King Fahd University of Petroleum & Minerals
Computer Engineering Dept

COE 342 – Data and Computer Communications

Term 061

Dr. Ashraf S. Hasan Mahmoud

Rm 22-148-3

Ext. 1724

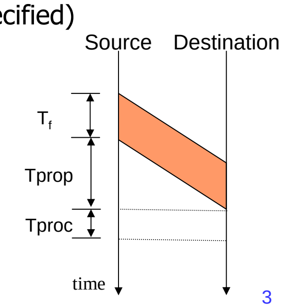
Email: ashraf@ccse.kfupm.edu.sa

What is Data Link Control

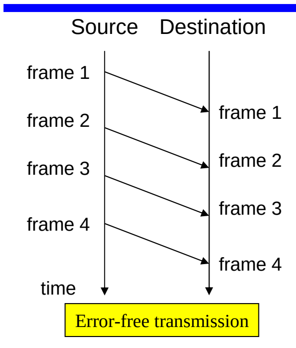
- The logic or procedures used to convert the raw stream of bits provided by the physical layer into a "*reliable*" connection
- Requirements and Objectives:
  - Frame synchronization
  - Flow control
  - Error control
  - Addressing
  - Multiplexing data and control on connection
  - Link management

Flow Control

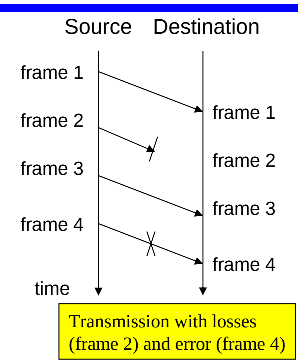
- A scheme to ensure that transmitter does not overwhelm receiver with data
- Transmission of one frame:
  - T_f: time to transmit frame
  - Tprop: time for signal to propagate
  - Tproc: time for destination to process received frame – small delay (usually ignored if not specified)
- Tproc may be ignored if not specified



Flow Control (2)



Models of Frame Transmission



- The destination has a limited buffer space. How will the source know that destination is ready to receive the next frame?
- In case of errors or lost frame, the source need to retransmit frames – i.e. a copy of transmitted frames must be kept. How will the source know when to discard copies of old frames?
- Etc.

Stop-and-Wait Protocol

- Protocol:
  - Source transmits a frame
  - After the destination receives frame, it sends ACK
  - Source, upon the receipt of ACK, can now send the next frame
- Destination can stop source by withholding the ACK
- Simple
- Animation for Stop-and-Wait
- NOTE: ONLY one frame can be in transit at any time

Stop-and-Wait Protocol: Efficiency

- After every frame, source must wait till acknowledgment ➔ Hence link propagation time is significant
- Total time to for one frame:

  T_total = Tf + 2Tprop + Tproc + Tack

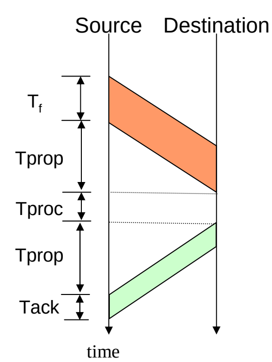
  if we ignore Tproc and Tack (usually very small)

  T_total = Tf + 2Tprop
- Link utilization, U is equal to

  U = Tf/ (T_total), or

  = 1 / (1+2(Tprop/Tf)) = 1 / (1 + 2 a)

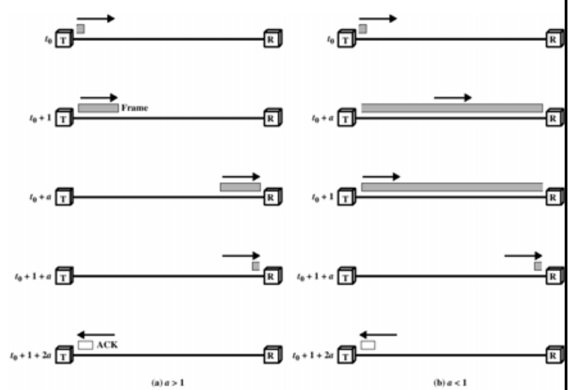
  where a = Tprop/Tf = length of link in bits
- If a < 1 (i.e. Tf > Tprop – when 1st transmitted bit reaches destination, source will still be transmitting ➔ U is close 100%
- If a > 1 (i.e. Tf < Tprop – frame transmission is completed before 1st bit reaches destination ➔U is low
- See figure 7.2



Stop-and-Wait Protocol: Efficiency (2)

