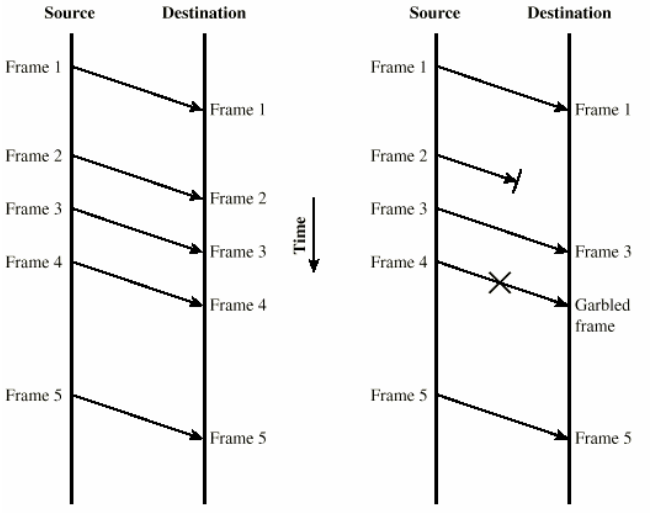
Data Link Control

CSE 3213
Fall 2007

Flow Control

- Ensuring the sending entity does not overwhelm the receiving entity
  - —Preventing buffer overflow
- Transmission time
  - —Time taken to emit all bits into medium
- Propagation time
  - —Time for a bit to traverse the link

Model of Frame Transmission



(a) Error-free transmission

(b) Transmission with losses and errors

3

Stop and Wait

- Source transmits frame
- Destination receives frame and replies with acknowledgement
- Source waits for ACK before sending next frame
- Destination can stop flow by not send ACK
- Works well for a few large frames

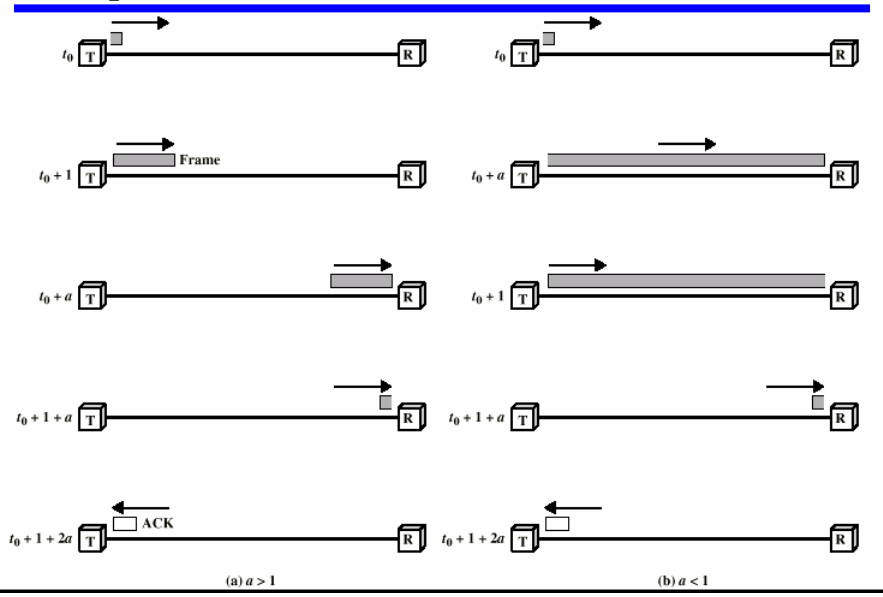
4

Fragmentation

- Large block of data may be split into small frames
  - —Limited buffer size
  - —Errors detected sooner (when whole frame received)
  - —On error, retransmission of smaller frames is needed
  - —Prevents one station occupying medium for long periods
- Stop and wait becomes inadequate

