King Fahd University of Petroleum & Minerals
Computer Engineering Dept

COE 342 – Data and Computer Communications

Term 032

Dr. Ashraf S. Hasan Mahmoud

Rm 22-148-3

Ext. 1724

Email: ashraf@ccse.kfupm.edu.sa

Lecture Contents

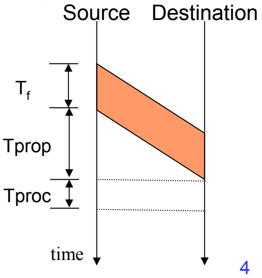
1. Flow Control
   a. Stop-and-Wait flow control
   b. Sliding-Window flow control
2. Error Detection (Parity Check, CRC)
3. Error Control
   a. Stop-and-Wait ARQ
   b. Go-Back-N ARQ
   c. Selective-Reject ARQ
4. High-Level Data Link (HDLC)
5. Other Data Link Control Protocols

What is Data Link Control

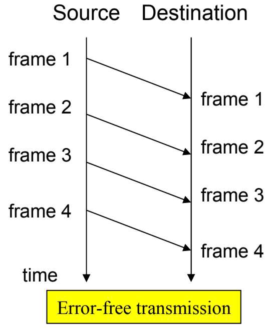
- The logic or procedures used to convert the raw stream of bits provided by the physical layer into a "*reliable*" connection
- Requirements and Objectives:
  - Frame synchronization
  - Flow control
  - Error control
  - Addressing
  - Multiplexing data and control on connection
  - Link management

Flow Control

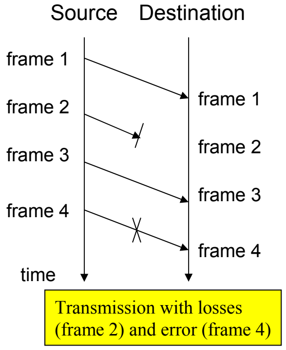
- A scheme to ensure that transmitter does not overwhelm receiver with data
- Transmission of one frame:
  - T_f: time to transmit frame
  - Tprop: time for signal to propagate
  - Tproc: time for destination to process received frame – small delay (usually ignored if not specified)
- Tproc may be ignored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

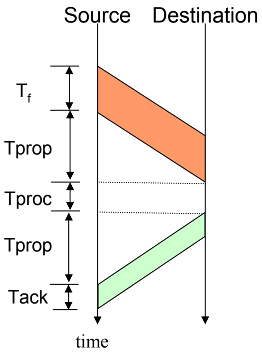
  if we ignore Tproc and Tack (usually very small)

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

  U = Tf/ (T_total), or

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

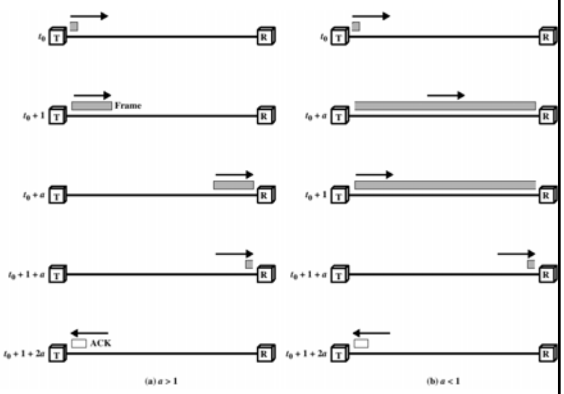
  where a = Tprop/Tf = length of link in bits
- If a < 1 (i.e. Tf > Tprop – when 1st transmitted bit reaches destination, source will still be transmitting ➔ U is close 100%
- If a > 1 (i.e. Tf < Tprop – frame transmission is completed before 1st bit reaches destination ➔U is low
- See figure 7.2



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)



(a) $a > 1$ (b) $a < 1$

Sliding Window Protocol

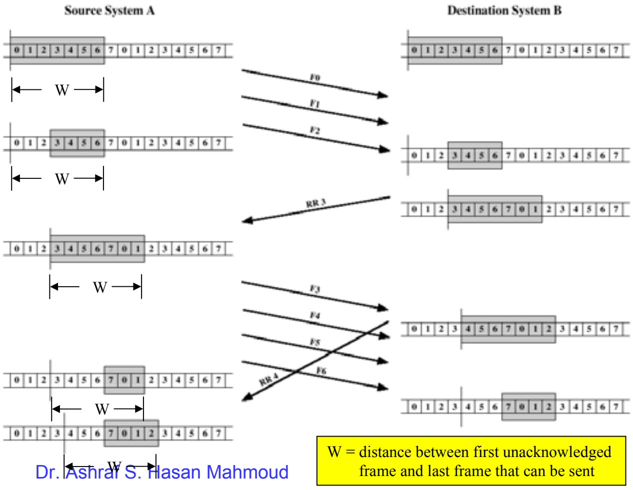
- Stop-and-Wait can be very inefficient when a > 1
- Protocol:
  - Assumes full duplex line
  - Source A and Destination B have buffers each of size W frames
  - For k-bit sequence numbers:
    - Frames are numbered: 0, 1, 2, …, 2^k-1, 0, 1, … (modulo 2^k)
    - ACKs (RRs) are numbered: 0, 1, 2, …, 2^k-1, 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



W = distance between first unacknowledged frame and last frame that can be sent

Sliding Window Protocol - Piggybacking

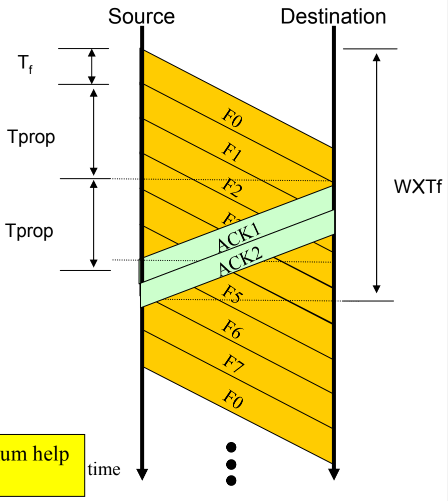
- When using sliding window protocol in full duplex connections:
  - Node A maintains its own transmit window
  - Node B maintains its own transmit window
  - A frame contains: data field + ACK field
  - There is a sequence number for the data field, and a sequence number for the ACK field

