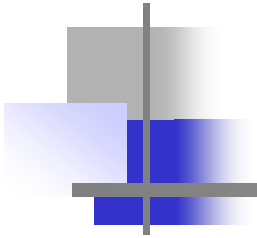
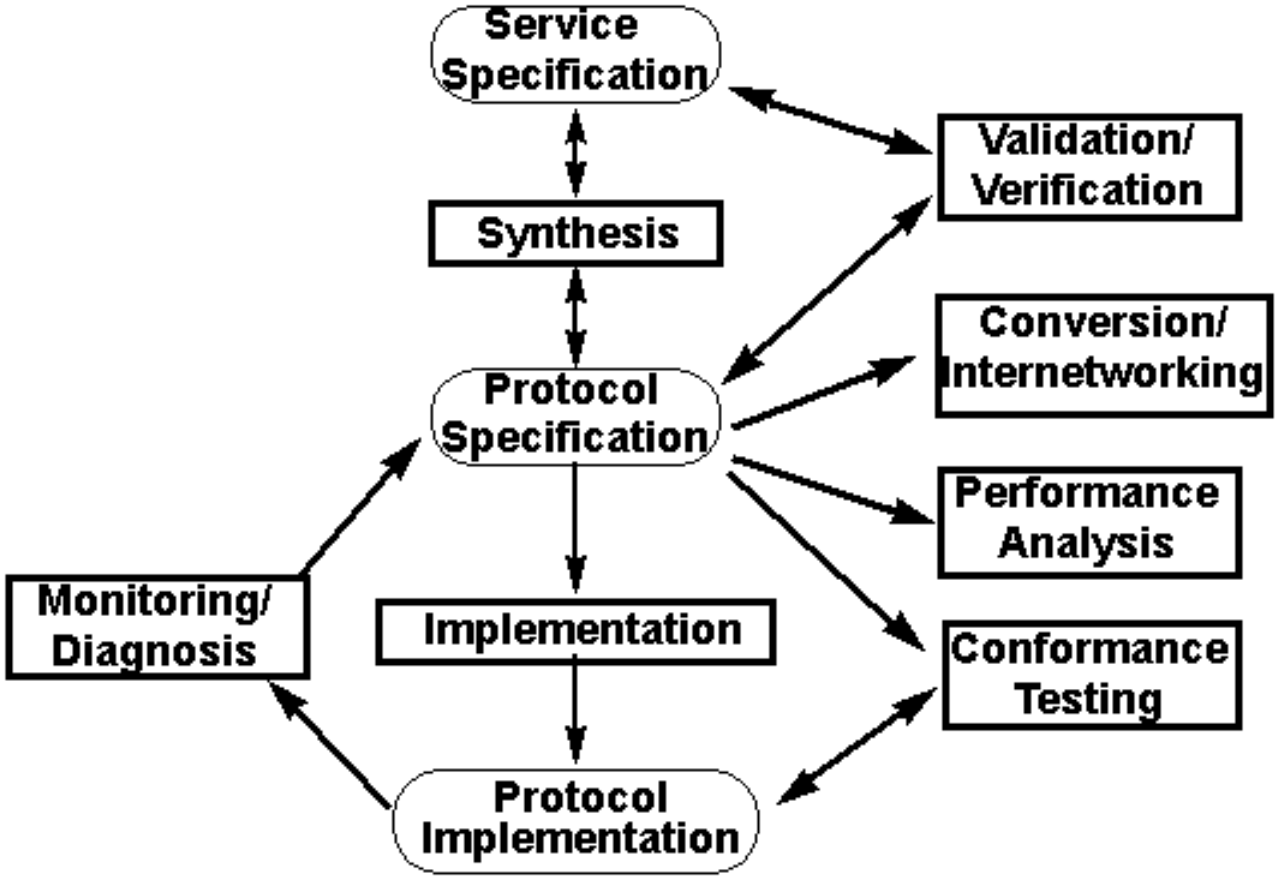