5

Stop and Wait Link Utilization

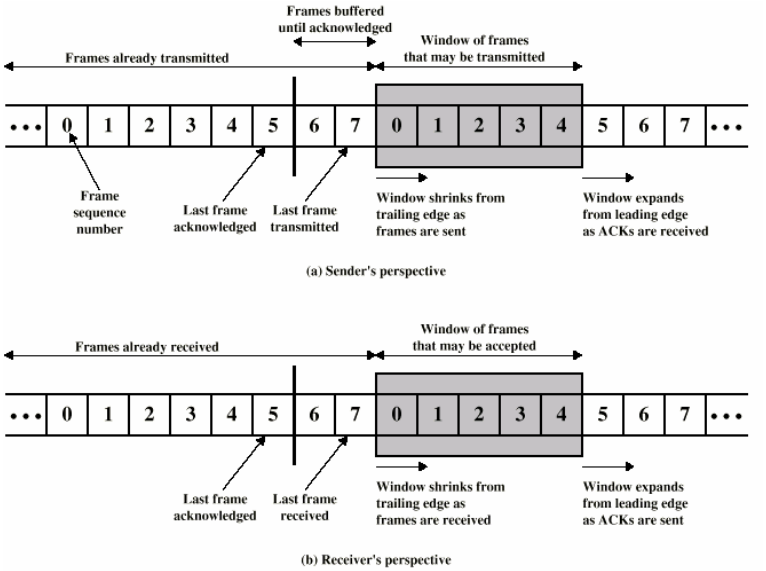


Sliding Windows Flow Control

- Allow multiple frames to be in transit
- Receiver has buffer of W frames long
- Transmitter can send up to W frames without ACK
- Each frame is numbered (for error control)
- ACK includes number of next frame expected
- Sequence number bounded by size of field (*k*)
  - —Frames are numbered modulo 2^k
- Max window size is $(2^k - 1)$

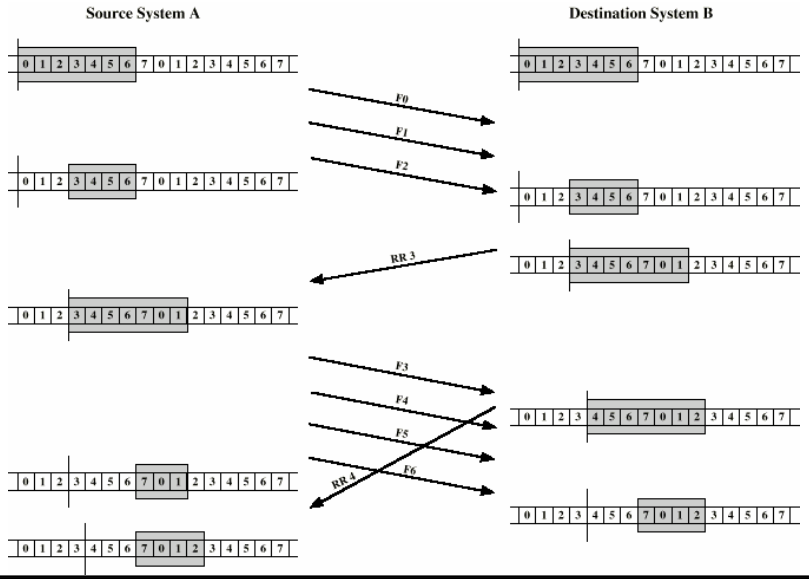
7

Sliding Window Diagram



8

Example Sliding Window



9

Sliding Window Enhancements

- RR*n* = received up to but not including frame *n*.
- Receiver can acknowledge frames without permitting further transmission (Receive Not Ready).
  - —RNR5 = received up to frame 4; temporarily stop transmission.
- Must send a normal acknowledge to resume.

10

Duplex Communications

- Two stations exchange data.
- Each maintains to windows: one for transmit, one for receive.
- Piggybacking: send both data and ACK in one frame, saving communication capacity.
- If having ACK but no data to send, send a separate ACK frame.
- If having data but no ACK to send, repeat the last sequence number sent (or use "ACK_valid" flag as in TCP).
  - —Duplicate ACKs are ignored.

