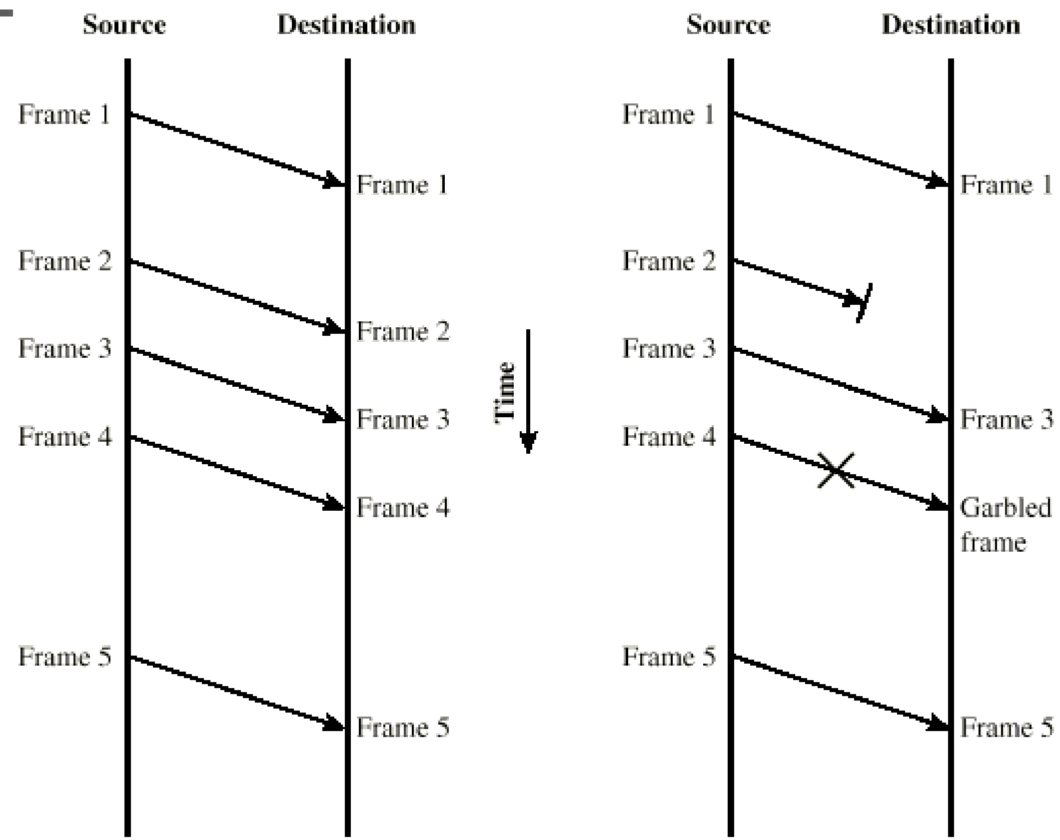
Flow Control, Reliable Transmission

Reference: William Stallings, "Data and Computer Communications", 6th edition, *Prentice Hall*, 2000.

What is Flow Control?

- Consider a link connecting sender and receiver nodes
  - If sender is faster and receiver is slower in processing frames then we say sender jams receiver. This leads to buffer overflow at the receiver
- Flow control is a technique used to ensure that a sender does not overwhelm (jam) the receiver
- Usually, receiver allocates a buffer of some size for frame transfer
- Flow control prevents buffer overflow at the receiver
- Flow control concerns with two end nodes: sender and receiver, SHOULD NOT be confused with congestion control
- Flow control is used at the connection level between end nodes at Layer-4 and link-level (could be optional) at Layer-2

Model of Frame Transmission



(a) Error-free transmission

(b) Transmission with losses and errors

Stop and Wait

- Source transmits a frame
- Destination receives a frame and replies with an acknowledgement (ACK)
- Source waits for ACK before sending the next frame
- Destination can stop flow by not sending ACK
- Simple implementation

Fragmentation

- Large block of data may be split into small frames for the following reasons
  - Limited buffer size
  - Errors detected sooner (when whole frame received)
  - On error, retransmission frames are small
  - Prevents one station occupying medium for long periods (broadcast network)
- Stop and wait becomes inadequate when a large number of frames are sent