Protocol Engineering

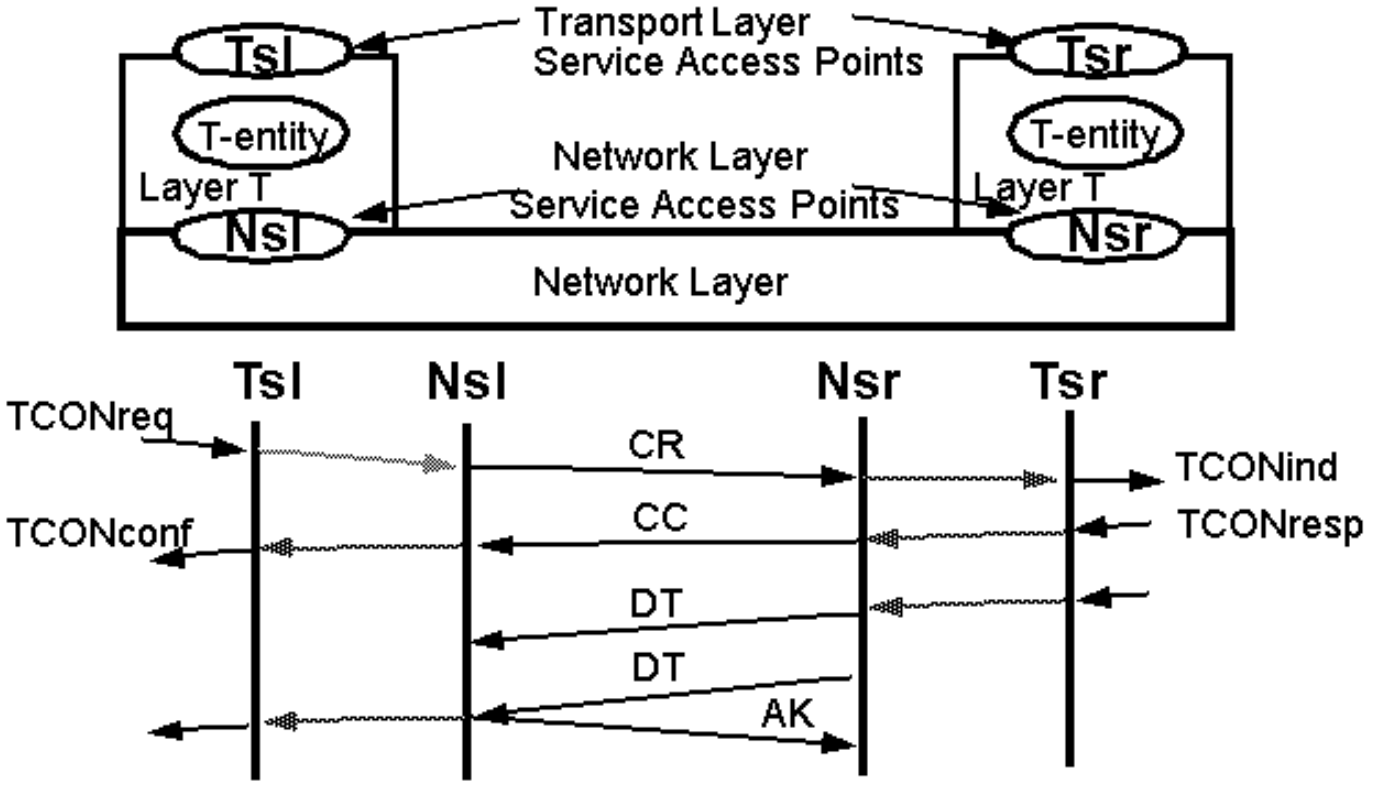
Protocol Specification using CFSM model

Lecture 30

Protocol Engineering



The sequence chart is good at capturing the normal/specific scenario of system interactions.