11

Error Control

- Detection and correction of errors
- Lost frames
- Damaged frames
- Automatic repeat request
  - —Error detection
  - —Positive acknowledgment
  - —Retransmission after timeout
  - —Negative acknowledgement and retransmission

12

Automatic Repeat Request (ARQ)

- Stop and wait
- Go back N
- Selective reject (selective retransmission)

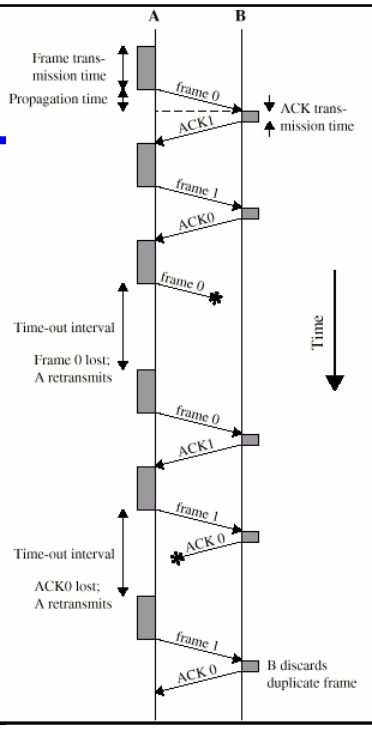
13

Stop and Wait

- Source transmits single frame
- Wait for ACK
- If received frame damaged, discard it
  - —Transmitter sets timer
  - —If no ACK within timeout, retransmit
- If ACK damaged,transmitter will not recognize it
  - —Transmitter will retransmit
  - —Receive gets two copies of the same frame
  - —Use ACK0 and ACK1 to discard duplicates

14

Stop and Wait - Diagram



Stop and Wait - Pros and Cons

- Simple
- Inefficient

16

Go Back N

- Based on sliding window
- If no error, ACK as usual with next frame expected
- Use window to control number of outstanding frames
- If error, reply with rejection
  - —Discard that frame and all future frames until error frame received correctly
  - —Transmitter must go back and retransmit that frame and all subsequent frames

17

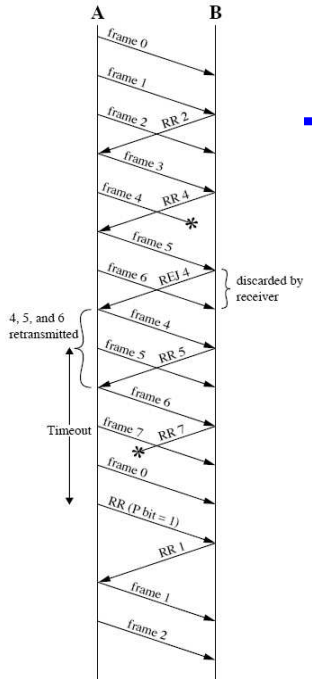
Lost/Damaged Frame: Case 1

Lost/damaged frame is not the last frame

- Receiver detects error in frame *i*, discard it (or frame *i* was simply lost in transit)
- Receiver receives frame (*i*+1) out of order
- Receiver sends rejection REJ_i
- Sender gets REJ_i
- Sender retransmits frame *i* and all subsequent frames

18

Go Back N - Diagram



19

Lost/Damaged Frame: Case 2

Frame *i* was lost/damaged and is the last frame of the message

- Receiver gets nothing and returns neither acknowledgement nor rejection
- Sender times out and sends a request for acknowledgment: RR(P=1)
  - —RR(P=0) is a regular ACK.
  - —Receiver responds to the request by sending RR_i
  - —Sender then retransmits frame *i*
- Alternative: Sender times out and retrx frame *i*.