- Remember: a = Tprop/Tf = length of link in bits

- If a < 1 (i.e. Tf > Tprop – when 1st transmitted bit reaches destination, source will still be transmitting ➔ U is close 100%
- If a > 1 (i.e. Tf < Tprop – frame transmission is completed before 1st bit reaches destination ➔U is low
- Stop-and-Wait is efficient for links where a << 1 (long frames compared to propagation time)



Sliding Window Protocol

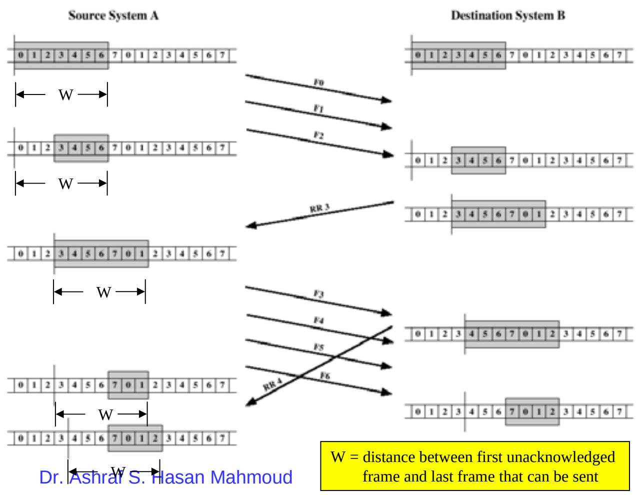
- Stop-and-Wait can be very inefficient when a > 1
- Protocol:
  - Assumes full duplex line
  - Source A and Destination B have buffers each of size W frames
  - For k-bit sequence numbers:
    - Frames are numbered: 0, 1, 2, …, 2^k-1, 0, 1, … (modulo 2^k)
    - ACKs (RRs) are numbered: 0, 1, 2, …, 2^k-1, 0, 1, … (modulo 2^k)
  - A is allowed to transmit up to W frames without waiting for an ACK
  - B can receive up to W consecutive frames
  - ACK J (or RR J), where 0<=J<= 2^k-1, sent by B means B is have received frames up to frame J-1 and is ready to receive frame J
  - B can also send RNR J: B have received all frames up to J-1 and is not ready to receive any more
- Window size, W can be less or equal to 2^k-1

Sliding Window Protocol (2)

- Example of Sliding-Window-Protocol: k = 3 bits, W = 7

Observations:

- A may tx W = 7 frames (F0, F1, …, F6)
- After F0, F1, & F2 are tx-ed, window is shrunk (i.e. can not transmit except F3, F4, …, F6)
- When B sends RR3, A knows F0, F1 & F2 have been received and B is ready to receive F3
- Window is advanced to cover 7 frames (starting with F3 up to F1)
- A sends F3, F4, F5, & F6
- B responds with RR4 when F3 is received – A advances the window by one position to include F2



Sliding Window Protocol - Piggybacking

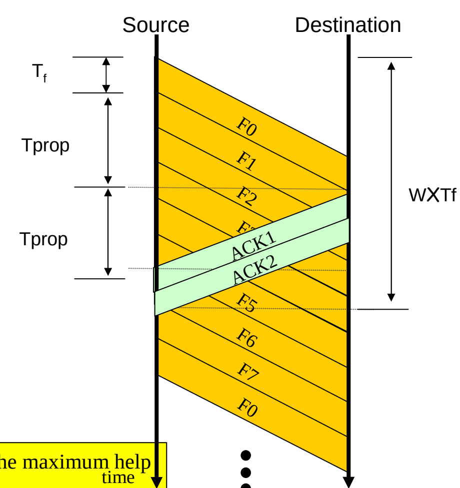
- When using sliding window protocol in full duplex connections:
  - Node A maintains its own transmit window
  - Node B maintains its own transmit window
  - A frame contains: data field + ACK field
  - There is a sequence number for the data field, and a sequence number for the ACK field

Sliding Window Protocol - Efficiency

- Again we can distinguish two cases:
- Case 1: W ≥ 2a + 1
- Case 2: W < 2a + 1

Sliding Window Protocol - Efficiency - Case 1

- Assume k=3, W = 7
  (ignoring Tack)
- Source can continuously keep transmitting!!
  - Because the ACK can arrive to source before the window is completed
- Utilization = 100%

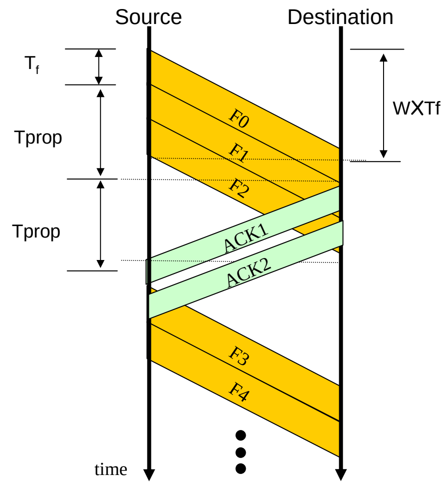


Sending RR1 or ACK1 as soon as F0 is received is the maximum help the destination can do to increase utilization

