

Data Link Control - DLC

Presentation D

Study: 6.1, 6.2, 6.3, 6.4 (pages 196,197),
6.5, 7.1, 7.2, 7.3

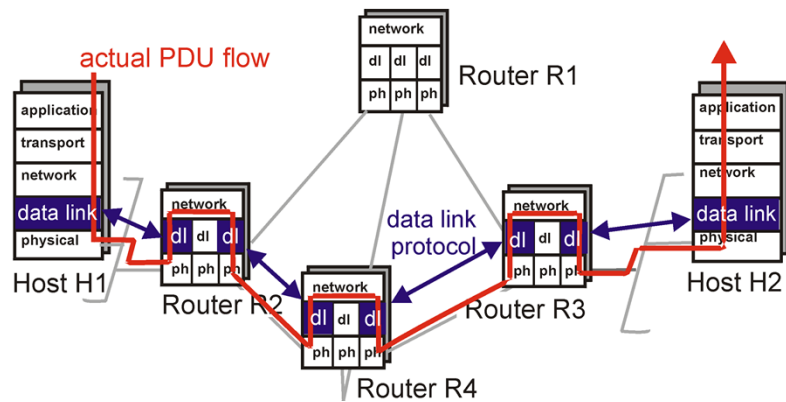
Kannan Srinivasan

09-11-2012

Introduction

- So far we have studied the physical layer, which is primarily concerned with how to transfer a stream of bits over a given transmission system from one to another end (device). Some error detecting and synchronization can be included at the physical level.
- For two devices linked by a transmission system to exchange data, a higher degree of cooperation is required and that is provided by the Data Link Control (DLC) layer.
- DLC is concerned with blocks of bits, called frames, and it assumes services of the physical layer and provides a layer of logic above it.
- When DLC is used, the transmission system between two communicating systems is referred to as a data link.

Link Layer: setting the context

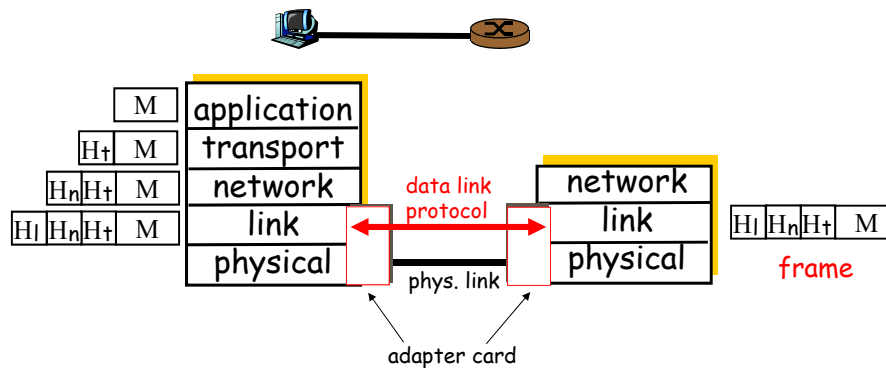


d. xuan

3

Link Layer: setting the context

- two *physically connected* devices:
— host-router, router-router, host-host
- unit of data: *frame*



d. xuan

4

Data Link Control Functions

- **frame synchronization**: a mechanism to synchronize messages at the transmitter and receiver;
- **framing and link access**
 - encapsulation of packets (datagrams) into frames adding header & trailer; **addresses in header not IP addresses**;
 - implementing access to transmission medium (easier for point to point links, than for more involved shared medium)
- **error detection** (and **correction**): detection (correction) of a frames with errors introduced by the transmission system;
- **flow control**: ensuring the sending entity does not overwhelm the receiving entity; important even over an error free link;
- **error control**: its function is to deliver frames without errors, duplication and in the proper order to the upper layer;
 - seldom used on low bit error links, such as fiber optic links

g. babic

Presentation D

5

Asynchronous Transmission

- A strategy for frame synchronization that avoids timing problem by sending small number of bits:
 - data transmitted one character at a time
 - timing only needs maintaining within each character
 - resynchronize with each character

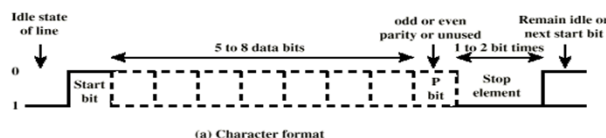


Figure 6.1

- Asynchronous serial bit stream:
 - speed in bits/sec should be given in advance
 - number of bits/character should be given in advance
 - start bit and stop element (1 or 2 bits)
 - parity bit and parity error (odd and even parity)
 - framing error

g. babic

Presentation D

6

Asynchronous: Behavior & Timing Error

- In a steady stream of characters sent, intervals between characters are uniform (length of stop element).
- After stop element, i.e. in idle state, receiver looks for transition 1 to 0, then samples next 5-8 intervals (character length), and then looks for stop element.

