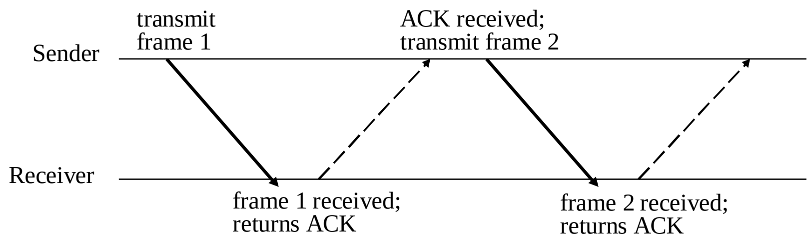
Data Link Layer, Part 5
Sliding Window Protocols

Preface

- We are about to discuss a series of protocols used between a pair of directly-connected sender and receiver.
- These protocols serve three purposes
  1. to guarantee delivery reliability,
  2. to enforce correct ordering of frames delivered to the network layer at the receiving end, and
  3. to provide flow control.

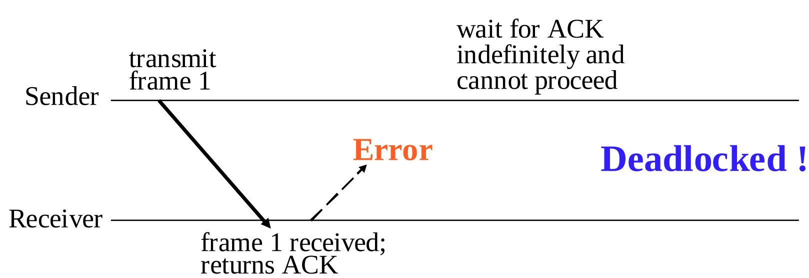
Stop And Wait

- ❑ The sender transmits a frame.
- ❑ If the receiver successfully receives the frame, it delivers the frame to its network layer and returns an **acknowledgement** (ACK) to the sender.
- ❑ Only after receiving the ACK will the sender be allowed to transmit the next frame.



Problem

- ❑ This simple protocol provides flow control but dose not enforce reliability
  - – Actually, it fails in noisy links.
  - – What would happen when a transmission (the frame or the ACK) is corrupted ?

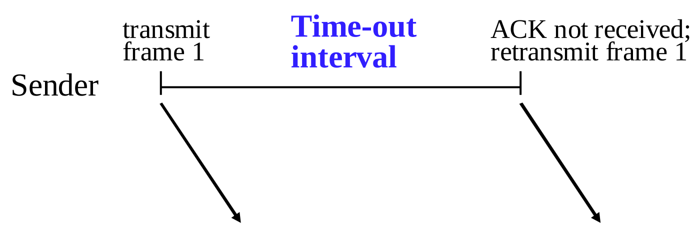


Discussion

- ❑ What could cause the receiver not to receive the frame successfully ?
  - – transmission errors reported by the error detection mechanism
  - – the frame is so corrupted by noises that the receiver dose not even recognize its arrival
  - – when the frame arrives at the receiver, there is not enough memory to accommodate it

The Time-out Mechanism

- ❑ The sender re-transmits a frame if the ACK does not arrive within some predetermined length of time.



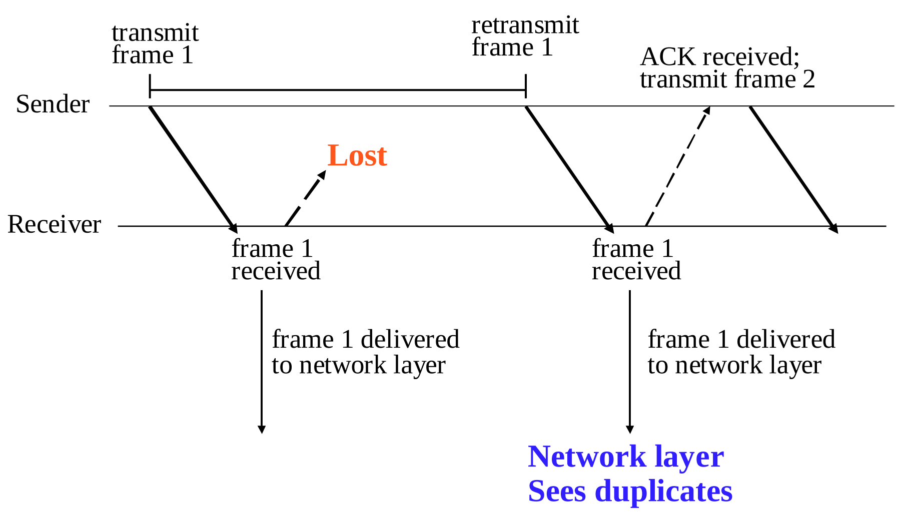
- ❑ This implies that the sender DLL must keep the frame until it receives the ACK.