It is difficult to show several/all possible scenarios simultaneously on a chart and that often leads to ambiguity in the specification.

The use of Formal Methods

- **Provide a formal and unambiguous way** of designing and documenting protocols.
- **Allow formal analysis** (*verification*/*validation*/*performance analysis*) before protocols are implemented.
- **Allow automatic and direct generation of executable programs** from the formal specification.

Specification Languages

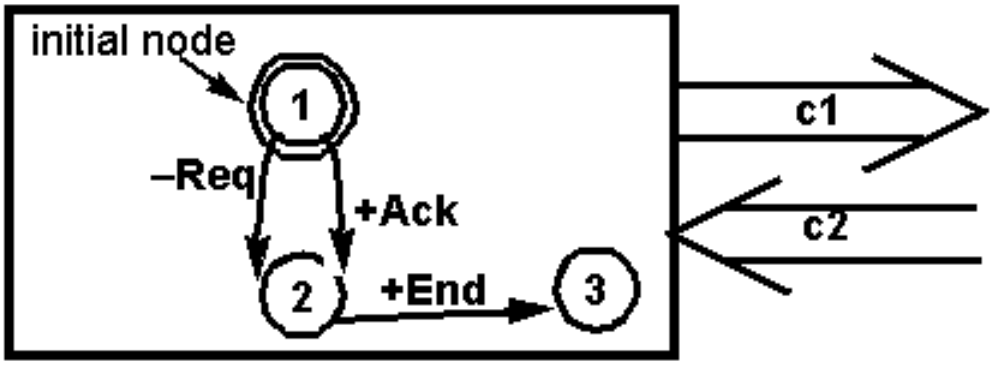
- Informal methods
  - such as the sequence chart
- Formal methods
  - **State Transition Models**
    -- Finite State Machines (FSM),
    -- Communicating FSM (CSFM),
    -- Petri nets
  - **Programming Languages Models**
    -- Abstract Programs
    -- CCS (Calculus of Communicating systems), CSP
    -- Temporal logic
  - **Hybrid Models**
    -- Extended FSM (EFSM)

Language Standards
- SDL (FSM + extensions)
- Estelle (EFSM + extended Pascal)
- LOTOS (CCS)
- ASML

Communicating Finite State Machines (CFSM)