20

Lost/Damaged RR

Case 1:

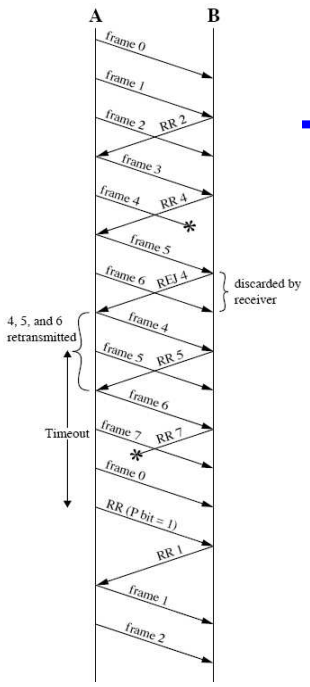
- Receiver gets frame *i* and sends acknowledgement (*i+1*) which is lost
- Acknowledgements are cumulative, so next acknowledgement (*i+n*) may arrive before sender times out on frame *i*

Case 2:

- If sender times out, it sends request for ACK with P bit set as before: RR(P=1)
- Receiver resends RR(*i+1*) (or possibly RR(*i+m*), *m*>1)
- The RR(P=1) packet is also timed in case it will be lost.
  - This can be repeated a number of times before a reset procedure is initiated

21

Go Back N - Diagram



22

Lost/Damaged REJ

- Receiver stops accepting data and is waiting for retransmitted frames.
- Sender times out and sends a request for acknowledgment: RR(P=1)
- Receiver responds to the request by resending REJ_i
- Sender retransmits frame *i* and all subsequent frames

23

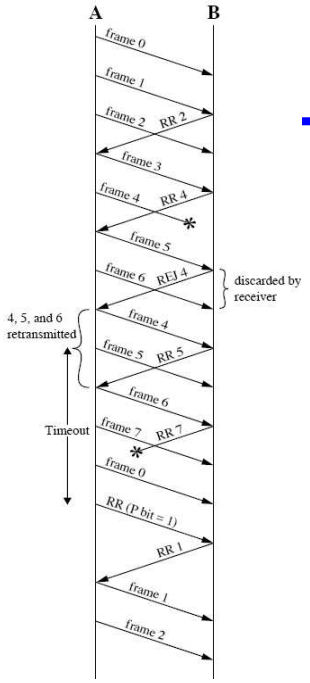
Go-Back-N Example

- Sequence numbers are 3 bit long, and start from 0.
- Window size is 6 frames.
- A sends data to B and B has no data to send.
- The sum of the propagation time and the transmission time for every data frame, and every RR/REJ is 1 time unit.
- B sends an acknowledgment (either positive or negative) every 2 time units.
- The time-out period for each data frame is 4 time units.

  Draw the flow diagram showing the sequence of interactions between A and B, assuming that
  - the 5th frame is damaged, then
  - acknowledgment RR 7 is lost

24

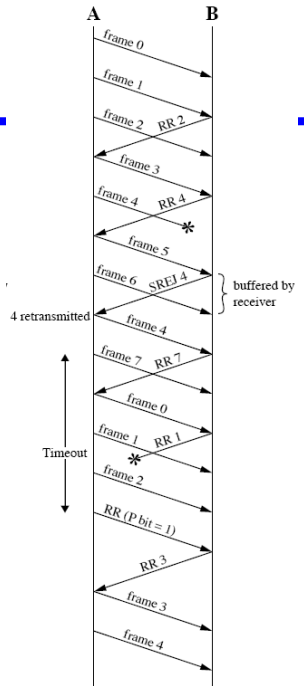
Go Back N - Diagram



Selective Reject ARQ

- Also called selective retransmission
- Only rejected frames are retransmitted
- Subsequent frames are accepted by the receiver and buffered
- Minimizes number of retransmissions
- Receiver must maintain large enough buffer and logic to insert repair frames into the proper sequence.
- More complex logic in transmitter to send frames out of sequence
  $\Rightarrow$ Go-back-N is more commonly used.
- Useful choice for satellite links due to long propagation delay
- Max window size (2^{k-1})
- In some implementations, the transmitter simply retrx a frame on time-out rather than sending RR(P=1).

Selective Reject - Diagram



Reading

- Sections 7.1 and 7.2, Stallings' book.
- Demo: http://gaia.ecs.csus.edu/~zhangd/oscal/selectiveReject/selectiveReject.html