Sliding Window Protocol - Efficiency

- Again we can distinguish two cases:
- Case 1: $W \geq 2a + 1$
- 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%



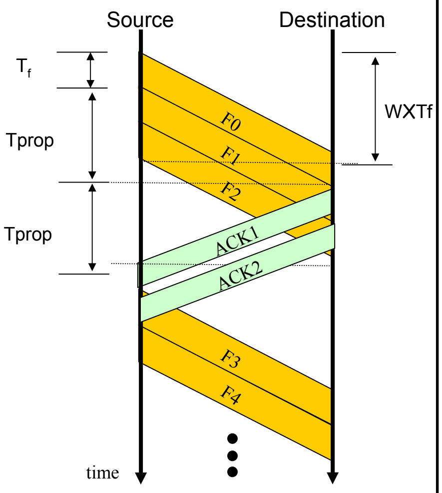
Sending ACK0 as soon as F0 is received is the maximum help the destination can do to increase utilization

---

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 = $\dfrac{W \times Tf}{Tf + 2 \times Tprop}$

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



Sliding Window Protocol - Efficiency

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

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

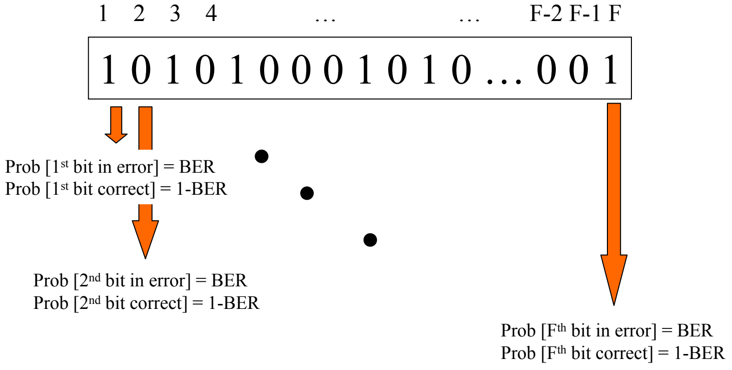
where a = Tprop/Tf or length of link in bits

- 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 Detection



1 2 3 4 … … F-2 F-1 F

1 0 1 0 1 0 0 0 1 0 1 0 … 0 0 1

Prob [1st bit in error] = BER
Prob [1st bit correct] = 1-BER

Prob [2nd bit in error] = BER
Prob [2nd bit correct] = 1-BER

Prob [Fth bit in error] = BER
Prob [Fth bit correct] = 1-BER

Prob [k bits in error in frame] = $\binom{F}{k}(BER)^k(1-BER)^{F-k}$

Error Detection – cont'd

- Hence, for a frame of F bits,

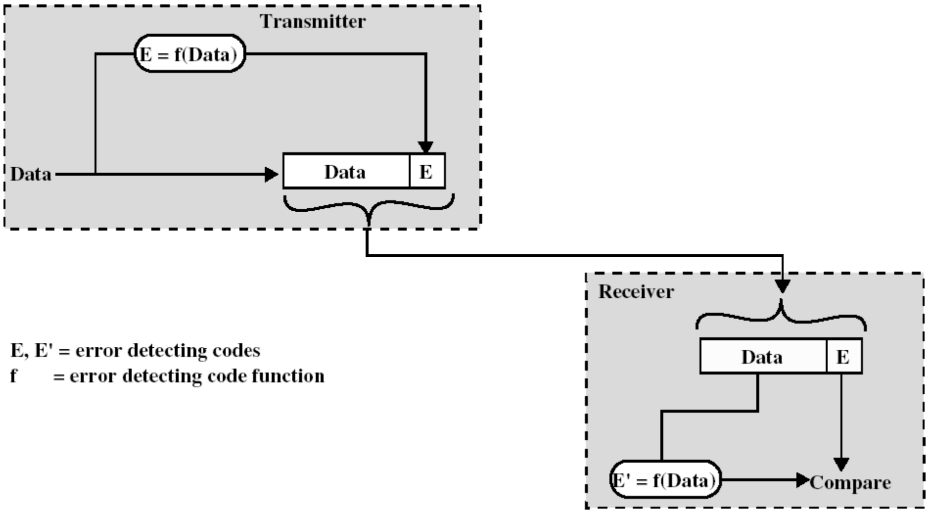
Prob [frame is correct] = Prob [0 bits in error]
= $(1-BER)^F$

Prob [frame is erroneous] = Prob[1 OR MORE bits in error]
= 1 – Prob[0 bits in error]
= 1 - $(1-BER)^F$

Or

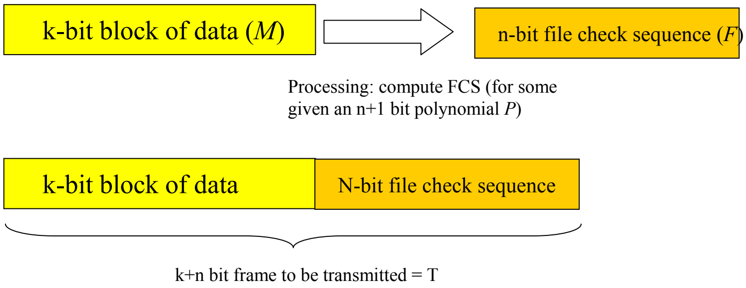
Prob [frame is erroneous] = Prob [1 bit in error] +
Prob[2 bits in error] + … +
Prob[F bits in error]
= 1 – Prob[0 bits in error]
= 1 - $(1-BER)^F$