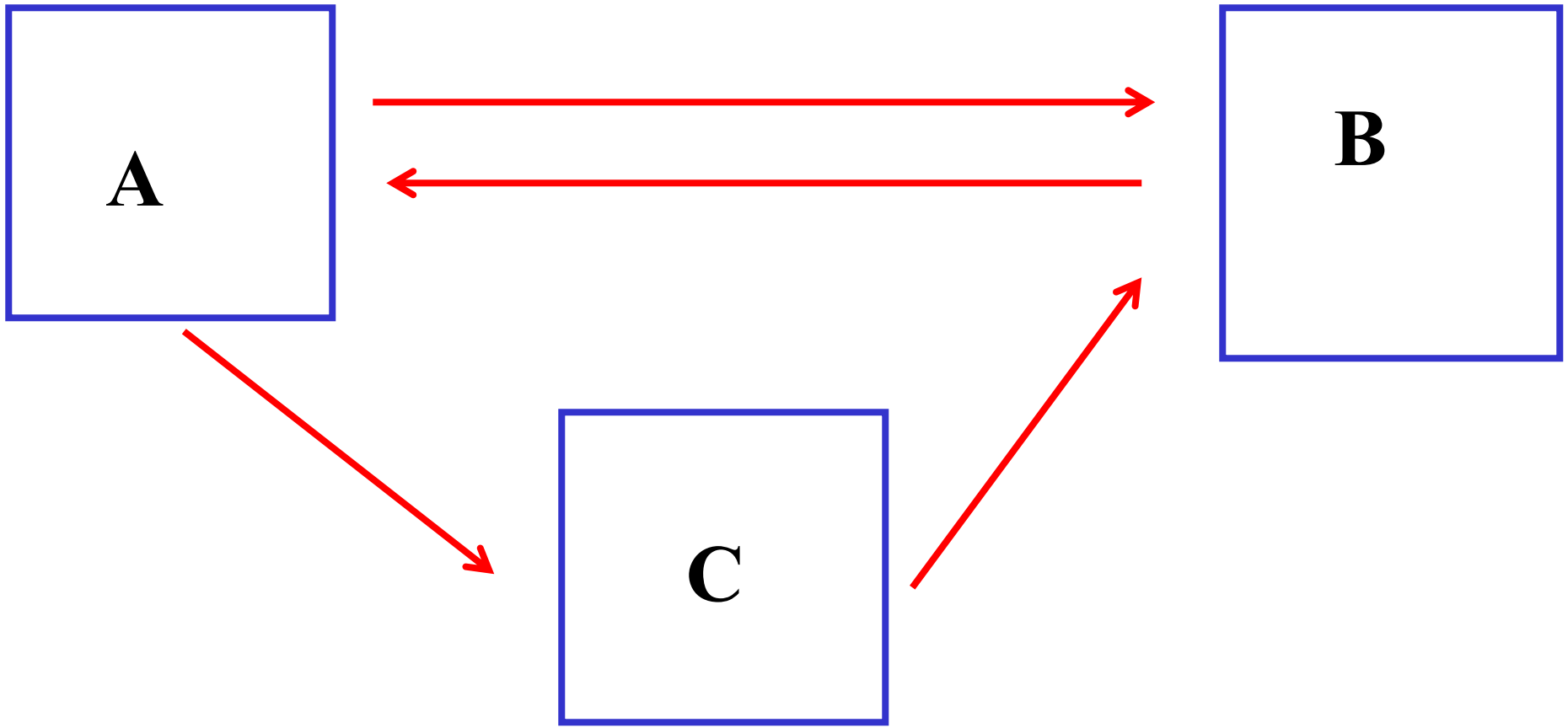
- **Protocol is described as a set of Communicating Finite State Machines**.
- Each **CFSM represents a component** (or process) **of the network** (in OSI term, a protocol entity, e.g. sender, receiver).
- Each **CFSM is represented by a directed labelled graph** where
  - **Nodes represent states** (conditions) of the process;
  - **Edges represent transitions** (events) of the process.

Transitions



- Transitions include actions
  - **taken the process** (e.g. the sending a message) or
  - **external stimuli** (e.g. the reception of a message).
- The sending message transition is labelled as **-Msg**
  - where Msg is the type of messages being sent.
- The receiving message transition is labelled as **+Msg**
  - where Msg is the head message on the incoming FIFO queue of the CFSM

Network of CFSMs



CFSM operating semantic

- **The channels** that connect CFSM's are assumed to be FIFO queues.
  - An error-prone channel is modelled as a CFSM.
- **Initial node**--starting state of a CFSM.
  - Final node -- no transition.
  - Receiving node -- all (outgoing) transitions are receiving transitions. If no message or incorrect msg in the channel, the node will be blocked.
  - Sending node -- all transitions are sending transitions. They are not blocked.
  - Mix node -- has both receiving and sending transition.

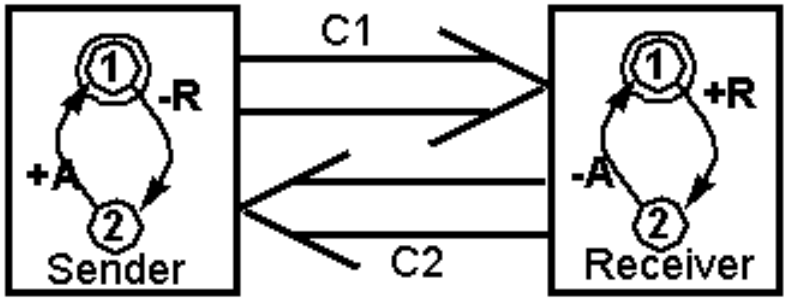
- **Starting at the initial node**, a CFSM traverses the nodes and transitions.
  - The node currently being visited is called the current node.
- When a machine traverses a sending transition, it sends/appends a message with the same label to its outgoing channel.