Sliding Windows Flow Control

- Allow multiple frames to be in transit to improve link utilization
- Receiver has buffer of length W
- Transmitter can send up to W frames without waiting for ACK
- Each frame is numbered (called sequence numbers)
- ACK includes the sequence number of next frame expected
- ACK n (or) RR n (receiver ready) indicates that the receiver has received frames numbered up through n-1 and is ready to receive frame n onwards (cumulative acknowledgement)
- Sequence number bounded by size of field (k)
  - K-bit sequence number in frame header
  - Frames are numbered modulo 2^k (0 through 2^k)
  - For K=3, frames are numbered 0,1,2,3,,4,5,6,7,0,1,…6,7,0…

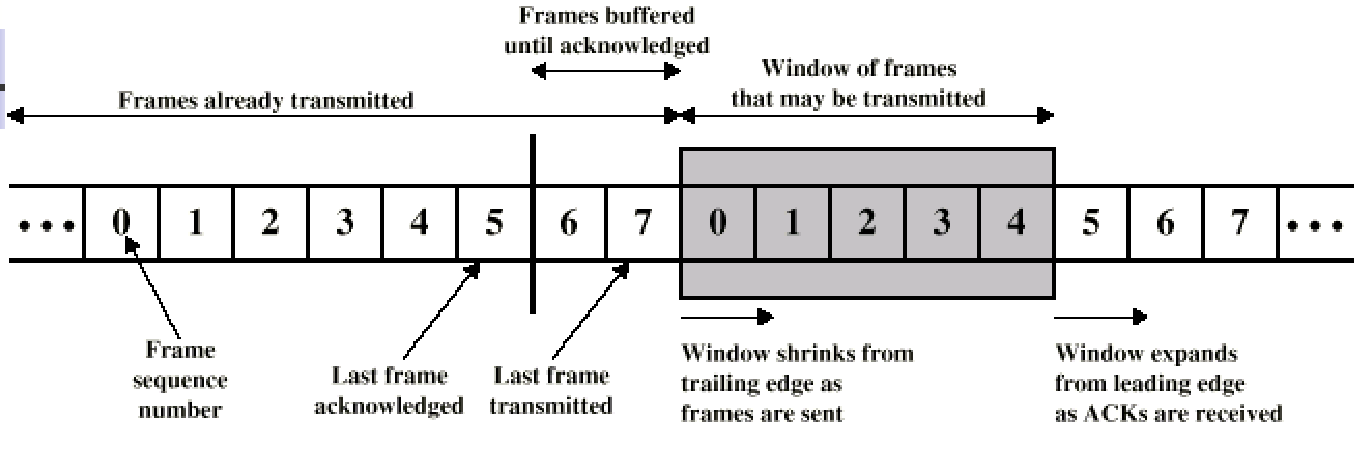
Sliding Window Flow Control (contd.)

- Sender
  - Keeps a window of (max) size W
  - Initially window size is W
  - Window size indicates the maximum number of frames that could be transmitted without need to wait for any acknowledgement
  - When a frame is sent, window shrinks by one
  - When an ACK is received, window expands by one
  - Keeps the frames in buffer that are not yet acknowledged (Why?)

