Link Layer: Basic Techniques

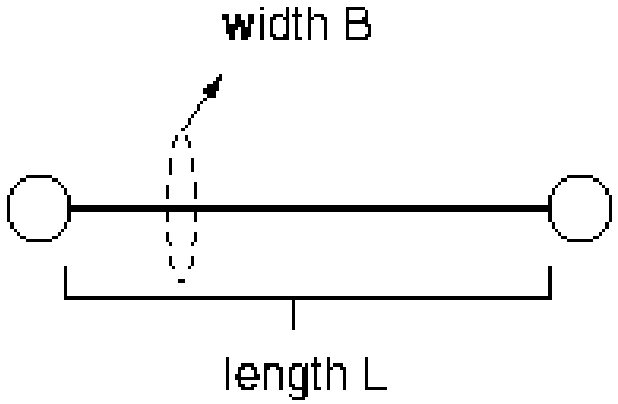
Data Transmission

Link speed unit: bps

$\longrightarrow$ abstraction

$\longrightarrow$ ignore carrier frequency, coding etc.

Simplest case: point-to-point link



$\longrightarrow$ wired or wireless

Interested in *completion time*:

$\longrightarrow$ time elapsed between sending/receiving first bit

$\longrightarrow$ i.e., how long will it take?

- Single bit:

  $\rightarrow \approx L/\text{SOL}$ (lower bound)

  $\rightarrow$ latency (or propagation delay)

  $\rightarrow$ optical fiber, wireless: exact

- Multiple, say S, bits:

  $\rightarrow \approx L/\text{SOL} + S/B$

  $\rightarrow$ latency + transmission time

Latency vs. transmission time: which dominates?

$\longrightarrow$ a lot to send, a little to send, ...

$\longrightarrow$ satellite, Zigbee, WLAN, broadband WAN

Reliable Transmission

Main method: ARQ (Automatic Repeat reQuest)

$\longrightarrow$ use retransmission

$\longrightarrow$ used in both wired/wireless

- function duplication

  $\rightarrow$ link layer, transport layer, etc.

- alternative: FEC (forward error correction)

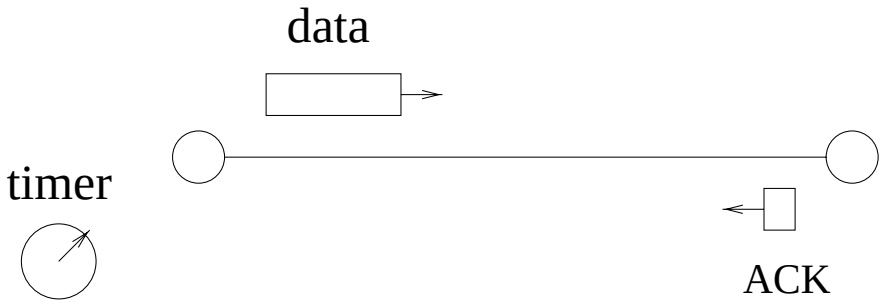
  $\rightarrow$ transmit redundant information

  $\rightarrow$ not assured

  $\rightarrow$ pros and cons?