- A machine at a node cannot traverse its receiving transition unless there is a message matched with the same label on the head of its incoming channel.
- When a machine traverses a receiving transition, it removes the matched head message of its incoming channel.

- **Among several possible transitions, a machine traverses one non-deterministically**

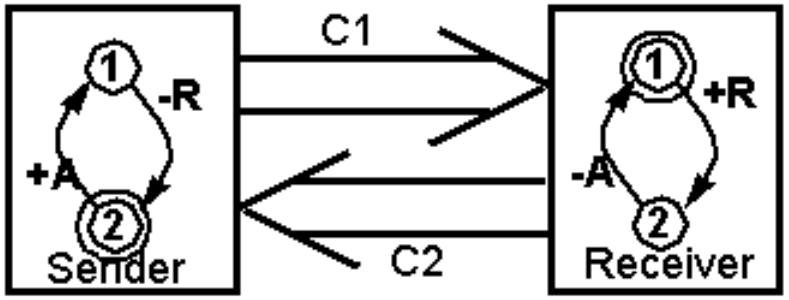
Networks of CFSMs

- **Example 1:** Simple request-response protocol.



R: request
A: Acknowledgment

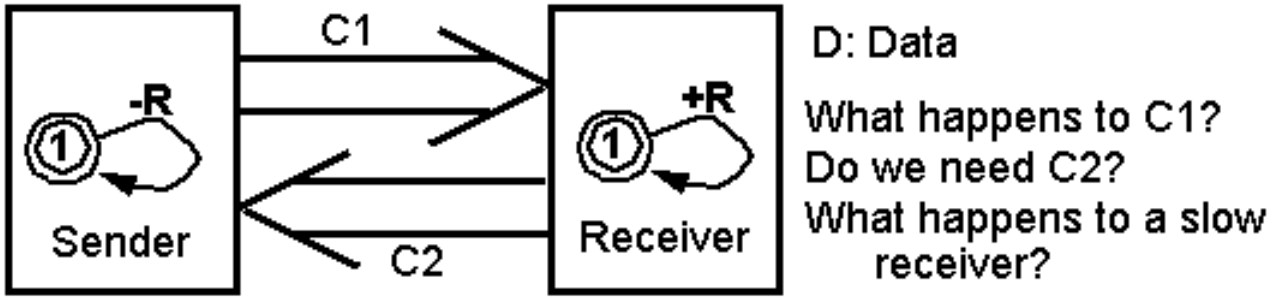
- **Example 2:** What happens if we change the initial node of a CFSM?



R: request
A: Acknowledgment

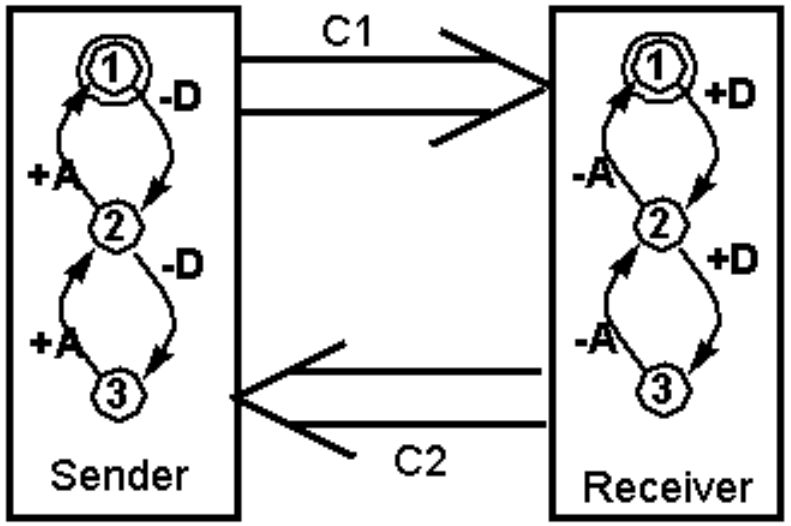
Networks of CFSMs

- **Example 3:** An aggressive protocol with a self-sending



Networks of CFSMs

- **Example 4:** A simple sliding window protocol with a window size of 2.