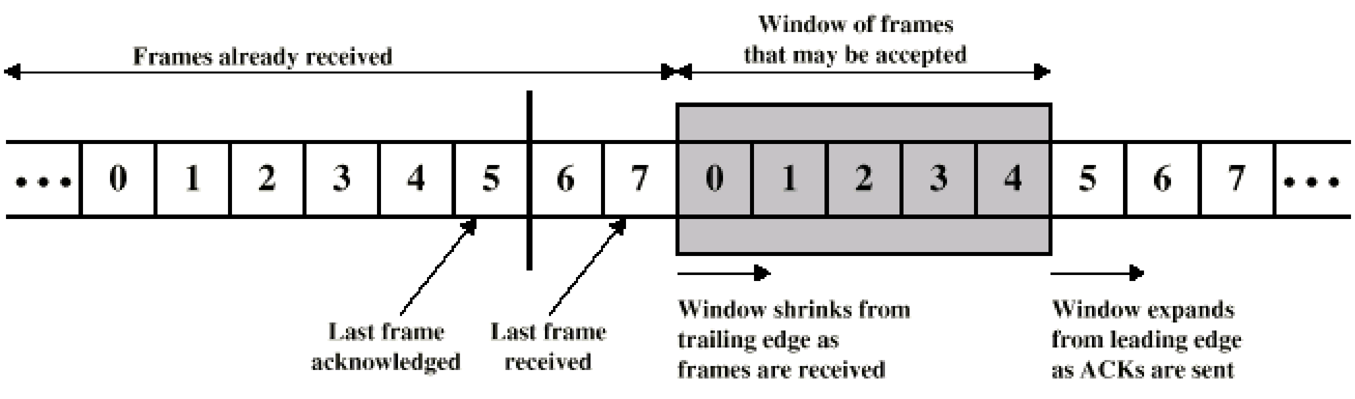
Sliding Window Flow Control (contd.)

- Receiver
  - Keeps a window of (max) size W
  - Initially window size is W
  - Window size indicates the maximum number of frames that could be received (without need to send any acknowledgement)
  - When a frame is received, window shrinks by one
  - When an ACK is sent window expands by one
  - Keeps in buffer, the frames received but ACKs have not yet been sent (Why?)