Sliding Window Protocol - Efficiency - Case 2

- Assume k = 3, W = 3 (ignoring Tack)
- Source can NOT continuously keep transmitting!!
  - Because the ACK can NOT arrive to source before the window is completed
- Utilization =
$$\frac{W \times T_f}{T_f + 2 \times T_{prop}}$$

$$= \frac{W}{1 + 2a}$$



Sliding Window Protocol - Efficiency

- Refer to Appendix A
- When window size is W (for error free), link utilization, U, is given by

$$U = \begin{cases} 1 & W \geq (2a+1) \\ \dfrac{W}{2a+1} & W < (2a+1) \end{cases}$$

where a = Tprop/Tf or length of link in bits

- Sliding window protocol can achieve 100% utilization if W >= (2a + 1)

Sliding Window Protocol

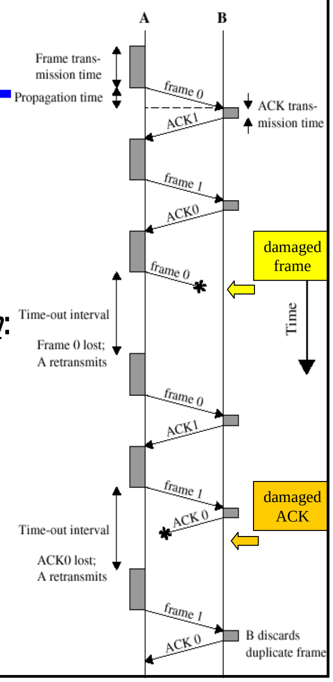
- Animation for Sliding Window protocol
- Sliding Window Protocol Simulation (http://www.cs.stir.ac.uk/~kjt/software/comms/jasper/SWP3.html)

Error Control

- Types of Errors:
  - Lost frame
  - Damaged frame
- Error control Techniques (Automatic Repeat Request - ARQ):
  - Error detection – discussed previously
  - +ve ACK
  - Retransmission after timeout
  - -ve ACK and retransmission
- ARQ Procedures: convert an unreliable data link into a reliable one.
  - Stop-and-wait
  - Go-back-N
  - Selective-reject

Stop-and-Wait ARQ

- Based on the stop-and-wait control flow procedure - Stop-and-Wait Protocol slide
- Two types of errors:
  1. Frame lost or damaged – *Solution*: timeout timer
  2. Damaged or lost ACK – The timeout timer solves this problem



Go-Back-N ARQ

- Based on the sliding-window flow control procedure - Sliding Window Protocol slide
- Three types of errors:
  1. i^{th} frame damaged:
     a. If A send subsequent frames (i+1, i+2, …), B responds with REJ i ➔ A must retransmit i^{th} frame and ***all subsequent frames***
     b. If A does not send subsequent frames and B does not respond with RR or REJ (since frame was damaged) ➔ timeout timer at A expires – send a POLL signal to B; B sends an RR i, i.e. it expect the i^{th} frame – A sends the i^{th} frame again

     Check for status of B before resending the frame
  2. Damaged RR (B receives i^{th} frame and sends RR i+1 which is lost or damaged):
     a. Since ACKs are cumulative – A may receive a subsequent RR j (j >i+1) before A times out
     b. If A times out, it sends a POLL signal to B – if B fails to respond (i.e. down) or its response is damaged subsequent POLLs are sent; procedure repeated certain number of time before link reset
  3. Damaged REJ – same as 1.b

Selective-Reject ARQ

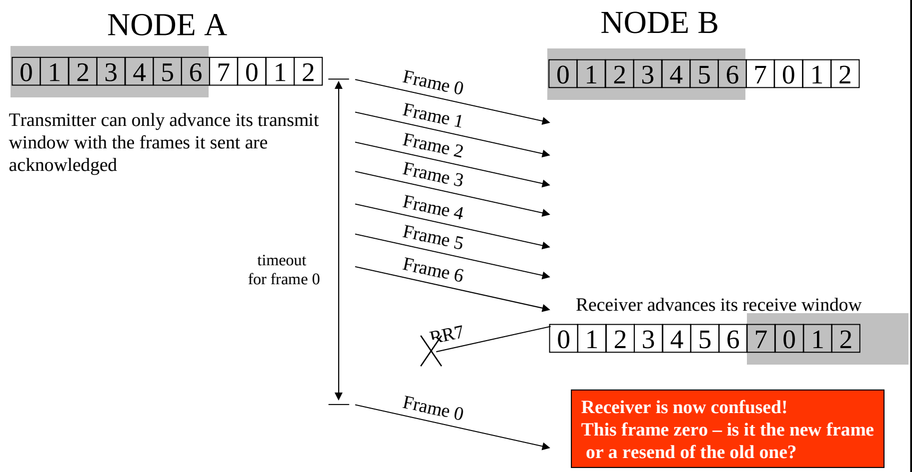
- In contrast to Go-Back-N, the only frames retransmitted are those that receive –ve ACK (called SREJ) or those that time out
- More efficient:
  - Rx-er must have large enough buffer to save *post*-SREJ frames
  - Buffer manipulation – re-insertion of out-of-order frames