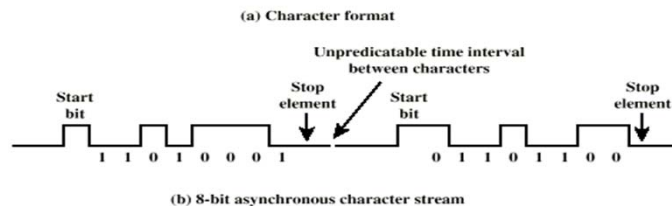


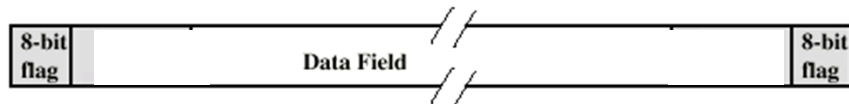
Figure 6.1

Synchronous Transmission: Bit Level

- A strategy for frame synchronization of larger blocks of data
- Can use separate clock line
 - Good over short distances
 - Subject to impairments
- Embed clock signal in data
 - e.g. Manchester encoding
 - e.g. Carrier frequency (analog)
- More efficient, i.e. lower overhead than asynchronous (with 2-3 bit overhead per character, i.e. $\approx 20\%$ overhead)
- But how to indicate start and end of block?

Synchronous Transmission: Block Level

- Use pre-amble and post-amble:
 - e.g. series of SYN (hex 16) characters
 - e.g. flag sequence "01111110" at the beginning and at the end of a block; very often used



- But there is a problem with flag sequence, as with any other approach above!
- What if flag pattern appears somewhere in a block?
- Receiver interprets the received sequence as two blocks.

g. babic

Presentation D

9

Bit Stuffing Procedure

- At transmitter:
 - When it finds in the transmitting block a sequence of 5 ones, it sends one additional 0 after the sequence of those 5 ones.
 - Thus a flag pattern can't appear in the middle of a block
 - Example
original pattern: 000111111111111001111100111110
transmitted pattern: 00011111011111011001111100011111010
- At receiver:
 - When it finds in the receiving block a sequence of 5 ones followed by 0, it discards the 0 and accepts only 5 ones.
 - Note: a sequence of 6 ones followed by 0 is interpreted as the flag sequence, while a pattern with more than 6 ones in sequence is considered as a corrupted block.

g. babic

Presentation D

10

Types of Data Error

- An error occurs when a bit is altered between transmission and reception
- **Single bit errors:** One bit altered and adjacent bits not affected
 - P_b – probability a bit is received in error; called **the bit error rate** or **BER**
 - $P_1 = (1 - P_b)^F$; probability a F-bit block arrives with no bit errors
 - $P_2 = 1 - P_1$; probability a block arrives with errors
- **Burst errors:** a burst error of length B is a contiguous sequence of B bits in which first, last and any number of intermediate bits are in error
 - Impulse noise or fading in wireless
 - Effect greater at higher data rates

g. babic

Presentation D

11

Principle of Error Detecting Process

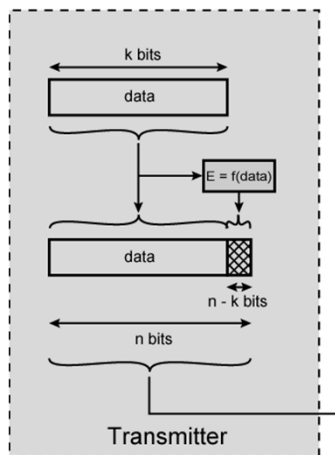


Figure 6.3

- n-k additional bits added by transmitter for error detection code

g. babic

Presentation D

12

Error Detecting Techniques

- **Parity check** – simplest method
 - Value of parity bit is such that character has even (even parity) or odd (odd parity) number of ones
 - Even number of bit errors goes undetected
- **Cyclic Redundancy Check – CRC**
 - one of the most common and the most powerful methods
 - functions slightly different (although equivalent) to the **principle of error detecting process** on the previous slide
 - For a block of k bits transmitter generates additional $n-k$ bit sequence, called CRC code or frame check sequence - FCS
 - Transmitted n bits are exactly divisible by some number
 - Receiver divides received block by that number and if no remainder, assumes no error