Sliding Window Illustration

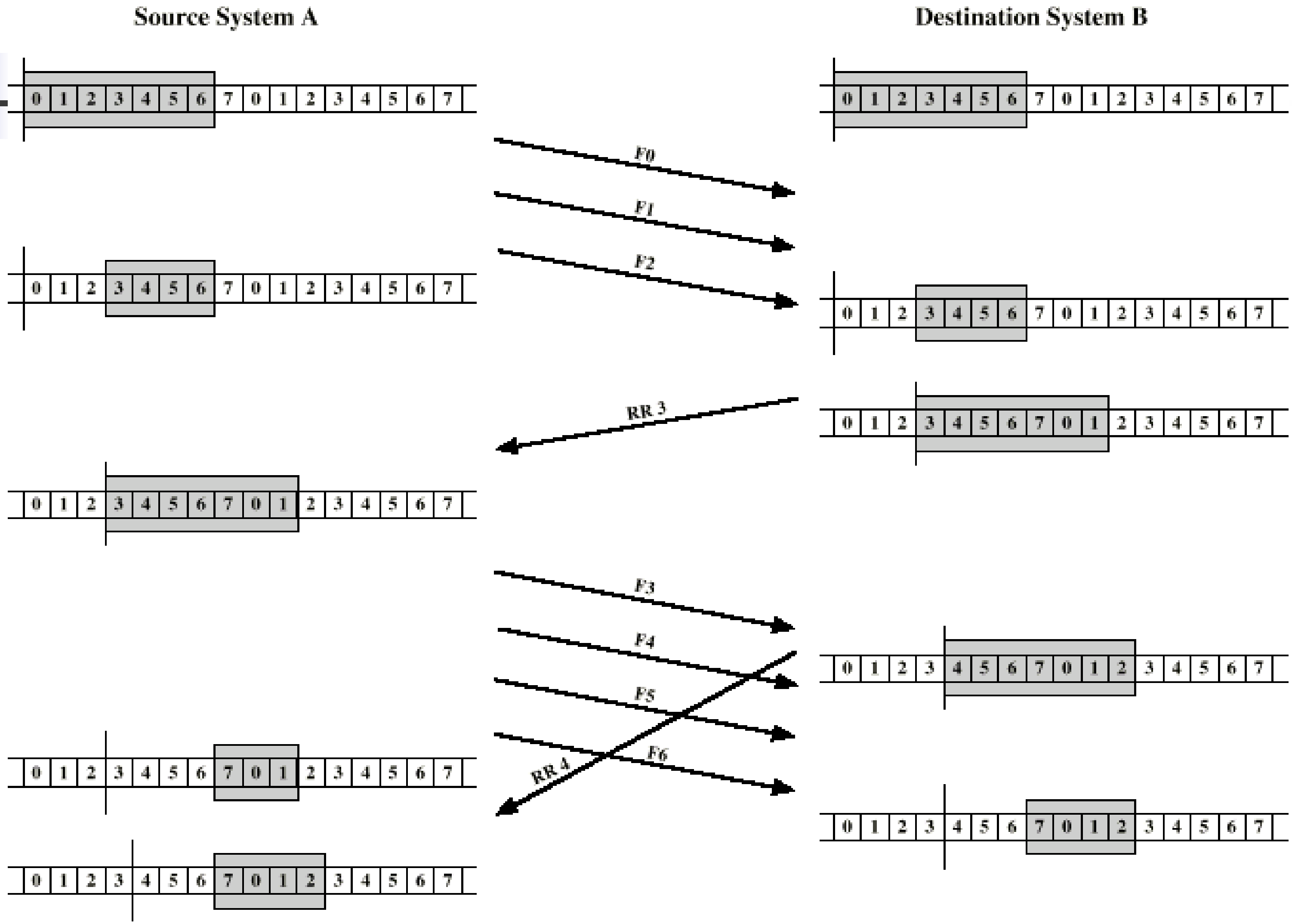


(a) Sender's perspective



(b) Receiver's perspective

Example Sliding Window



Sliding Window Enhancements

- Receiver can acknowledge frames without permitting further transmission (Receive Not Ready)
  - Must send a normal acknowledgement to resume
- If duplex, use piggybacking
  - Two-way communication between nodes
  - A node can be a sender for some data and receiver for some data
  - Receiver sends ACK (RR) as a part of data frame to the other node
    - ACK no is specified as a part of header
    - A sends its data frame F3 to B. B receives F3.
    - B sends its data frame F6 which carries ACK 4 (specifies 4 in ACK field of header)
  - Receiver has no data frame to send
    - use a separate acknowledgement frame
  - Receiver has data frame but no acknowledgement to send,
    - send last acknowledgement number again

Error Control and Reliable Transmission

- Detection and correction of errors
- Lost frames
  - Noise burst affects the frame to the extent that the frame is not recognizable by the receiver
- Damaged frames
  - Receiver can recognize the frame but some bits are in error
  - If the receiver is not sure about frame sequence number, then damaged frame requires the same treatment as lost frame
- Automatic repeat request (ARQ)
  - Error detection
  - Positive acknowledgment (ACK, RR)
  - Retransmission after timeout
  - Negative acknowledgement and retransmission (NACK, REJ (reject) SREJ (selective reject))

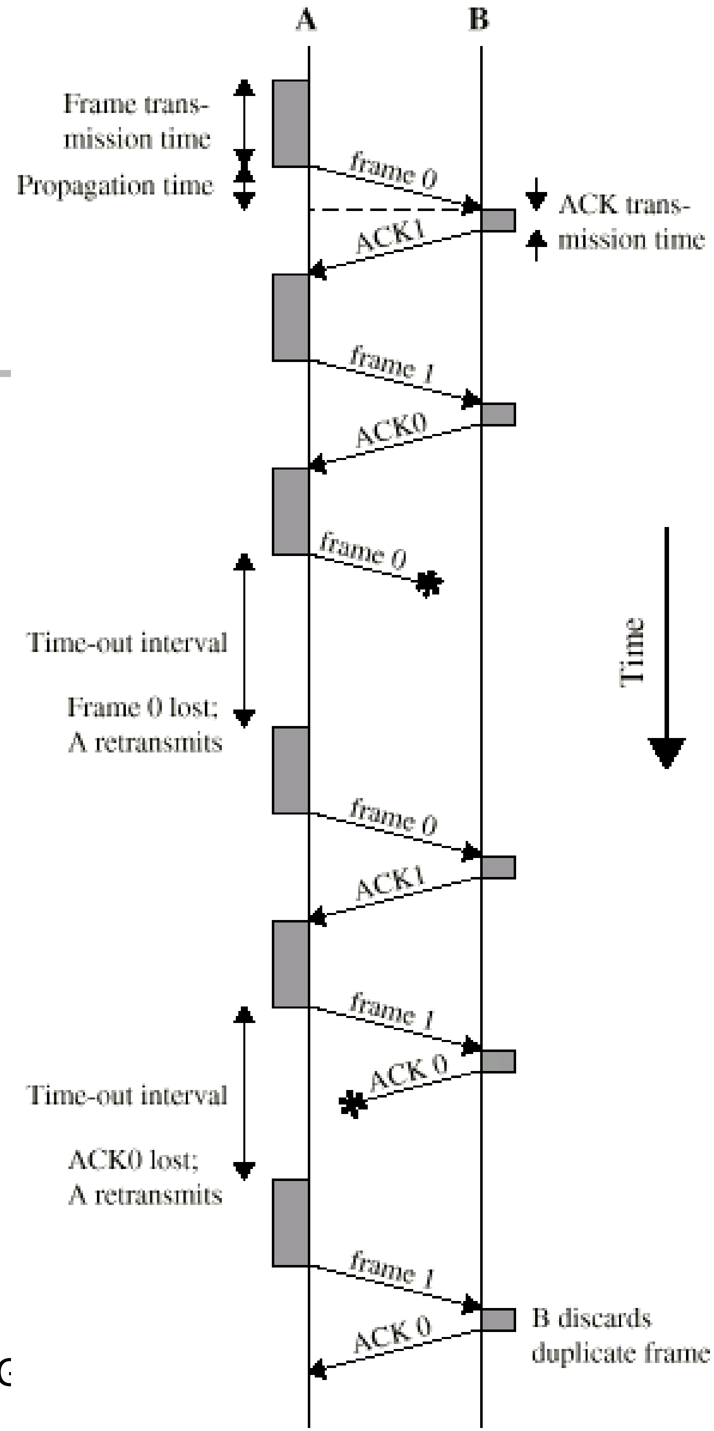
Automatic Repeat Request (ARQ)