Error Detection (2)



E, E' = error detecting codes
f = error detecting code function

Cyclic Redundancy Check (CRC)



Processing: compute FCS (for some given an n+1 bit polynomial *P*)

k+n bit frame to be transmitted = T

- Modulo 2 arithmetic (like XOR) is used to generate the FCS:
  - 0 ± 0 = 0; 1 ± 0 = 1; 0 ± 1 = 1; 1 ± 1 = 0
  - 1 X 0 = 0; 0 X 1 = 0; 1 X 1 = 1

CRC – Mapping Binary Bits into Polynomials

- Consider the following k-bit word or frame and its polynomial equivalent:

$$b_{k-1}\ b_{k-2}\ \ldots\ b_2\ b_1\ b_0 \Rightarrow b_{k-1}x^{k-1} + b_{k-2}x^{k-2} + \ldots + b_1x^1 + b_0$$

where b_i $(k-1 \leq i \leq 0)$ is either 1 or 0

- Example1: an 8 bit word M = 11011001 is represented as $M(x) = x^7+x^6+x^4+x^3+1$

CRC – Mapping Binary Bits into Polynomials - cont'd

- Example2: What is $x^4M(x)$ equal to?

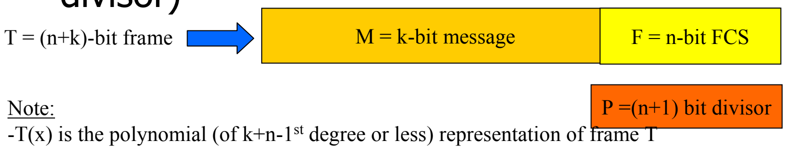
  $x^4M(x) = x^4(x^7+x^6+x^4+x^3+1) = x^{11}+x^{10}+x^8+x^7+x^4$, the equivalent bit pattern is 110110010000 (i.e. four zeros added to the left of the original M pattern)

- Example3: What is $x^4M(x) + (x^3+x+1)$?

  $x^4M(x) + (x^3+x+1) = x^{11}+x^{10}+x^8+x^7+x^4+ x^3+x+1$, the equivalent bit pattern is 110110011011 (i.e. pattern 1011 = x^3+x+1 added to the left of the original M pattern)

CRC Calculation

- T = (k+n)-bit frame to be tx-ed, n < k
- M = k-bit message, the first k bits of frame T
- F = n-bit FCS, the last n bits of frame T
- P = pattern of n+1 bits (a predetermined divisor)



Note:
-T(x) is the polynomial (of k+n-1st degree or less) representation of frame T
-M(x) is the polynomial (of k-1st degree or less) representation of message M
- F(x) is the polynomial (of n-1st degree or less) representation of FCS
- P(x) is the polynomial (of nth degree or less) representation of the divisor P
-T(x) = X^n M(x) + F(x) – refer to example 3 on previous slide

---

CRC Calculation (2)

- Design: frame T such that it divides the pattern P with no remainder?
- Solution: Since the first component of T, M, is the data part, it is required to find F (or the FCS) such that T divides P with no remainder

Using the polynomial equivalent:
$T(x) = X^nM(x) + F(x)$
One can show that $F(x)$ = remainder of $x^nM(x) / P(x)$
i.e if $x^nM(x) / P(x)$ is equal to $Q(x) + R(x)/P(x)$, then $F(X)$ is set to be equal to $R(X)$.

Note that:

Polynomial of degree k+n
------------------------------- = polynomial of degree k + remainder polynomial of degree n-1
Polynomial of degree n

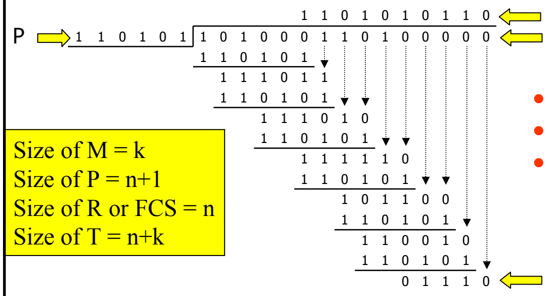
CRC Calculation - Procedure

1. **Shift pattern M n bits to the lift**
2. **Divide the new pattern 2^nM by the pattern P**
3. **The remainder of the division R (n bits) is set to be the FCS**
4. **The desired frame T is 2^nM plus the FCS bits**

Note:
2^nM is the pattern resulting from shifting the pattern M n bits to the left. In other words, the polynomial equivalent of the pattern 2^nM is $x^nM(x)$

CRC Calculation – Example

- **Message M = 1010001101 (10 bits) ➔ k = 10**
  Pattern P = 110101 (6 bits – note 0th and nth bits are 1s)
  ➔n + 1 = 6 ➔ n = 5
  Find the frame T to be transmitted?
- **Solution:**



```
                      1 1 0 1 0 1 0 1 1 0   <= Q
P => 1 1 0 1 0 1 | 1 0 1 0 0 0 1 1 0 1 0 0 0 0 0   <= 2^nM
                   1 1 0 1 0 1
                     1 1 1 0 1 1
                     1 1 0 1 0 1
                         1 1 1 0 1 0
                         1 1 0 1 0 1
                             1 1 1 1 1 0
                             1 1 0 1 0 1
                                 1 0 1 1 0 0
                                 1 1 0 1 0 1
                                   1 1 0 0 1 0
                                   1 1 0 1 0 1
                                     0 1 1 1 0   <= R
```

(note a zero is added to the left to make n bits)

Size of M = k
Size of P = n+1
Size of R or FCS = n
Size of T = n+k

- **FCS = R is equal to 01110**
- **Frame T = 101000110101110**
- **As an exercise, verify that T divided by P has no remainder**

CRC Calculation – The previous example BUT using Polynomials

- Message M = 1010001101 (10 bits) ➔ k = 10
- M(x) = $x^9 + x^7 + x^3 + x^2 + 1$ ➔ $x^5M(x) = x^{14} + x^{12} + x^8 + x^7 + x^5$

