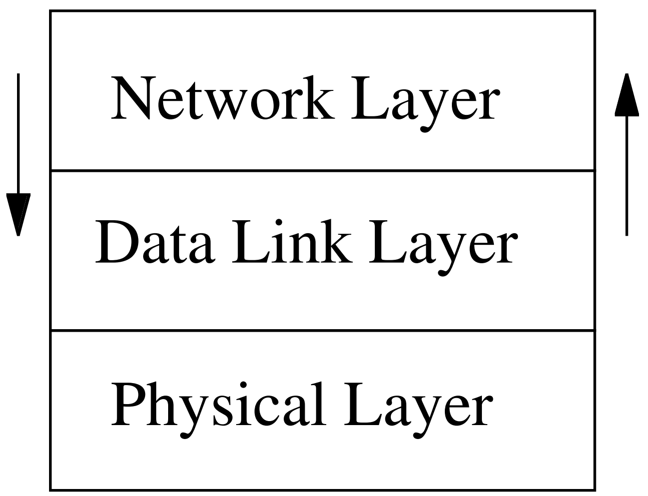
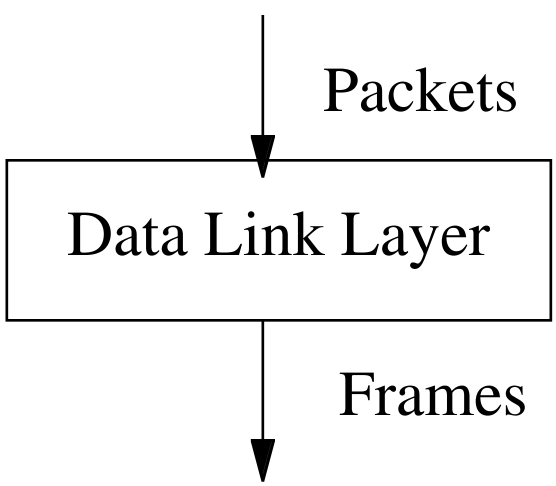
Error and Flow Control

Network and data link layers only communicate via messages with specific data structures.

The data link layer processes those structures with a set of procedures.





Error and Flow Control

- Data Structures
- Procedures
- Events
- Protocols

Error and Flow Control

Data Structures

- Bit ==> 0,1
- SequenceNr ==> Sequence number
- Packet ==> Information from the network layer
- Frame
  - Kind
    - control (supervisory)
    - data
  - Seq ==> SequenceNr
  - Ack
  - Info ==> packet

Error and Flow Control

Procedures

- FromNetworkLayer
  - Fetch information from the network layer
- ToNetworkLayer
  - Deliver information to the network layer
- FromPhysicalLayer
  - Fetch information from the physical layer
- ToPhysicalLayer
  - Deliver information to the physical layer

Error and Flow Control

Procedures

- StartTimer
- StopTimer
- StartAckTimer
- StopAckTimer
- EnableNetworkLayer
  – Turn on flow of information from the network layer
- DisableNetworkLayer
  – Turn off flow of information from the network layer

Error and Flow Control

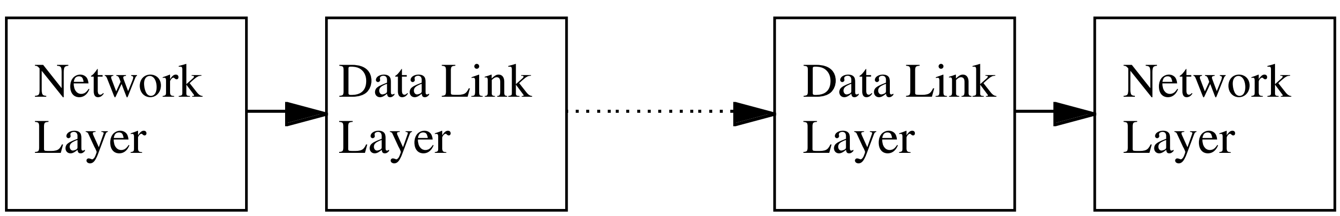
Events

- Networks are Asynchronous
  - Arrival time of packet and acknowledgments are unknown
- Arrival of packet and acknowledgments triggers some action by the protocol
  - Action is a function of the type of arrival
  - State of the protocol
- Examples:
  - FrameArrival
  - CksumErr