- Stop-and-Wait ARQ
- Go-Back-N ARQ
  - based on sliding window control
  - Max window size is 2^k-1 where k is the seq-no field.
- Selective-Reject (selective retransmission) ARQ
  - based on sliding window control
  - Max window size is 2^{k-1} where k is the seq-no field.
- Go-Back-N is widely used when compared to Selective-Reject
- Typical window size
  - 7 for LANs (low speed link, short distance)
  - 127 for WANs (high speed link, long distance), WHY?

Stop and Wait ARQ

- Sender transmits a frame
- When a frame arrives, receiver sends ACK
- Sender waits for ACK before transmitting the next frame
- If received frame damaged/lost, receiver discards it
  - Transmitter uses TIMEOUT mechanism
  - If no ACK within timeout, retransmit
- If ACK damaged/lost, transmitter will not recognize it
  - Transmitter will retransmit
  - Receiver gets two copies of the same frame
  - Alternate between ACK0 and ACK1 to solve the above problem

Stop and Wait ARQ illustration



Stop and Wait - Pros and Cons

- Simple implementation
- Poor link utilization
- Link utilization (U) is defined as the ratio between the actual time spent for transmitting the given data frames and the total time the link is engaged until the completion of the transfer of the frames
  - Numerically equal to throughput rate

Go Back N ARQ

- Based on sliding window
- If no error, receiver sends ACK (or RR) as usual with next frame expected
- Use window to control the number of outstanding frames
- If error,
  - Receiver discards that frame and all future frames until error frame received correctly
  - Sender goes back and retransmits that frame and all subsequent frames

Go Back N - Damaged Frame

- Receiver detects error in frame *i*
- Receiver sends REJ *i*
  - discards any new frame received until frame *i* is correctly received
- Sender gets REJ-*i*
- Sender retransmits frame *i* and subsequent frames

Go Back N - Lost Frame (case 1)

- Frame *i* lost
- Sender sends *i+1*
- Receiver gets frame *i+1* out of sequence
- Receiver sends REJ-*i*
- Sender goes back to frame *i* and retransmits frame *i* and subsequent frames

Go Back N - Lost Frame (case 2)

- Frame *i* lost and sender does not send additional frames
- Receiver gets nothing and returns neither ACK nor REJ
- Sender times out and sends RR (command) frame with P (probe) bit set to 1
- Receiver interprets this RR as command which it acknowledges with the number of the next frame it expects (RR *i*)
- Sender then retransmits frame *i* and subsequent frames