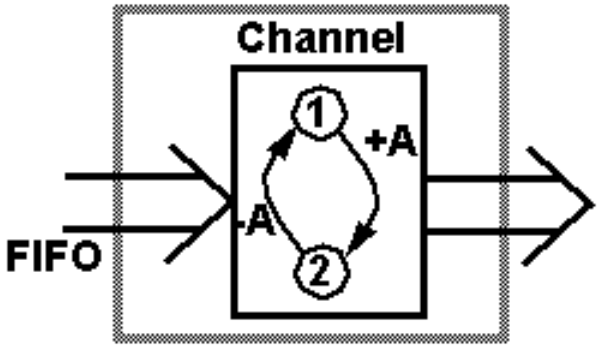


D: Data
A: Acknowledgment

What if we distinguish the two data messages and the two acks?

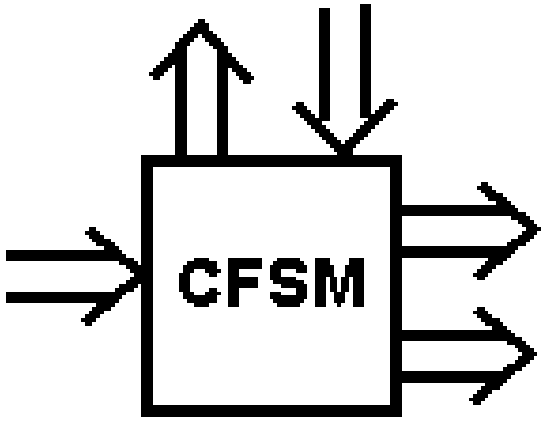
CFSM Modeling Exercises

- How to specify the channel behavior as a CFSM which
  - loses every other packet,
  - loses packets sometimes,
  - loses and corrupts the packet sometimes?



CFSM Modeling Exercises

- How to extend the model to specify CFSMs with the multiple channels?

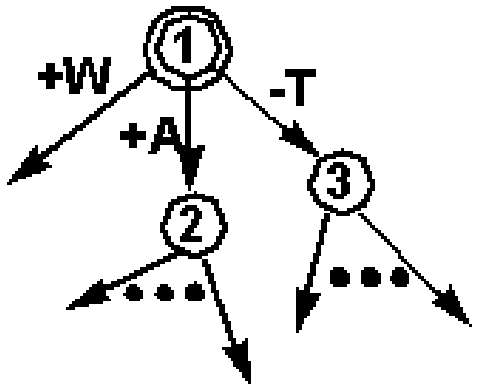


Pros and Cons of the CFSM model

- CFSM deals only with the state-transition aspect of protocols;
  - it does not address the data aspect of protocols, e.g., message content or format.
- It can not handle protocols where state variables have a wide range of values.
  - Extended FSM were proposed but EFSM becomes difficult to analyse.
- The FIFO channels assumption in CFSM is very powerful,
  - just think about how to model an unbounded or even a large buffer using Petri net.