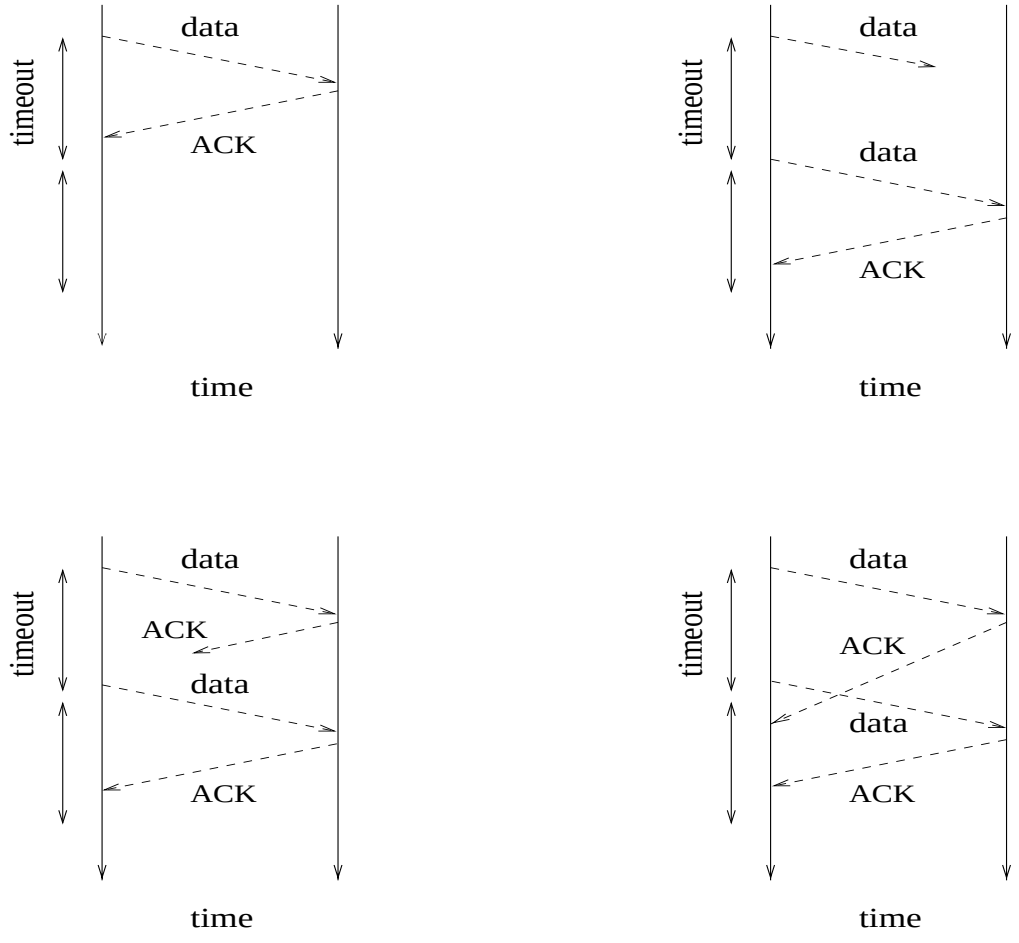
ARQ: three components

- timer
- acknowledgment (ACK)
- retransmit



Special case: stop-and-wait

Handle one packet (i.e., frame) at a time.



Issue of RTT (Round-Trip Time) & timer management:

- what is proper value of timer?

  $\rightarrow$ RTT estimation

- easier for single link

  $\rightarrow$ RTT is more well-behaved

- more difficult for multi-hop path in internetwork

  $\rightarrow$ latency + queueing effect

A "good" thing about stop-and-wait:

$\longrightarrow$ simple throughput formula

Stop-and-wait throughput (bps):

- RTT
- frame size (bits)

  $\longrightarrow$ throughput = frame size / RTT

Another important problem: not keeping the pipe full.

$\longrightarrow$ delay-bandwidth product

$\longrightarrow$ volume of data travelling on the link

High throughput: want to keep the pipe full

Ex.: Link BW 1.5 Mbps, 45 msec RTT

- if frame size 1 kB, then throughput:

  $\rightarrow 1024 \times 8/0.045 = 182$ kbps

  $\rightarrow$ utilization: only 182 kbps/1500 kbps $= 0.121$

- note: delay-bandwidth product

  $\rightarrow 1.5$ Mbps $\times$ 45 msec $= 67.5$ kb ≈ 8 kB

What happens to utilization if RTT increases to 90 msec?

What happens if bandwidth increases to 3 Mbps?

$\longrightarrow$ how to reduce bandwidth wastage?

Sliding Window Protocol

$\longrightarrow$ send block (i.e., window) of data

Issues:

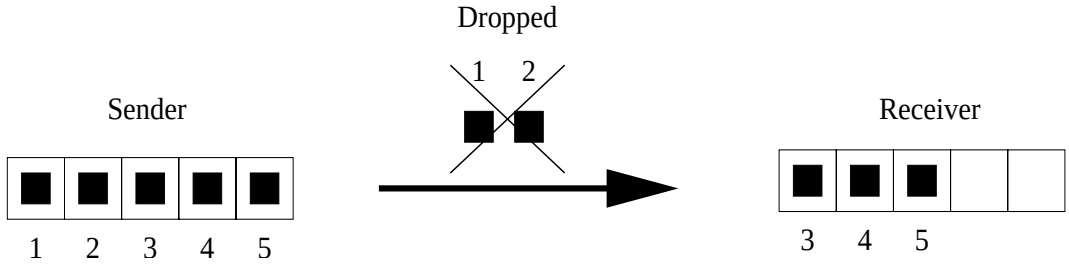
- Shield application process from reliability management chore

  $\rightarrow$ exported semantics: continuous data stream

  $\rightarrow$ simple app abstraction: e.g., `read` system call

- Both sender and receiver have limited buffer capacity

  $\rightarrow$ task: plug holes & flush buffer



Simple solution when receiver has infinite buffer capacity:

- sender keeps sending at maximum speed
- receiver informs sender of holes

  → "I'm missing this and that"