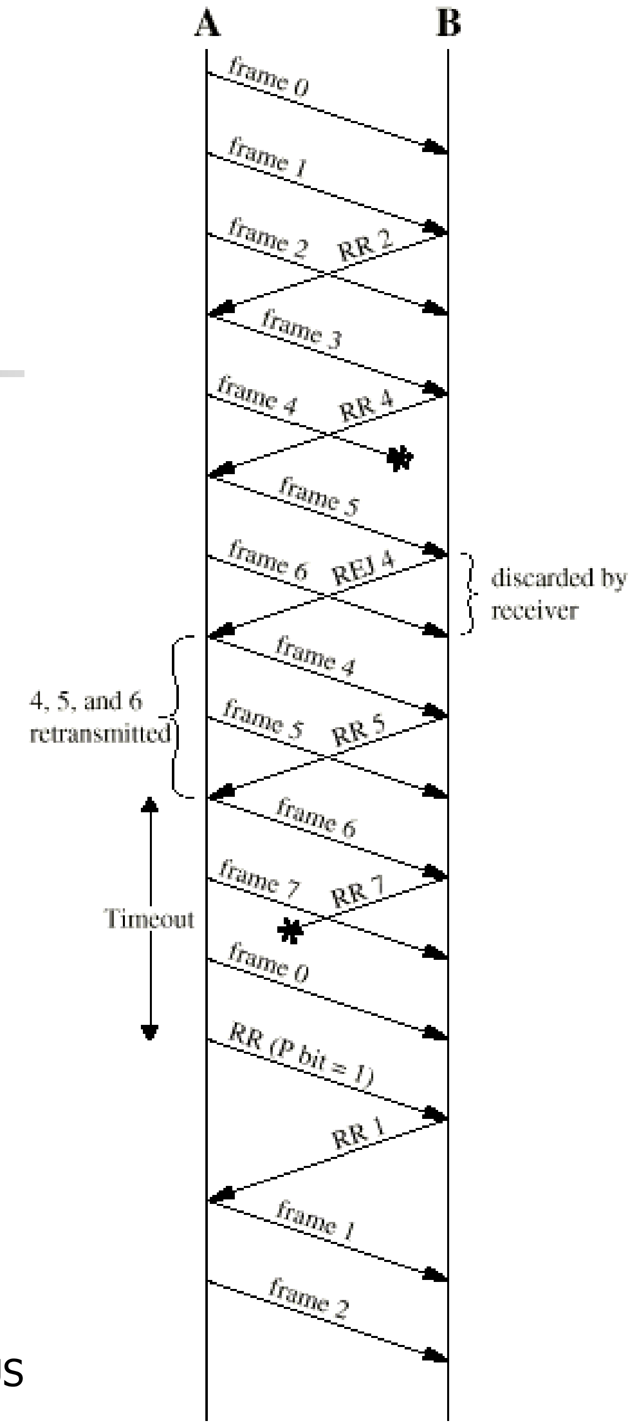
Go Back N – Damaged/Lost Acknowledgement

- Receiver gets frame *i* and sends RR (*i+1*) which is lost
- Acknowledgements are cumulative, so next acknowledgement (*i+n, n>1*) may arrive before sender times out on frame *i* (case 1)
- If sender times out, it sends RR command frame with P bit set as before (case 2)
  - This can be repeated a MAX number of times before a reset procedure is initiated
- Damaged REJ – similar to LOST frame (case 2)

Go Back N Pros and Cons

- Better link utilization than stop-and-wait
- less complex implementation (compared to selective reject ARQ)
- Good choice when errors are rare
- Retransmission of error-free frames reduces link utilization