Discussion

- Causes for not receiving the ACK:
  - the frame is not received due to previously discussed causes
  - the frame is successfully delivered but the ACK is not received by the sender
- This protocol is still flawed: although the delivery of the frame is guaranteed, the receiver's network layer may see multiple copies of a frame.

CS 656 7

The Problem of Duplicates



CS 656 8

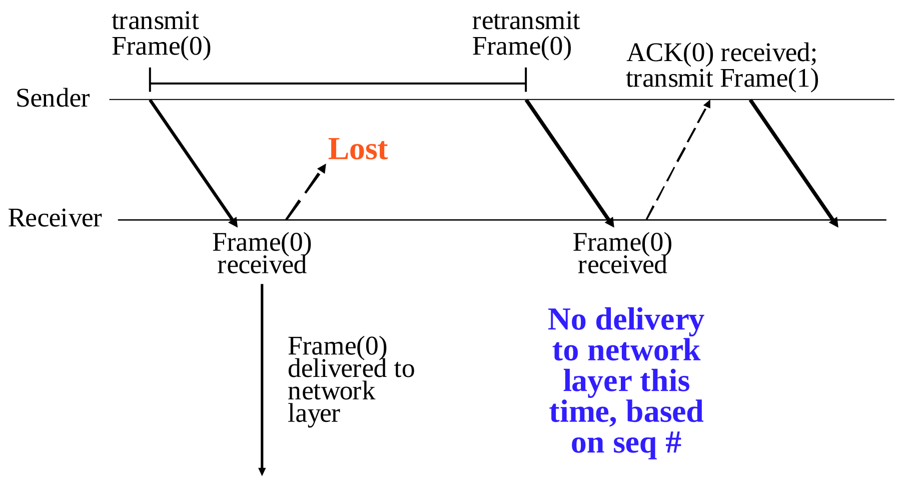
Sequence Numbers

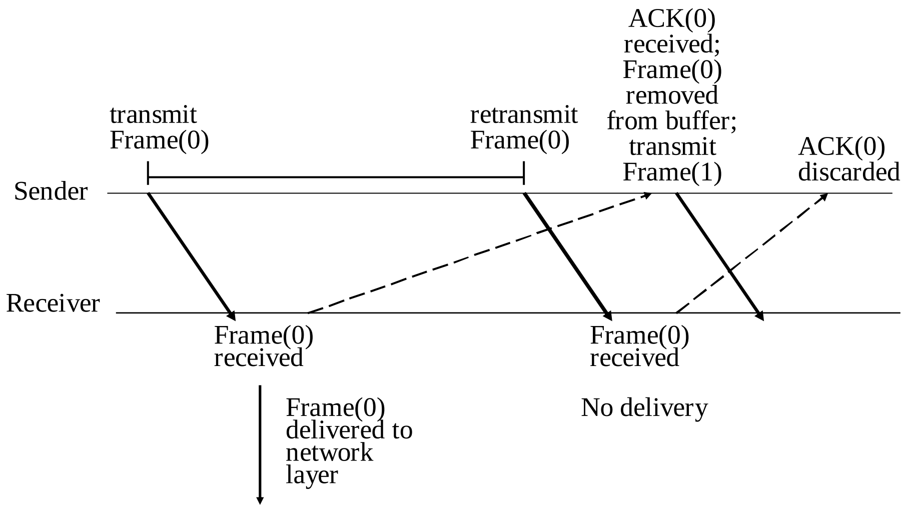
- Frames are numbered alternately:

  0, 1, 0, 1, …

  – These are 1-bit sequence numbers.
- Receiver acknowledges the correct frames with an ACK of the same sequence number as the frame.
- If not receiving the corresponding ACK after a time-out period, the sender retransmits the frame.