  Pattern P = 110101 (6 bits – note 0th and nth bits are 1s) ➔n + 1 = 6 ➔ n = 5
- $P(x) = x^5 + x^4 + x^2 + 1$

  Find the frame T to be transmitted?
- Solution:

$$
\begin{array}{cccc|cccccccccccccc}
\multicolumn{4}{c}{} & \multicolumn{1}{c}{x^9} & +x^8 & & +x^6 & & +x^4 & & +x^2 & +x & & & & & \\
\cline{5-18}
x^5 & +x^4 & +x^2 & +1 & x^{14} & & +x^{12} & & & & +x^8 & +x^7 & & +x^5 & & & & \\
\cline{1-4}
\multicolumn{4}{c}{} & \multicolumn{1}{c}{x^{14}} & +x^{13} & & +x^{11} & & +x^9 & & & & & & & & \\
\cline{5-10}
\multicolumn{4}{c}{} & \multicolumn{1}{c}{} & x^{13} & +x^{12} & +x^{11} & & +x^9 & +x^8 & +x^7 & & +x^5 & & & & \\
\multicolumn{4}{c}{} & \multicolumn{1}{c}{} & x^{13} & +x^{12} & & +x^{10} & & +x^8 & & & & & & & \\
\cline{6-11}
\multicolumn{4}{c}{} & \multicolumn{1}{c}{} & & & x^{11} & +x^{10} & +X^9 & & +x^7 & & +x^5 & & & & \\
\multicolumn{4}{c}{} & \multicolumn{1}{c}{} & & & x^{11} & +x^{10} & & +X^8 & & +x^6 & & & & & \\
\cline{8-13}
\multicolumn{4}{c}{} & \multicolumn{1}{c}{} & & & & & x^9 & +x^8 & +x^7 & +x^6 & +x^5 & & & & \\
\multicolumn{4}{c}{} & \multicolumn{1}{c}{} & & & & & X^9 & +x^8 & & +x^6 & & +x^4 & & & \\
\cline{10-15}
\multicolumn{4}{c}{} & \multicolumn{1}{c}{} & & & & & & & +x^7 & & +x^5 & +x^4 & & & \\
\multicolumn{4}{c}{} & \multicolumn{1}{c}{} & & & & & & & +x^7 & +x^6 & & +x^4 & & +x^2 & \\
\cline{12-17}
\multicolumn{4}{c}{} & \multicolumn{1}{c}{} & & & & & & & & x^6 & +x^5 & & & +x^2 & \\
\multicolumn{4}{c}{} & \multicolumn{1}{c}{} & & & & & & & & x^6 & +x^5 & & +x^3 & & +x \\
\cline{13-18}
\multicolumn{4}{c}{} & \multicolumn{1}{c}{} & & & & & & & & & & & +x^3 & +x^2 & +x
\end{array}
$$

- FCS = $R(x) = x^3 + x^2 + x$

  (or $0x^4+x^3+x^2+x$)

➔ R is equal to 01110

- Frame T = 101000110101110
- As an exercise, verify that T divided by P has no remainder

---

CRC Calculation – The previous example BUT using Polynomials – cont'd

- Message M = 1010001101 (10 bits)

➔ M(x) = $x^9 + x^7 + x^3 + x^2 + 1$

➔ ➔ $x^5M(x) = x^{14} + x^{12} + x^8 + x^7 + x^5$

- Pattern P = 110101

➔ $P(x) = x^5 + x^4 + x^2 + 1$

- $R(x) = x^3 + x^2 + x$
- $Q(x) = x^9+x^8+x^6+x^4+x^2+x$
- $T(X) = x^5M(x) + R(x)$

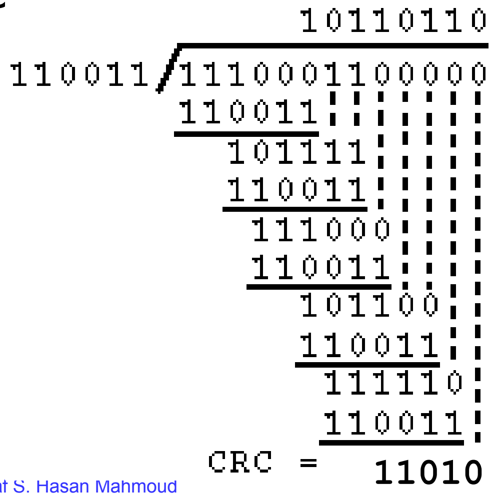
  $= x^{14} + x^{12} + x^8 + x^7 + x^5 + x^3 + x^2 + x$, or

  T = 1010001101**01110**

- <u>Exercise</u>: Verify that $Q(x) P(x) + R(x) = x^5 M(x)$

Example: Problem 6-12

7-11: for P=110011 and M = 11100011, find the CRC

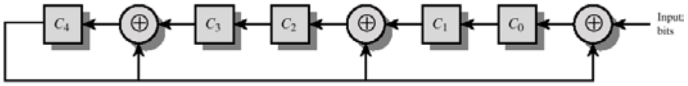


```
                 10110110
        110011/11100011000000
               110011
                101111
                110011
                 111000
                 110011
                  101100
                  110011
                   111110
                   110011
        CRC =       11010
```

CRC – Receiver Procedure

- Tx-er transmits frame T
- Channel introduces error pattern E
- Rx-er receives frame $T_r = T \oplus E$ (note that if E = 000..000, then Tr is equal to T, i.e. error free transmission)
- T_r is divided by P, Remainder of division is R
- if R is ZERO, Rx-er assumes no errors in frame; else Rx-er assumes erroneous frame
- If an error occurs and T_r is still divisible by P → UNDETECTABLE error (this means the E is also divisible by P)