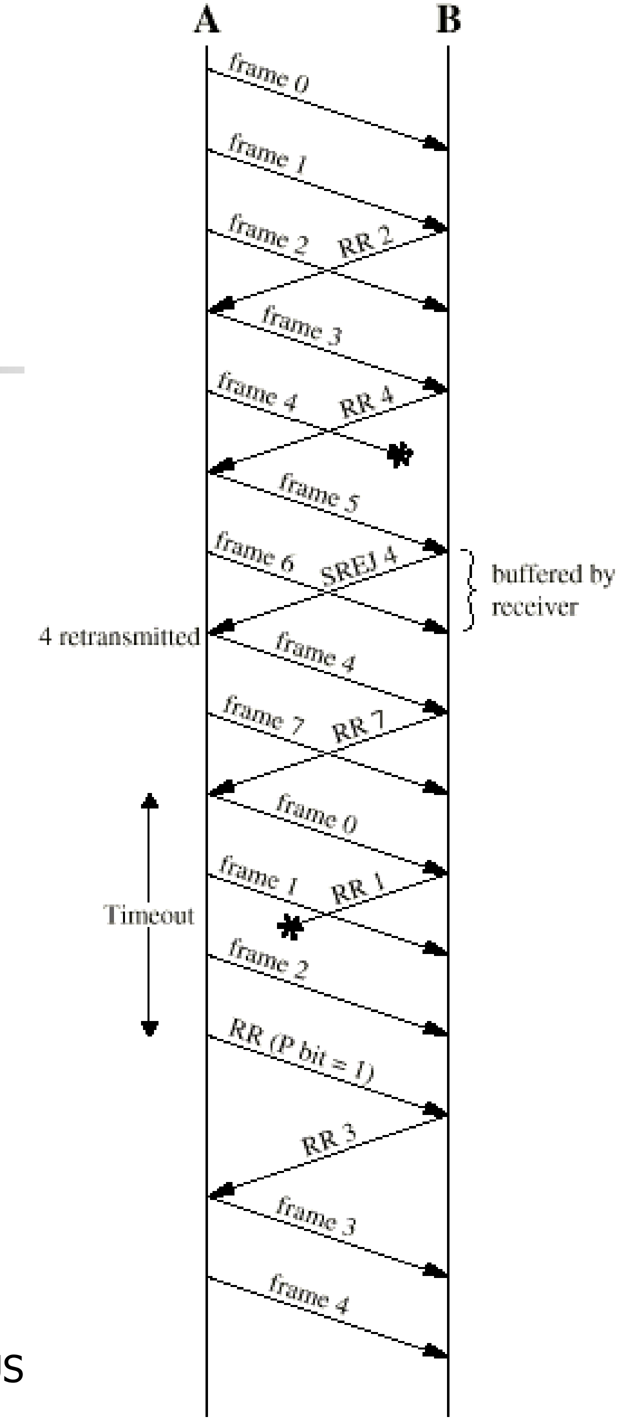
Go Back N Illustration



Selective Reject ARQ

- Also called selective retransmission
- Only rejected frames are retransmitted
- Subsequent frames are accepted by the receiver and buffered
- Need to arrange frames in order before delivering to the higher layer
- Minimizes retransmission, better link utilization than stop-and-wait and Go-back-N
- Unlike Go-back-N, Receiver must maintain large enough buffer
- More complex implementation

Selective-Reject Illustration



Performance Study

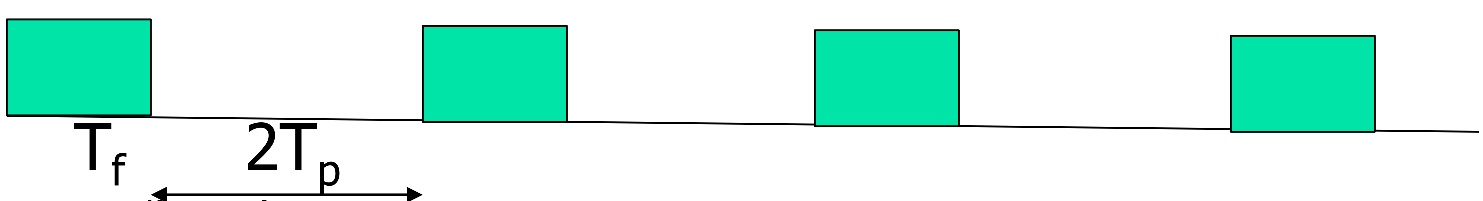
- Derive expression for Link Utilization
- The following assumptions and notation are made
- Frame transmission time T_f
- Link propagation time T_p
- Total time that the link is engaged for one frame transmission denoted by T_{total}
- Transmission time of ACK frame is negligible
- Frame error probability denoted by P_f
- Errors in ACK frames can be ignored
- Errors in retransmitted frames other than the frame initially in error can be ignored(in case of Go-back-N)