Window Size for Selective-Reject ARQ – Why?

- Window size: should less or equal to half range of sequence numbers
  - For n-bit sequence numbers, Window size is $\leq 2^{n-1}$ (remember sequence numbers range from 0,1, …, 2^n-1)
- Why? See next example

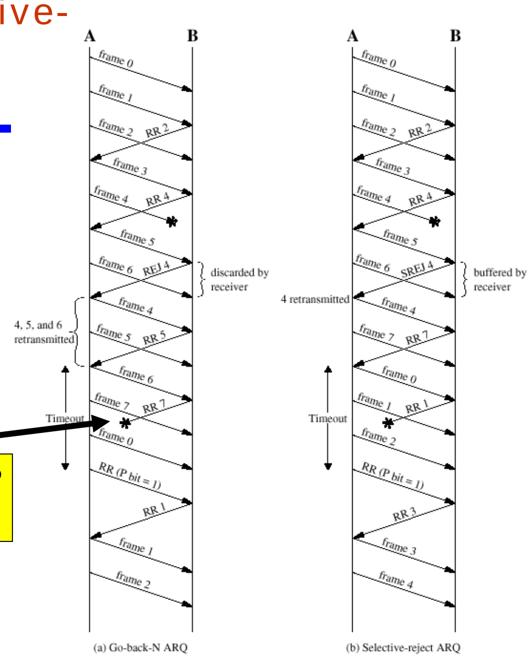
Window Size for Selective-Reject ARQ – Why? (2)

- Example: Consider 3-bit sequence number and window size of 7



Go-Back-N/Selective-Reject ARQ Examples

- With Go-back-N frames 4,5 and 6 are retransmitted
- With Selective-Reject only frame 4 is retransmitted



Did this lost RR7 affect flow?
How did the link recover?

Example: Problem 7-9

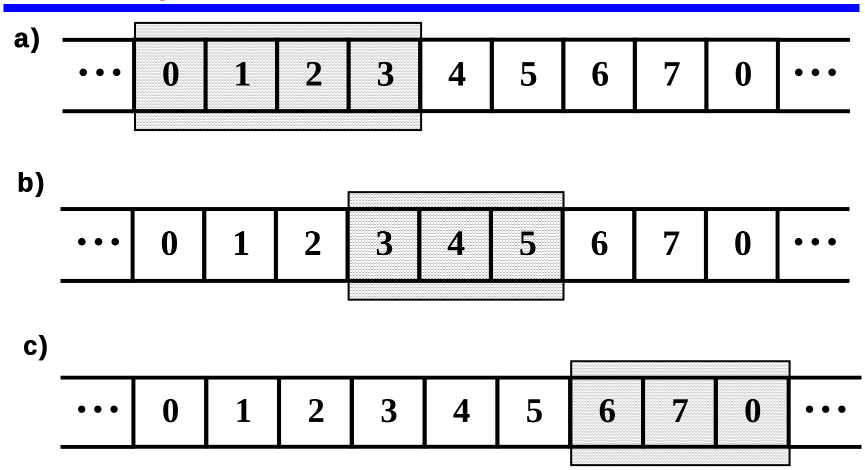
7-9: Two neighboring nodes A and B use a sliding-window protocol with a 3-bit sequence numbers. As the ARQ mechanism, go-back-N is used with a window size of 4. Assuming A is transmitting and B is receiving, show the window positions for the following succession of events:

a) Before A sends any frames

b) After A sends frame 0, 1, 2 and B acknowledges 0, 1 and the ACKs are received by A

c) After A sends frames 3, 4, and 5 and B acknowledges 4 and the ACK is received by A

Example: Problem 7-9 - Solution



Example Webpage

http://www.site.uottawa.ca/~elsaddik/abedweb/applets/Applets/Sliding_Window/sliding_window.html

This is an example of the end product of our project.