CRC – Transmitter Circuit



☐ = 1-bit shift register ⊕ = Exclusive-OR circuit

(a) Shift-register implementation

Shift register circuit for dividing by
P = $X^5+X^4+X^2+1$

	C_4	C_3	C_2	C_1	C_0	$C_4 \oplus C_3$	$C_4 \oplus C_1$	$C_4 \oplus$ input	input
Initial	0	0	0	0	0	0	0	1	1
Step1	0	0	0	0	1	0	0	0	0
Step2	0	0	0	1	0	0	1	1	1
Step3	0	0	1	0	1	0	0	0	0
Step4	0	1	0	1	0	1	1	0	0
Step5	1	0	1	0	0	1	1	1	0
Step6	1	1	1	0	1	0	1	0	1
Step7	0	1	1	1	0	1	1	1	1
Step8	1	1	1	0	1	0	1	1	0
Step9	0	1	1	1	1	1	1	1	1
Step10	1	1	1	1	1	0	0	1	0
Step11	0	1	0	1	1	1	1	0	0
Step12	1	0	1	1	0	1	0	1	0
Step13	1	1	0	0	1	0	1	1	0
Step14	0	0	1	1	1	0	1	0	0
Step15	0	1	1	1	0	1	1	0	—

Message to be sent (Initial–Step9)

Five zeros added (Step10–Step14)

(b) Example with input of 1010001101

CRC – Receiver Circuit

- **Tx-er transmits frame T**

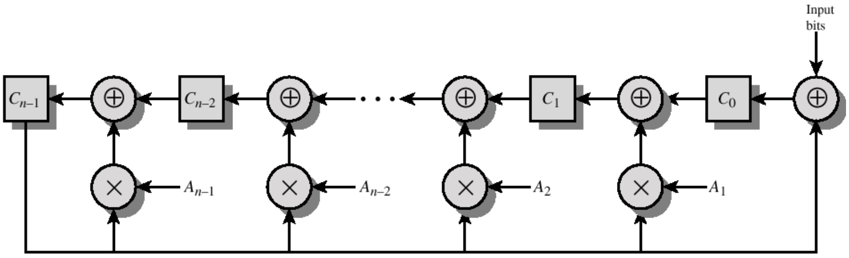


Figure 7.7 General CRC Architecture to Implement Divisor
$1 + A_1X + A_2X^2 + \ldots + A_{n-1}X^{n-1} + X^n$

Cyclic Redundancy Check (CRC)

- Animation for CRC Calculation

Example: Problem 6-13

A CRC is constructed to generate a 4-bit FCS for an 11-bit message. The generator polynomial is X^4+X^3+1

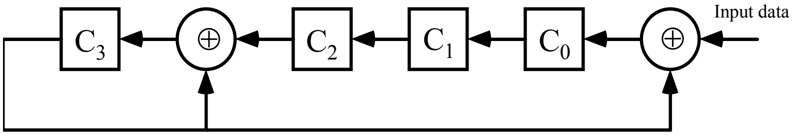
a) Draw the shift register circuit that would perform this task (see figure 6.5)

b) Encode the data bit sequence 10011011100 (leftmost bit is the LSB) using the generator polynomial and give the code word

c) Now assume that bit 7 (counting from the LSB) in the code word is in error and show that the detection algorithm detects the error

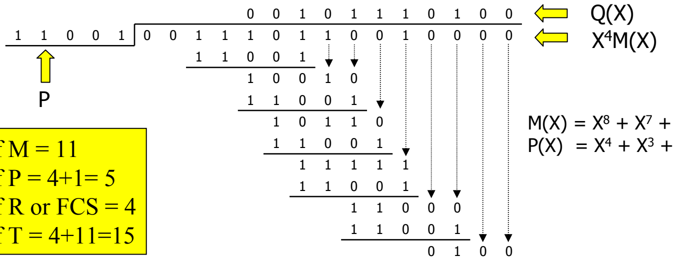
Example: Problem 6-13 - solution

a)



Example: Problem 6-13 - solution

b)



```
                    0 0 1 0 1 1 1 0 1 0 0
1 1 0 0 1 | 0 0 1 1 1 0 1 1 0 0 1 0 0 0 0
                1 1 0 0 1
                    1 0 0 1 0
                    1 1 0 0 1
                      1 0 1 1 0
                      1 1 0 0 1
                        1 1 1 1 1
                        1 1 0 0 1
                            1 1 0 0 0
                            1 1 0 0 1
                                  0 1 0 0
```

$M(X) = X^8 + X^7 + X^6 + X^4 + X^3 + 1$
$P(X) = X^4 + X^3 + 1$

Size of M = 11
Size of P = 4+1= 5
Size of R or FCS = 4
Size of T = 4+11=15

➔ R = 0 1 0 0 or $R(X) = X^2$

Transmitted Frame T = 001110110010100

$T(X) = X^4M(X) + R(X) = X^{12}+X^{11}+X^{10}+X^8+X^7+X^4+X^2$

Notes:

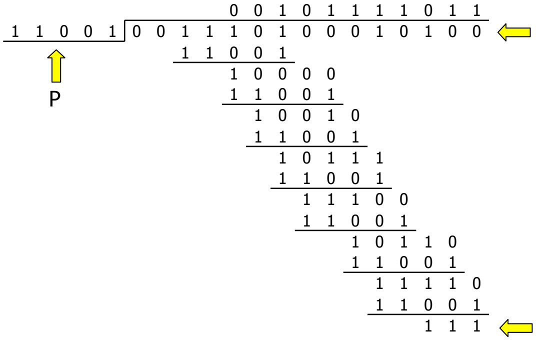
1. $X^4M(X)/P(X) = Q(X) + R(X)/P(X)$, where $Q(X) = X^8 + X^6 + X^5 + X^4 + X^2$ (as seen from the long division process)
2. One can verify that P(X) Q(X) + R(X) is indeed equal to $X^4M(X)$ {note that for the addition of polynomial terms modulo-2 applies; i.e. $X^9+X^9 = 0$}