Parameter a

- a = link propagation time (Tp)/frame transmission time (Tf)
- a = Number of frames that can be held on a link (how?)
  - Related to D×B product of the link where D is Tp

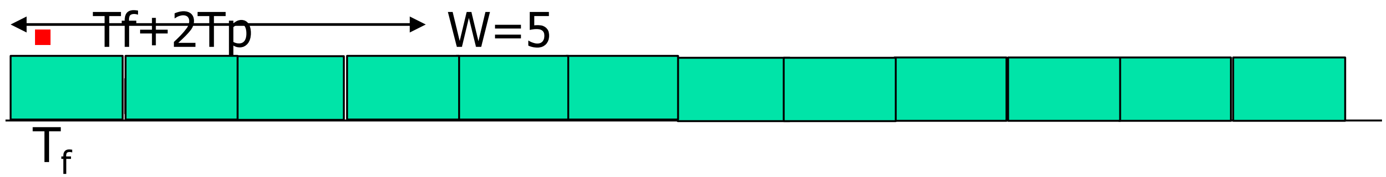
Performance of Stop-and-Wait (error-free and error cases)

- $P_f = 0$ (error-free case)
  - One frame is transmitted for every T_f+2T_p time
  - $T_{total} = T_f+2T_p$
  - Link utilization $U = T_f / T_{total} = 1 / (1+2a)$
- Pf > 0 (error case)
  - Let N_r be the expected number of times a frame is transmitted
  - N_r can be shown to be equal to $1/(1-P_f)$ (How?)
  - $U = T_f /(N_r \times T_{total}) = (1-P_f) / (1+2a)$

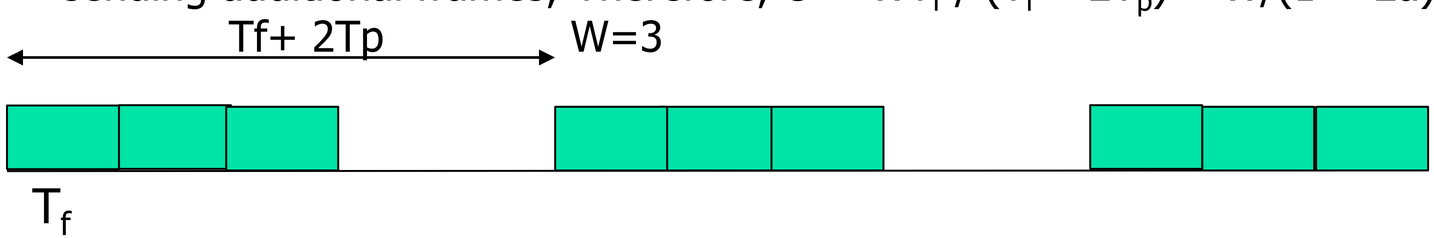


Sliding Window Protocol-Error-free Communication

- ACK frame is received by the sender after $T_f + 2T_p$ from the time at which frame transmission starts
- Window size $WT_f \geq T_f + 2T_p$ ($W \geq 1 + 2a$)
  - Sender receives ACK frame before the window is exhausted; it can send frames continuously; Therefore, U = 1



- Window size $WT_f < T_f + 2T_p$ ($W < 1 + 2a$)
  - Sender sends W frames and waits until $T_f + 2T_p$ to receive an ACK before sending additional frames; Therefore, $U = WT_f / (T_f + 2T_p) = W/(1 + 2a)$



Sliding Window Protocol ARQ (with errors)

- Selective Reject
  - $U = (1-P_f), \; W \geq 1 + 2a$
  - $U = [W(1-P_f)] / (1 + 2a), \; W < 1 + 2a$

- Go back N
  - $U = (1-P_f) / (1 + 2aP_f), \; W \geq 1 + 2a$
  - $U = [W(1-P_f)] / [(1 + 2a)(1-P_f+WP_f)], \; W < 1 + 2a$

- For proof, refer "Data and Computer Communications" by William Stallings and/or other reference books