Example





- ❑ When Frame(0) is received the second time, it is acknowledged but not delivered to network layer

Improving Performance

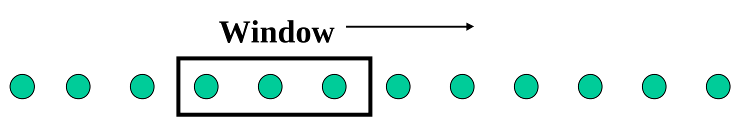
- ❑ "Stop-and-wait" wastes bandwidth.
- ❑ Solution is to allow multiple outstanding frames.
  - – that is, to send the next frame before the current one is acknowledged.
- ❑ Two approaches:
  1. **go-back-n:** receiver discards all frames following an error, forcing the sender to "go back" to the lost frame, and retransmit all frames from that point
  2. **selective repeat:** receiver stores correct frames following the lost one(s); sender retransmits only lost frames

Go-Back-N

- Each frame is tagged with an *n*-bit sequence number.
- Only **2^n - 1** frames may be outstanding.
- The sender maintains a set of 2^n - 1 buffers, called a (sliding) **window**, to keep unacknowledged frames.
- When transmitting a frame, the frame is also copied to a slot of the window.
  - the copy is for retransmissions
  - the slot is freed when the frame's ACK arrives

Sliding Windows

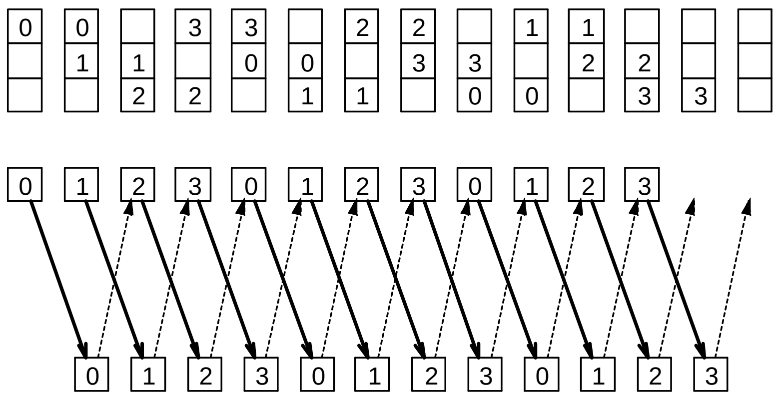
- Image a sequence of frames waiting for transmission.
- Both Go-back-N and Selective-repeat define a window that slides from left to right over time.



- At a given moment, only those frames in the window can be transmitted.
- They are thus called **Sliding Window Protocols**.

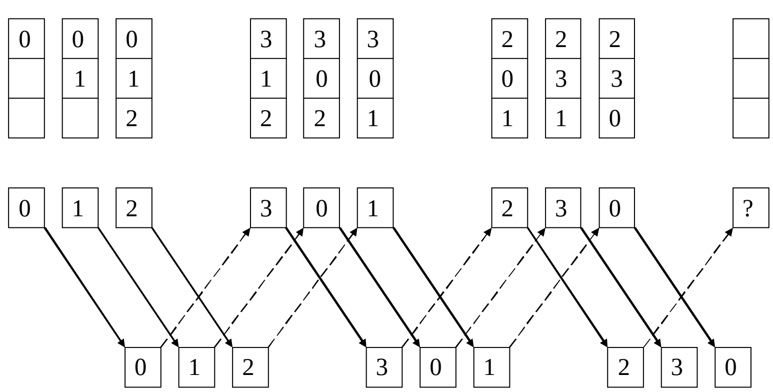
Example 1: Large Window

When window size is large *relative* to the round-trip time, we can take full advantages of the bandwidth.