Example: Problem 6-13 - solution

c) Received frame (LSB from the left) = 0 0 1 0 1 0 0 0 1 0 1 1 1 0 0

dividing by P yields a nonzero remainder ➔ error is detected

Remainder = 0111



```
                    0 0 1 0 1 1 1 1 0 1 1
1 1 0 0 1 | 0 0 1 1 1 0 1 0 0 0 1 0 1 0 0
                1 1 0 0 1
                    1 0 0 0 0
                    1 1 0 0 1
                      1 0 0 1 0
                      1 1 0 0 1
                        1 0 1 1 1
                        1 1 0 0 1
                          1 1 1 0 0
                          1 1 0 0 1
                              1 0 1 1 0
                              1 1 0 0 1
                                1 1 1 1 0
                                1 1 0 0 1
                                    1 1 1
```

RECEIVED FRAME

P

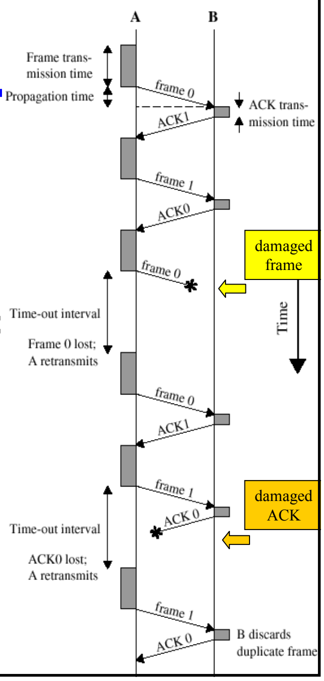
NON ZERO REMAINDER

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
  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

Check for status of B before resending the frame

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

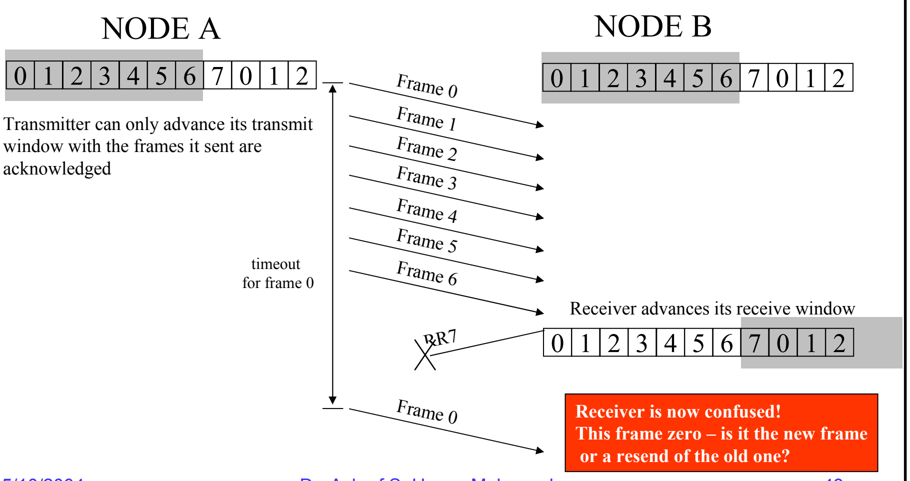
---

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



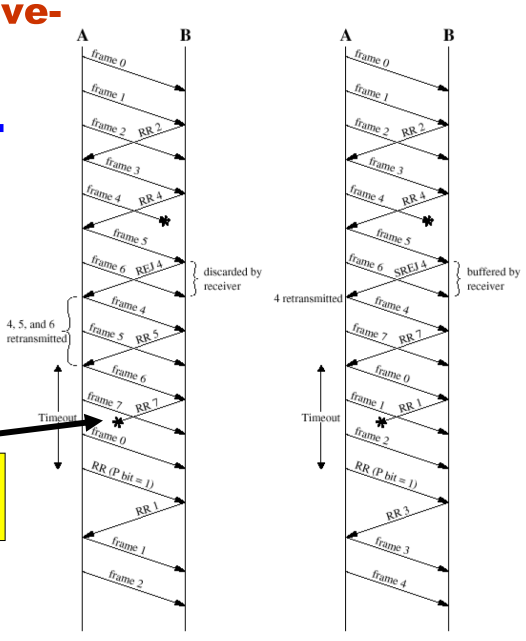
Transmitter can only advance its transmit window with the frames it sent are acknowledged

Receiver advances its receive window

**Receiver is now confused!
This frame zero – is it the new frame or a resend of the old one?**

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



(a) Go-back-N ARQ

(b) Selective-reject ARQ

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

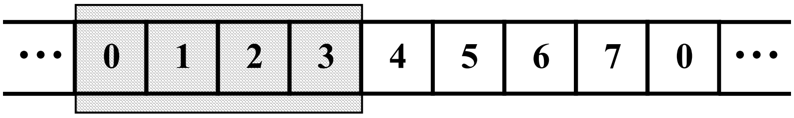
Example: Problem 7-9

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:

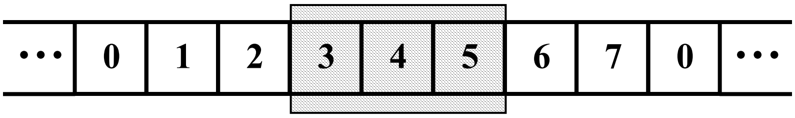
- **a) Before A sends any frames**
- **b) After A sends frame 0, 1, 2 and B acknowledges 0, 1 and the ACKs are received by A**
- **c) After A sends frames 3, 4, and 5 and B acknowledges 4 and the ACK is received by A**

Example: Problem 7-9 - Solution

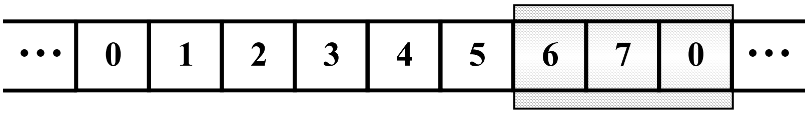
a)



b)



c)