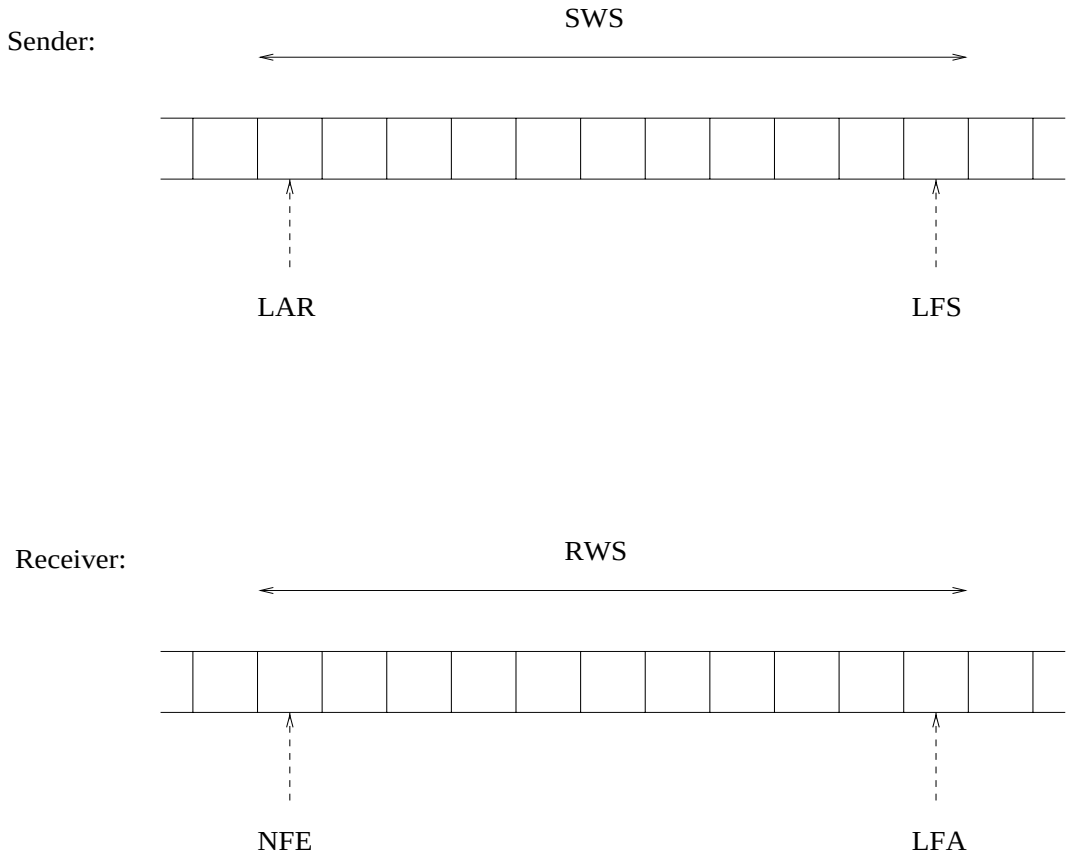
  → called negative ACK
- sender retransmits missing frames

Drawbacks?

What about positive ACK?

⟶ pros and cons

Sliding window operation with positive ACK:



- *SWS*: Sender Window Size (sender buffer size)
- *RWS*: Receiver Window Size (receiver buffer size)
- *LAR*: Last ACK Received
- *LFS*: Last Frame Sent
- *NFE*: Next Frame Expected
- *LFA*: Last Frame Acceptable

Assign sequence numbers to frames.

$\longrightarrow$ IDs

Maintain invariants:

- $\text{LFA} - \text{NFE} + 1 \leq \text{RWS}$
- $\text{LFS} - \text{LAR} + 1 \leq \text{SWS}$

Sender:

- Receive ACK with sequence number X
- Forwind LAR to X
- Flush buffer up to (but not including) LAR
- Send up to $\text{SWS} - (\text{LFS} - \text{LAR} + 1)$ frames
- Update LFS

Receiver:

- Receive packet with sequence number Y
- Forwind to (new) first hole & update NFE

  $\rightarrow$ NFE need not be $Y + 1$
- Send cumulative ACK (i.e., NFE)
- Flush buffer up to (but not including) NFE to application
- Update LFA $\leftarrow$ NFE + RWS $- 1$

Sequence number wrap-around problem:

$$\text{SWS} < (\text{MaxSeqNum} + 1)/2$$

$\longrightarrow$ why?

$\longrightarrow$ consider special case: stop-and-wait

$\longrightarrow$ is sequence number needed?