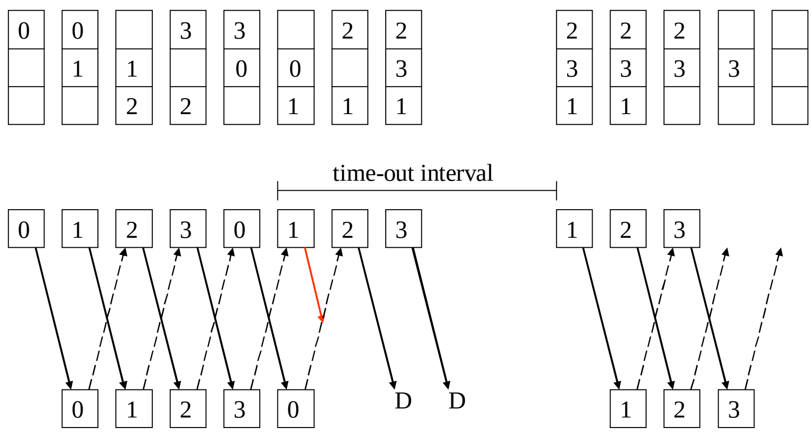
Example 2: Small Window

If window size is small *relative* to the round-trip time, then the “stop-and-wait” phenomenon still occurs.

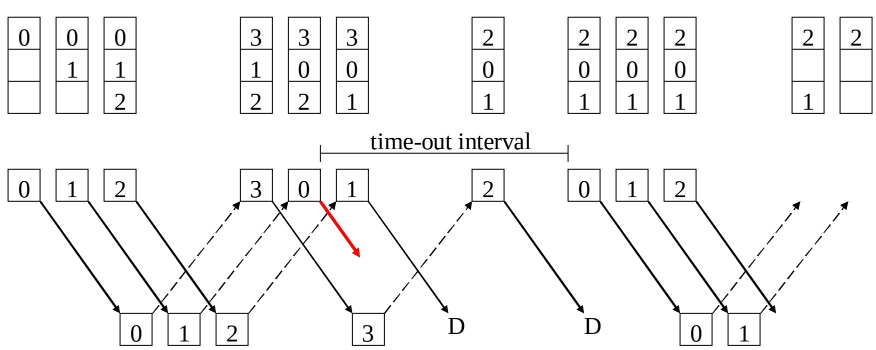


Go-Back-N with Transmission Errors

Case 1: *n* large enough



Case 2: *n* too small



Why the Restriction 2^n-1 ?

- Consider n=2. Let us allow 4 outstanding frames and see what happens.
- The receiver sees the following frames:

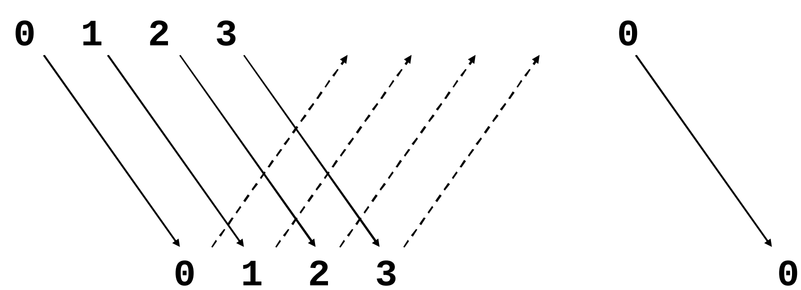
0 1 2 3 0

- Question: what is the second frame 0 ?
  - There are two legitimate possibilities, and we have no way to tell.

CS 656 19

Scenario 1

- Nothing wrong; temporarily nothing to send after 3

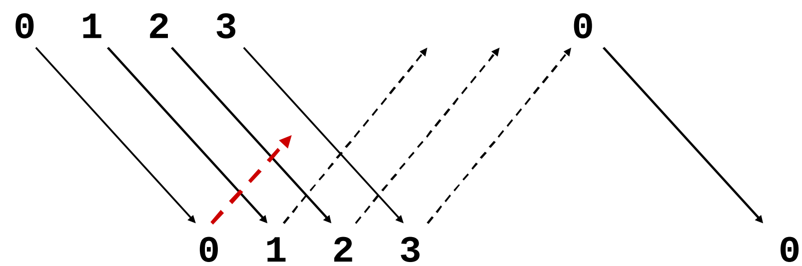


- In this case, the second frame 0 is a new frame (the 5th frame).