High-Level Data Link Control Protocol (HDLC)

- One of the most important data link control protocols
- Basic Characteristics:
  - Primary Station: issues *commands*
  - Secondary Station: issues *responses* – operates under the control of a primary station
  - Combined Station: issues commands and responses
- Two link configurations are defined:
  - Unbalanced: one primary plus one or more secondary
  - Balanced: two combined (functions as primary and/or secondary) stations

---

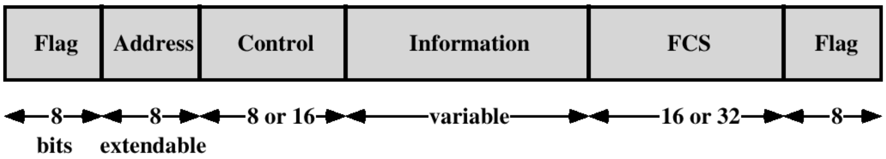
High-Level Data Link Control Protocol (HDLC) (2)

- Three transfer modes are defined:
  - Normal Response Mode (NRM) – used in unbalanced conf.; secondary may only tx data in response to a command from primary
  - Asynchronous Balanced Mode (ABM) – used in balanced conf.; either combined station may tx data without receiving permission from other station
  - Asynchronous Response Mode (ARM) – used in unbalanced conf.; Secondary may initiate data tx without explicit permission; primary still retains line control (initialization, error recovery, …)
- Animation for HDLC

HDLC - Applications

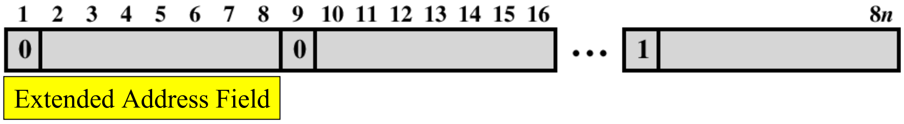
- NRM:
  - Point-multi-point (multi-drop line): one computer (primary) polls multiple terminals (secondary stations)
  - Point-to-point: computer and a peripheral
- ABM: most widely used (no polling involved)
  - Full duplex point-to-point
- ARM: rarely used

HDLC – Frame Structure – Flag Field



- Flag Field: unique pattern 01111110
  - Used for synchronization
  - To prevent this pattern form occurring in data ➔ *bit stuffing*
    - Tx-er inserts a 0 after each 5 1s
    - Rx-er, after detecting flag, monitors incoming bits – when a pattern of 5 1s appears; the 6th/7th bit are checked:
      - If 0, it is deleted
      - If 10, this is a flag
      - If 11, this is an ABORT
- Pitfalls of bit stuffing: one bit errors can split one frame into two or merge two frames into one

HDLC – Frame Structure - Address Field



- Address field identifies the secondary station that transmitted or is to receive frame
- Not used (but included for uniformity) for point-to-point links
- Extendable – by prior arrangement
- Address = 11111111 (single octet) used for broadcasting; i.e. received by all secondary stations

---

HDLC – Frame Structure - Control Field

	1	2	3	4	5	6	7	8
I: Information	0	N(S)			P/F	N(R)		
S: Supervisory	1	0	S		P/F	N(R)		
U: Unnumbered	1	1	M		P/F	M		

N(S) = Send sequence number
N(R) = Receive sequence number
S = Supervisory function bits
M = Unnumbered function bits
P/F = Poll/final bit

- First 2 bits of field determine the type of frame
  - Information frame (I): carry user data (upper layers) – flow and error control info is piggybacked on these frames as well
  - Supervisory frame (S): carry flow and error control info when there is no user data to tx
  - Unnumbered frame (U): provide supplementary link control
- Poll/Final (P/F) bit:
  - In command frames (P): used to solicit response from peer entity
  - In response frames (F): indicate response is the result of soliciting command

HDLC – Frame Structure - Control Field (2)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Information	0	N(S)							P/F	N(R)						
Supervisory	1	0	S		0	0	0	0	P/F	N(R)						

- "Set-mode" command ➔ extends control field to 16 bit for S and I frames
- Extension: 7-bit sequence numbers rather than 3-bit ones
- Unnumbered frames always use 3-bit sequence numbers

HDLC – Frame Structure – Information/FCS Fields

- Information field:
  - Present ONLY in I-frames and some U-frames
  - Contains integer number of octets
  - Length is variable – up to some system defined maximum
- FCS field:
  - Error detecting code
  - Calculated from *ALL* remaining bits in frame
  - Normally 16 bits (CRC-CCITT polynomial = $X^{16}+X^{12}+X^5+1$)
  - 32-bit optional FCS

HDLC Operation

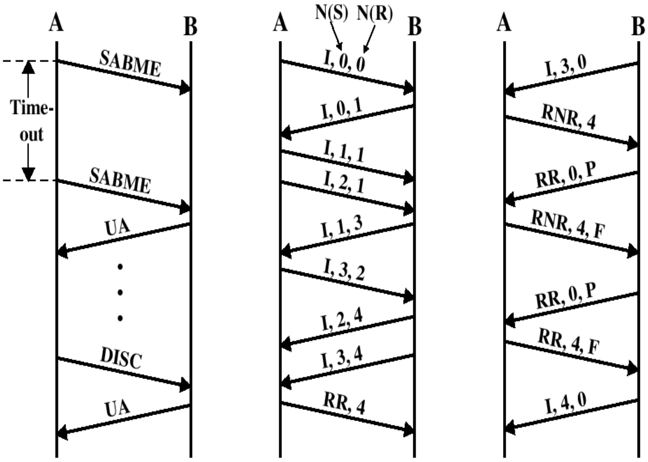
- Initialization
  - One side signals to the other the need for initialization
  - Specifies which of the three modes to use: NRM, ABM, or ARM
  - Specifies 3- or 7-bit sequence numbers
  - The other side can accept by sending unnumbered acknowledgment (UA)
  - The other side can reject by sending - A disconnected mode (DM) frame is sent
- Data Transfer
  - Exchange of I-frames: data and can perform flow/error control
  - S-frames can be used as well: RR, RNR, REJ, or SREJ
- Disconnect
  - DISC frame ➔ UA

