communication

2. Asnchronous systems and synchronous systems

2.1. Asnchronous systems and algorithms

- In an **asynchronous system**:
  - message delays are **finite but arbitrary**
  - the relative processor speeds may be **unbounded**
  - there is **no common clock**
- An **asynchronous DA** consists of **pieces of code**, one for the receipt of each **message type**, and possibly for **internal events**
- So an asynchronous DA is **event-driven**
- **Pseudo-code:**

upon receipt of message do
actions

when condition do
actions

2.2. Synchronous systems and algorithms

- In a **synchronous system**:
  - message delays are **bounded**
  - the relative processor speeds are **bounded**
  - there is access to a **common clock**
- An **synchronous DA** consists of **rounds**
- So a synchronous DA is **time-driven**
- **Pseudo-code:**

do some number of times

round { **receive** all messages of the previous round
do local computations
send messages

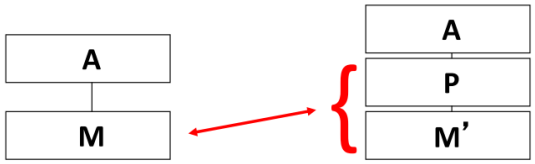
2.3. Simulation

Simulation: Make a system/model **behave differently** through a **software layer**

Design a DA for an "easy" system, and then run it on a "difficult" system

Example

- Suppose you have a DA A on model M
- Model M' is more complicated (e.g., may exhibit errors)
- Design a simulation (protocol) P for M' to let M' behave like M



- Run A on top of P

Local Simulation

- to every process(or) in the system, (M',P) looks like M
- to an outside observer there may be a difference

e.g. synchronizers, which make an asynchronous system look like a synchronous system

Global simulation

- also to an outside observer there is no difference