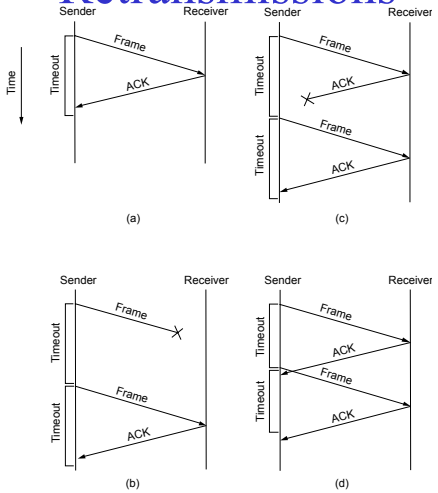
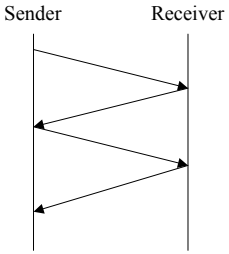
Housekeeping

Acknowledgements, Timeouts & Retransmissions



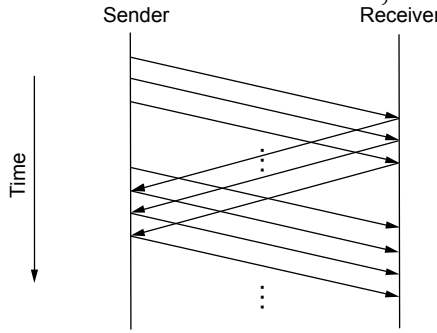
Stop-and-Wait



- Problem: keeping the pipe full
- Example
  - 1.5Mbps link x 45ms RTT = 67.5Kb (8KB)
  - 1KB frames imples 1/8th link utilization

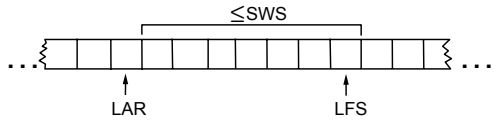
Sliding Window

- Allow multiple outstanding (un-ACKed) frames
- Assume no reording of frames
- Upper bound on un-ACKed frames, called *window*



SW: Sender

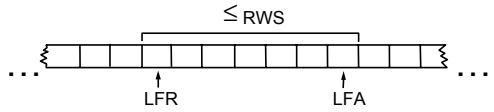
- Assign sequence number to each frame (**SeqNum**)
- Maintain three state variables:
  - send window size (**SWS**)
  - last acknowledgment received (**LAR**)
  - last frame sent (**LFS**)
- Maintain invariant: **LFS - LAR <= SWS**



- Advance **LAR** when ACK arrives
- Buffer up to **SWS** frames

SW: Receiver

- Maintain three state variables
  - receive window size (**RWS**)
  - largest acceptable frame (**LAF**)
  - last frame received (**LFR**)
- Maintain invariant: **LAF - LFR <= RWS**



- Frame **SeqNum** arrives:
  - if **LFR < SeqNum < = LAF** ⟶ accepted
  - if **SeqNum < = LFR** or **SeqNum > LAF** ⟶ discarded
- Send cumulative ACKs

Sequence Number Space

- **SeqNum** field is finite; sequence numbers wrap around
- Sequence number space must be larger then number of outstanding frames
- **SWS <= MaxSeqNum-1** is not sufficient
  - suppose 3-bit **SeqNum** field (0..7)
  - **SWS=RWS=7**
  - sender transmit frames 0..6
  - arrive successfully, but ACKs lost
  - sender retransmits 0..6
  - receiver expecting 7, 0..5, but receives second incarnation of 0..5
- **SWS < (MaxSeqNum+1)/2** is correct rule
- Intuitively, **SeqNum** "slides" between two halves of sequence number space

Questions

- How big should the SWS be?
- How many bits are needed for the seqno field?
- What happens when SWS != RWS?
- How long should the timeouts be?
- When should reliability be provided at the link layer? At the transport layer?

Concurrent Logical Channels

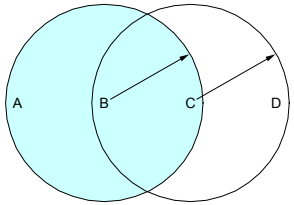
- Multiplex 8 logical channels over a single link
- Run stop-and-wait on each logical channel
- Maintain three state bits per channel
  - channel busy
  - current sequence number out
  - next sequence number in
- Header: 3-bit channel num, 1-bit sequence num
  - 4-bits total
  - same as sliding window protocol
- Separates *reliability* from *order*

Wireless LANs

- IEEE 802.11 family
- Bandwidth: 1, 2 or 11 Mbps, w/ higher proposals
- Physical Media
  - spread spectrum radio (2.4GHz)
  - diffused infrared (10m)

Collisions Avoidance

- Similar to Ethernet
- Problem: *hidden* and *exposed* nodes
- A & C Hidden from each other when sending to B



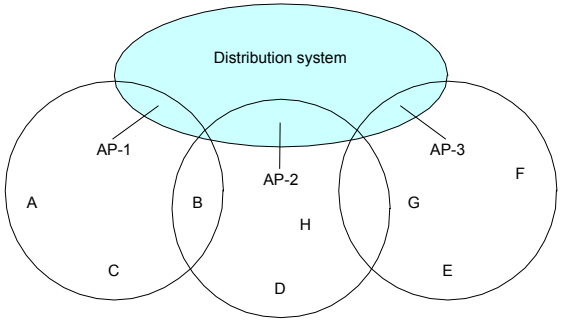
- B exposed to C when C sends to D

MACAW

- Multiple Access with Collision Avoidance for Wireless
- Sender transmits **`RequestToSend`** (RTS) frame
- Receiver replies with **`ClearToSend`** (CTS) frame
- Neighbors…
  - see CTS: keep quiet
  - see RTS but not CTS: ok to transmit
- Receive sends ACK when has frame
  - neighbors silent until see ACK
- Collisions
  - no collision detection
  - known when don't receive CTS
  - exponential backoff

Supporting Mobility

- Case 1: *ad hoc* net working
- Case 2: *access points* (AP)
  - tethered
  - each mobile node associates with an AP



Mobility (cont)

- Scanning (selecting an AP)
  - node sends **`Probe`** frame
  - all AP's w/in reach reply with **`ProbeResponse`** frame
  - node selects one AP; sends it **`AssociateRequest`** frame
  - AP replies with **`AssociationResponse`** frame
  - new AP informs old AP via tethered network
- When
  - active: when join or move
  - passive: AP periodically sends **`Beacon`** frame

Spread Spectrum

- Idea
  - spread signal over wider frequency band than required
  - originally designed to thwart jamming
- Frequency Hopping
  - transmit over random sequence of frequencies
  - sender and receiver share…
    - pseudorandom number generator
    - seed
  - 802.11 uses 79 x 1MHz-wide frequency bands

Spread Spectrum (cont)

- Direct Sequence
  - for each bit, send XOR of that bit and *n* random bits
  - random sequence known to both sender and receiver
  - called *n*-bit *chipping code*
  - 802.11 defines an 11-bit chipping code