Pros and Cons of the CFSM model

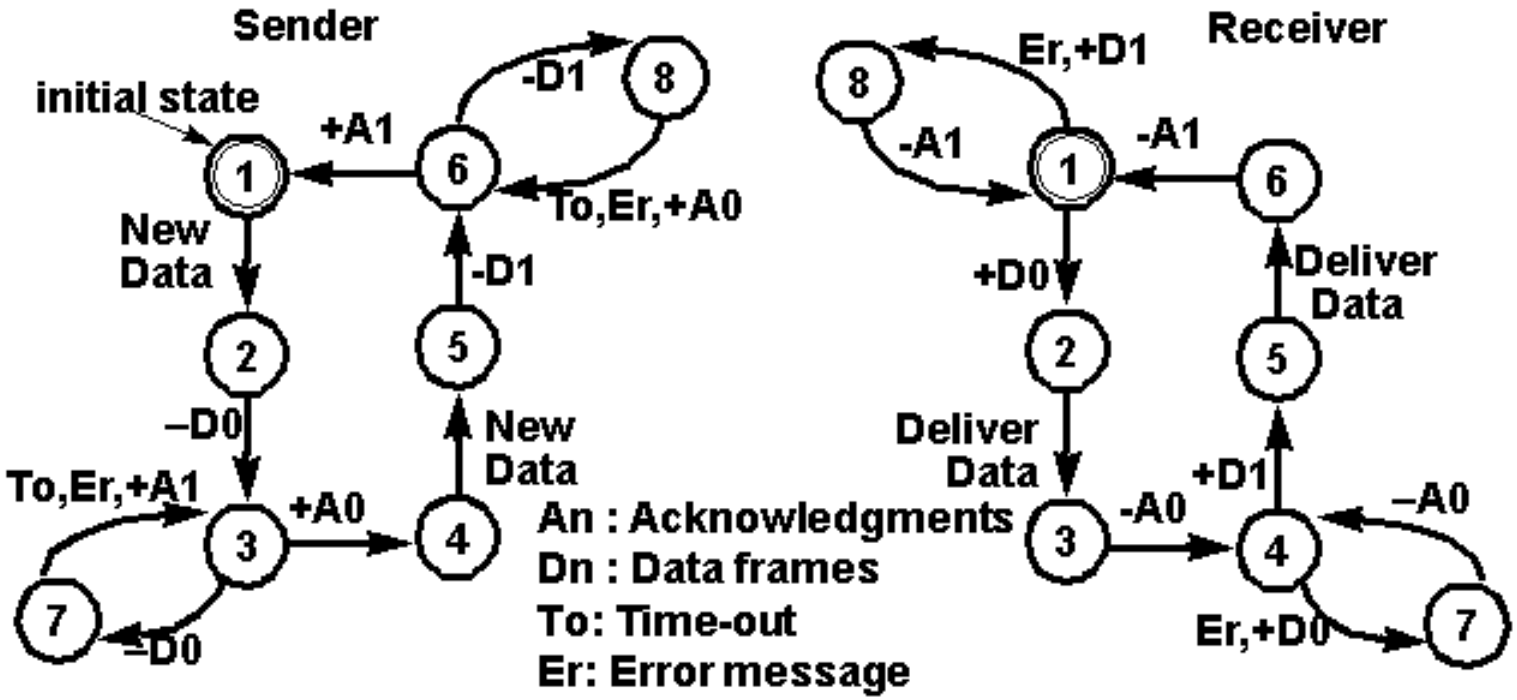
- **CFSM is an abstract model.**
  - The non-determinism in the execution of transitions of a mix node may result in different implementation.
  - You can always expand the specification, e.g., replacing node 1 with a subgraph



The Alternating Bit Protocol as CFSMs

- The **Alternating Bit Protocol** is used to guarantee the correct data delivery between a sender and receiver connected by an error channel that loses or corrupts messages.
  - *It got the name since it uses only one additional control bit in the message and this control bit only alternates when the previous message is correctly received.....*

The Alternating Bit Protocol



Outline for the next lecture

- *Verifying the Alternating Bit Protocol*
- *Protocol Verification using Reachability Analysis*
- *Protocol Design Errors*
- *Protocol Verification Exercises*
- *Pros and Cons of Reachability Analysis*
- *Tools for specification development*