Error and Flow Control

Protocol 1: The Unrestricted Simplex Protocol

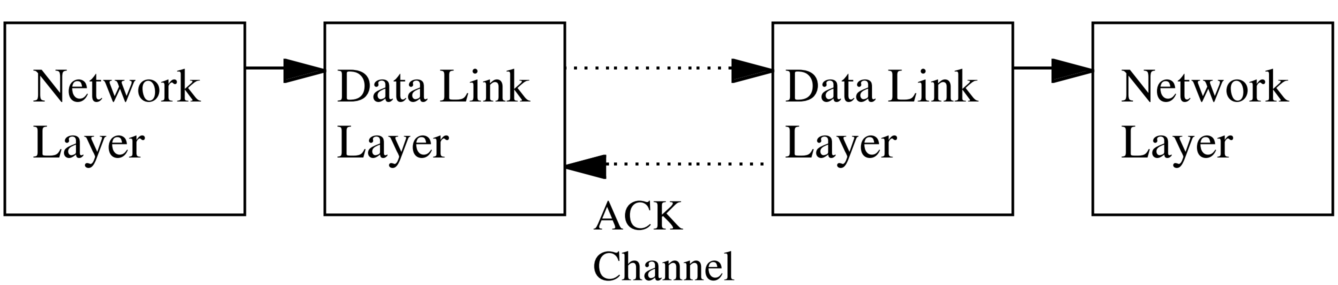
- Assumptions
  - One directional information flow
  - Infinite buffers
  - No errors
  - Network Layer always has a packet to send



Error and Flow Control

Protocol 2: The Simplex Stop & Wait Protocol

- Assumptions
  - One directional information flow
  - No errors
  - Network Layer always has a packet to send
  - Finite receive buffers
    - Finite buffer means that there must be some way to stop the transmitter from sending when the buffer is full



Error and Flow Control

Protocol 3: The Simplex Protocol for a Noisy Channel

- Assumptions
  - One directional information flow
  - Network Layer always has a packet to send
  - Finite receive buffers
- Data link protocols must address
  - When to retransmitt
  - What to retransmitt

Error and Flow Control

Protocol 3: The Simplex Protocol for a Noisy Channel

- **Timeout** to determine when to retransmitt
- Example:
  - Assume a 1 ms propagation time
  - Assume a .1 ms receiver packet processing time
  - Timeout interval >2.1 ms
    - If no acknowledgment received in 2.1 ms then,
      - Packet in error
      - Acknowledgment lost
  - Timeout interval too short
    - Duplicate packets
  - Timeout interval too long
    - Reduced throughput

Error and Flow Control

Protocol 3: The Simplex Protocol for a Noisy Channel

- **Sequence numbers** are used to determine what to retransmitt
  - Transmitter assigns a number to each frame
  - Receiver keeps track of the expected frame number
  - If the received sequence number does not *match* what is expected, then the frame is dumped

•Error and Flow Control

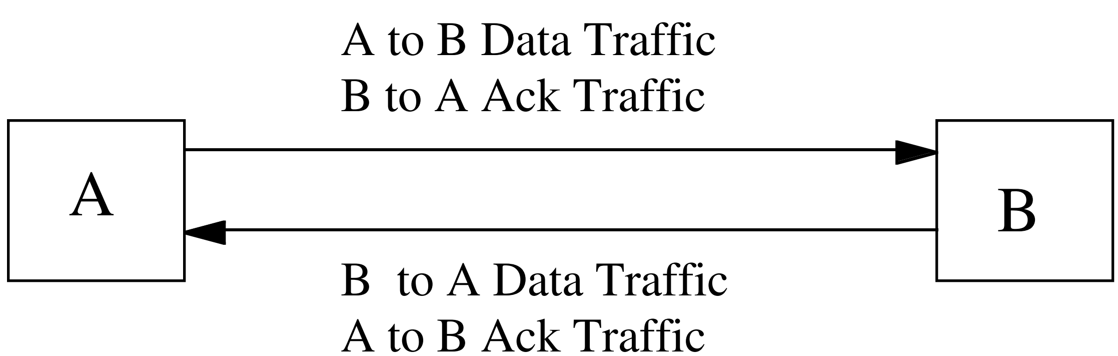
Protocol 3: The Simplex Protocol for a Noisy Channel

Event	Transmitter Action	Receiver Action
Packet arrived to DLL from NL	Store in Buffer. Insert s.seq. Transmit Frame & start timer.	
Frame arrive at receiver		Get frame from physical layer. If in error then dump it. If in-sequence and error free then, send to NL, increment Frame Expected, Send Acknowledgement
Receive Acknolwedgement	Remove from buffer Get next packet from NL Increment s.seq Store packet in Buffer. Insert s.seq. Transmit Frame & start timer.	

Error and Flow Control

Sliding Window Protocols

- Assumptions
  - Two directional information flow
  - Network Layer always has a packet to send
  - Finite receive buffers
  - Piggybacking
    - Put Acknowledgments in reverse traffic flow
    - Increases protocol efficiency
    - Reduces interrupts



Error and Flow Control

Sliding Window Protocols

- Transmitter keeps a list of frames it can send
  - Sending window
- Receiver keeps a list of frames it can receive
  - Receiving window
- Lists are sequence numbers in range $0 - 2^n-1$
- This allows 2^n-1 packets to be sent before getting and acknowledgment
- Requires 2^n-1 packets buffers