CS 656 20

Scenario 2

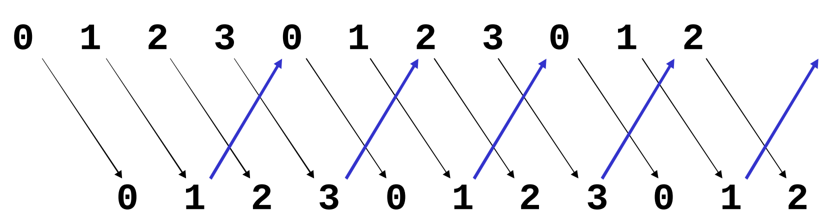
- Timeout and retransmit 1st frame



- The second frame 0 is a duplicate.
  - It must *not* be delivered to the network layer.
  - However, it *is* acknowledged to prevent the sender from time-out again.

Optimizations

- The receiver attaches ACKs to outgoing frames destined to the sender; this technique is called **piggybacking**.
- Further, we can "aggregate ACKs" that is, the ACK of the *i*-th frame also acknowledges the receipt of all the frames up to *i*.



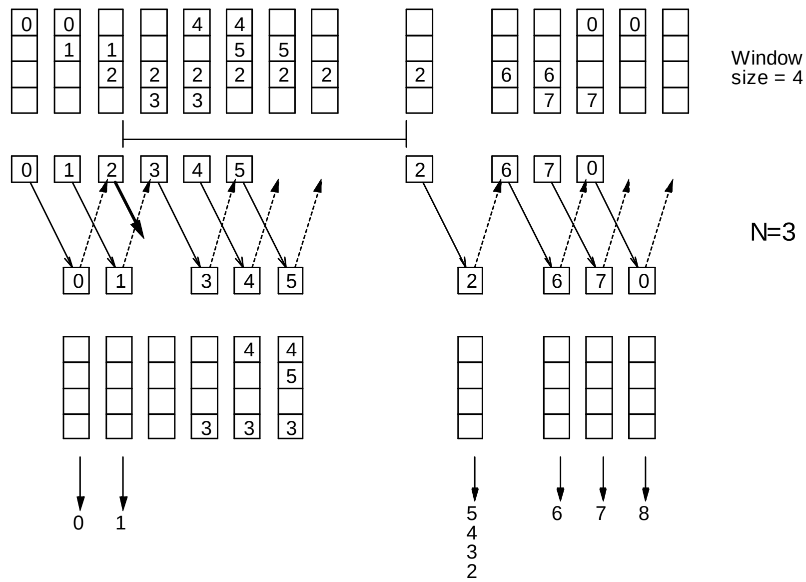
Discussions

- One issue of piggybacking is how long the receiver should wait for an outgoing data frame
  - waiting too long may cause the sender to time out and retransmit unnecessarily
  - Impatient receivers, on the other hand, lose the opportunities of piggybacking
- Another optimization is to have the receiver, upon a corrupted frame, immediately replies a **negative acknowledgement** (NACK), forcing the sender to retransmit before time-out.

Selective Repeat

- Sender maintains a window of size $\mathbf{2^{n-1}}$ to keep outstanding, unacknowledged frames.
- Sender retransmits a frame that is not acknowledged after time-out.
- Receiver acknowledges the receipt of a frame with an ACK containing the sequence no. of the frame.
- Receiver also maintains a window of size 2^{n-1}, used to store frames following a damaged one.
- Frames delivered to the network layer in order.

Selective Repeat in Action



Why the Restriction of 2^{n-1}

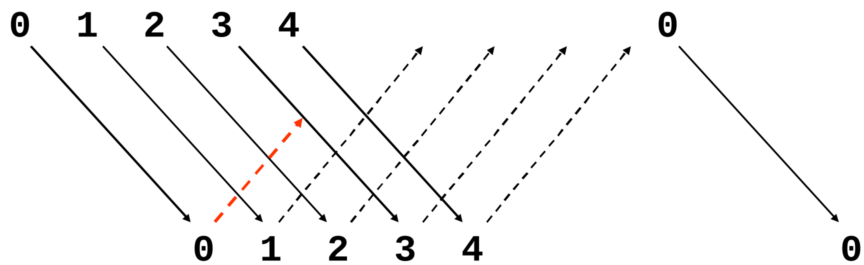
- Consider n=3. Let us allow 5 outstanding frames and see what happens.
- The receiver sees the following frames:

  0 1 2 3 4 0

- Again there are two readings for the second frame 0 and the receiver has no way to tell.