HDLC – Operation

a) Link Setup & Disconnect:
  - SABM command – starts timer
  - B responds with UA (or DM if not interested)
  - A receives UA and initializes its variables
  - To disconnect: issue DISC command

b) Two-Way Data Exchange:
  - Full-duplex exchange of I-frames

c) Busy Condition:
  - Note the use of the P and F bits

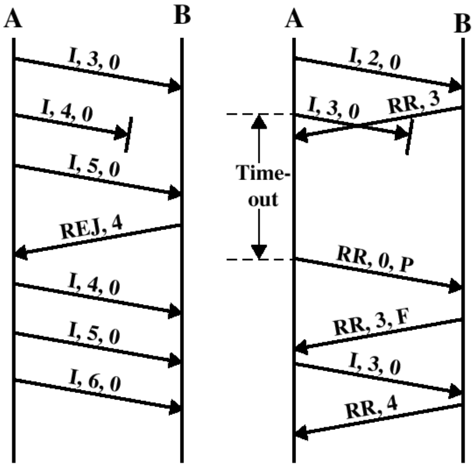


(a) Link setup and disconnect (b) Two-way data exchange (c) Busy condition

HDLC – Operation (2)

a) Reject Recovery:
- I-frame 4 was lost
- B receives I-frame 5 (out of order) – responds with REJ 4
- A resend I-frame 4 and all subsequent frames (Go-back-N)

b) Timeout Recovery:
- A sends I-frame 3 – but it is lost
- Timer expires before acknowledgement arrives
- A polls Node B
- B responds indicating it is still waiting for frame 3 – B set the F bit because this a response to A's solicitation



(d) Reject recovery (e) Timeout recovery

Other Data Link Control Protocols

- Link Access Procedure – Balanced (LAPB):
  - Part of X.25 packet-switching interface standard
  - Subset of HDLC – only ABM is provided
  - Designed for point-to-point
  - Frame format is same as HDLC
- Link Access Procedure – D-Channel (LAPD):
  - Part of ISDN – functions on the D-channel
  - 7-bit sequence numbers only
  - FCS field is always 16-bit
  - 16-bit address fields (two sub-addresses)

Other Data Link Control Protocols (2)

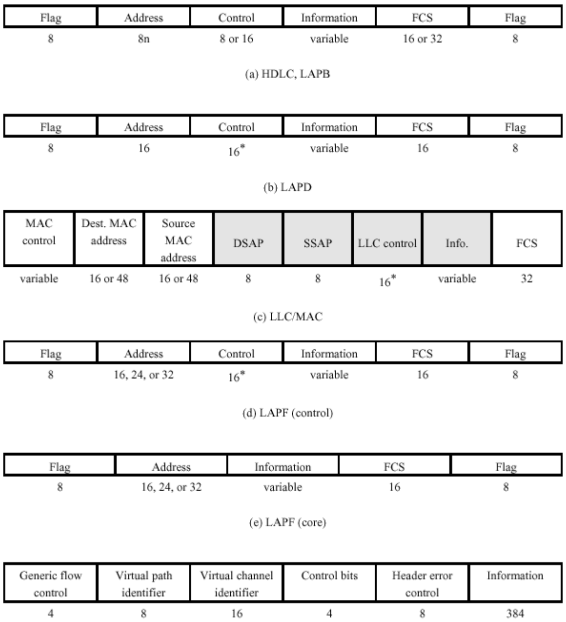
- Logical Link Control (LLC):
  - Part of IEEE802 family for LANs
  - Different frame format than HDLC
- Link Access Control Protocol for Frame-Mode Bearer Service (LAPF):
  - Designed for Frame Relay Protocol
  - Provides only ABM mode
  - Only 7-bit sequence numbers
  - Only 16-bit CRC field
  - Address field is 16, 24, or 32 bits long – containing a 10-bit, 16-bit, or 23-bit data link connection identifier (DLCI)
  - No control field – I.e. CANNOT do flow or error control (remember that frame relay was designed for fast and reliable connections!)

Other Data Link Control Protocols (3)

- Asynchronous Transfer Mode (ATM):
  - Like frame relay designed for fast and reliable links
  - NOT based on HDLC
  - New frame format – called CELL (53 bytes: 48 Bytes for payload or user data and 5 Bytes for overhead)
  - Cell has minimal overhead
  - NO error control for payload

Other Data Link Control Protocols (4)

- Frame Formats



Flag	Address	Control	Information	FCS	Flag
8	8n	8 or 16	variable	16 or 32	8

(a) HDLC, LAPB

Flag	Address	Control	Information	FCS	Flag
8	16	16*	variable	16	8

(b) LAPD

MAC control	Dest. MAC address	Source MAC address	DSAP	SSAP	LLC control	Info.	FCS
variable	16 or 48	16 or 48	8	8	16*	variable	32

(c) LLC/MAC

Flag	Address	Control	Information	FCS	Flag
8	16, 24, or 32	16*	variable	16	8

(d) LAPF (control)

Flag	Address	Information	FCS	Flag
8	16, 24, or 32	variable	16	8

(e) LAPF (core)

Generic flow control	Virtual path identifier	Virtual channel identifier	Control bits	Header error control	Information
4	8	16	4	8	384

(f) ATM

Textbook Problems of INTEREST

- **Textbook: 6.10, 6.12^{s}, 6.13^{s}, 6.14 7-2, 7-3, 7-4$^{s\text{(in class)}}$, 7-5, 7-9^{s}, 7-12, 7-18**
- **Homework: 6.14, 7.2, 7.5**