Scenario 1

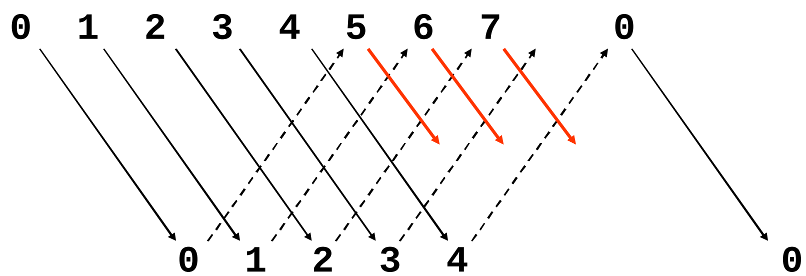
- Timeout and retransmit the 1st frame



- The second frame 0 is a duplicate.

CS 656 27

Scenario 2



- The second frame 0 is a new frame.
  - It shall be buffered in the receiver window to wait for the arrivals of 5 to 8.

CS 656 28

Discussion

- Stop-and-wait, go-back-n, and selective repeat achieve the same functions; differences are in performance.
- How does a slow receiver curb the sender ?
- How is reliability enforced ?
- How is correct ordering enforced ?

Performance Analysis

- *C*: channel capacity in bps
- *I*: interrupt and service time + propagation delay
  - Think of 2*I* as the round-trip time, excluding the transmission time of the frame itself
- *F*: number of bits per frame
- *T*: timeout interval

Stop-and-Wait without Errors

- At time ($F/C+2I$), the sender has processed the ACK.
- Total bandwidth during this period is

$$C(F/C+2I)=F+2CI$$

- So the utilization is

$$U=\frac{F}{F+2CI}$$

Stop-and-Wait with Errors

- A lost frame waits $F+CT$ bits of transmission capacity
- R: the average number of retransmissions per frame
- The total capacity used by a frame is

$$R(F+CT)+(F+2CI)$$

To determine R:

- L: the probability that a frame or its ACK is lost

- Probability that exactly k attempts are needed is $(1-L)L^{k-1}$
- Expected number of transmissions per frame is
$$\sum_{k=1}^{\infty} k(1-L)L^{k-1} = \frac{1}{L} = R+1$$
- We have, $R = L/(1-L)$
- So the utilization is
$$U = \frac{F}{\frac{L}{1-L}(F+CT)+(F+2CI)}$$

Sliding Window without Errors

- Sender can transmit *W*F/C* seconds before it must stop and wait
- ACK of the first frame arrives at time *F/C*+2*I*
- Large window (sender transmits continuously)
  - $W \times F/C \geq F/C + 2I$
  - $W \geq 1 + 2CI/F$
  - Of course, *U*=100%
- Small window (sender must stop and wait)
  - $W < 1 + 2CI/F$
  - Sender transmits *W* frames in time *F/C*+2*I*
  - Utilization is
$$U = \frac{WF}{F+2CI}$$

Selective Repeat with Errors

- To send W frames, the sender in average performs $(R+1)W=W(1-L)$ frame transmissions.
- $W \geq 1+2CI/F$
  - Transmission is still continuous, but extra frames must be sent to correct damaged ones
  - Thus, $U=(1-L)$
- $W < 1+2CI/F$
  - Efficiency drops by the same percentage as in the above case.
  - Thus,

$$U=(1-L)\frac{WF}{F+2CI}$$

The Meaning of *CI*

- Boundary between large and small windows has been $W=1+2CI/F$.
- What is CI ?
- Examples
  - Let $c=2*10^8$ m/sec be the signal propagation delay (approximately 2/3 of light speed)
  - 10Mbps over 1km, $CI = 10\times10^6\times(1000/c)\approx 50$ bits
  - 65 kbps over 3000 km, CI ≈ 960 bits
  - (satellite) 64 kbps with I=270 msec, CI ≈ 17000 bits

Discussion

- To determine the window size over a communication link, we must consider
  - The data rate of the link
  - The propagation delay over the link
- The second factor is especially important in WANs.
- For high-bandwidth and long-distance links, good performance can be achieved only with large windows.