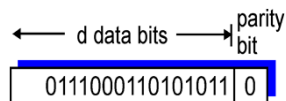
g. babic

Presentation D

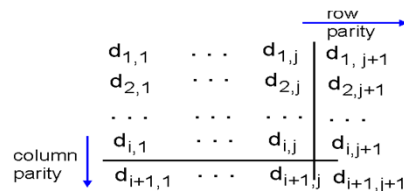
13

Parity Checking

Single Bit Parity: Detect single bit errors



Two Dimensional Bit Parity: Detect and correct single bit errors



| | | | | | |
|-----------|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| no errors | | | | | |

| | | | | | |
|------------------------------|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| parity error | | | | | |
| correctable single bit error | | | | | |

d. xuan

14

Cyclic Redundancy Check - CRC

- Modulo 2 arithmetic used, i.e. addition without carry and subtraction without borrow
- Equivalent to the exclusive-OR (XOR) operation
- Examples:

$$\begin{array}{r}
 10011 \\
 + 11001 \\
 \hline
 01010
 \end{array}
 \qquad
 \begin{array}{r}
 10011 \\
 - 11001 \\
 \hline
 01010
 \end{array}
 \qquad
 \begin{array}{r}
 10111 \times 101 \\
 \hline
 10111 \\
 00000 \\
 \hline
 10111 \\
 1001011
 \end{array}$$

- CRC principles:
 - D = k-bit block of data
 - F = (n-k)-bit FCS
 - T = n-bit frame to be transmitted
 - P = (n-k+1)-bit pattern; that is predetermined divisor

g. babic

Presentation D

15

CRC: Principles and Example

Given: D = 1010001101 (10 bits) and P = 110101 (6 bits)

Find: FCS

Since, k = 10 and (n - k + 1) = 6 → n = 15, then

FCS = xxxxx (n - k = 5 bits)

Step 1: $D \times 2^{(n-k)} = 1010001101 \times 2^5 = 101000110100000$

Step 2: $D \times 2^{(n-k)}$ is divided by P:

$$\begin{array}{r}
 110101 \overline{) 101000110100000} \\
 \underline{110101} \\
 111011 \\
 \underline{110101} \\
 111010 \\
 \underline{110101} \\
 111110
 \end{array}$$

g. babic

Presentation D

16

CRC: Principles and Example (continued)

$$\begin{array}{r}
 110101 \overline{) 101000110100000} \\
 \underline{110101} \\
 111011 \\
 \underline{110101} \\
 111010 \\
 \underline{110101} \\
 111110 \\
 \underline{110101} \\
 101100 \\
 \underline{110101} \\
 110010 \\
 \underline{110101} \\
 1110
 \end{array}$$

1110 → remainder = 01110

Step 3: Remainder added to the product from Step 1 to give T to be sent: T = 101000110101110

g. babic

Presentation D

17

CRC: Principles and Example (continued)

Step 4: At receiver, the received frame T' is divided by P. If there is a non-zero remainder then received block in error.

Let assume that block has not changed.

Thus, the block 101000110101110 received.

$$\begin{array}{r}
 110101 \overline{) 101000110101110} \\
 \underline{110101} \\
 11101 \\
 \dots \\
 110101 \\
 \underline{110101} \\
 0
 \end{array}$$

0 → remainder = 0

Because there is zero remainder, it is assumed that there have been no errors in received block.

g. babic

Presentation D

18

CRC and Polynomials

- It turns out that more convenient usage (and theory) can be developed using polynomial representation instead of bit representation.
- Example: **Given:** $D = 1010001101$ and $P = 110101$
Find: FCS
 $D = 1010001101 \rightarrow D(x) = x^9 + x^7 + x^3 + x^2 + 1$
 $P = 110101 \rightarrow P(x) = x^5 + x^4 + x^2 + 1$ (Note: $n-k = 5$)
- **Step 1:** multiply $D(x)$ by x^{n-k} : $x^5 \times D(x) = x^{14} + x^{12} + x^8 + x^7 + x^5$
- **Step 2:** divide (mod 2) $D(x)$ by $P(x)$ and find remainder $R(x)$
 See Figure 6.4 in the textbook $\rightarrow R(x) = x^3 + x^2 + x = 01110$
- **Step 3:** add the result of step 2 to step 1 to obtain $T(x)$:
 $T(x) = x^{14} + x^{12} + x^8 + x^7 + x^5 + x^3 + x^2 + x = 101000110101110$
- **Step 4:** A receiver divides the polynomial corresponding to the received sequence of bits by $P(x)$ and if the remainder is not zero, it assumes that a message is in error.

g. babic

Presentation D

19

Choosing Polynomial $P(x)$

- Highest power of $P(x)$ is chosen to be the length of the desired FCS, i.e. equal to $(n-k)$. At the minimum, $P(x)$ should have nonzero coefficients for its $x^{(n-k)}$ and x^0 terms.
- It can be shown that by a suitably chosen $P(x)$, the following errors are not divisible, thus detectable:
 - all single errors, if $P(x)$ has more than one nonzero term,
 - all double errors, as long as $P(x)$ has at least three terms,
 - any odd number of errors, as long as $P(x)$ contains a factor $(x+1)$;
 - any burst error for which the length of the burst is less or equal to $(n-k)$, i.e. the length of FCS.
 - most larger burst errors.
- **Examples of commonly used $P(x)$:**
 - **CRC-16** = $x^{16} + x^{15} + x^2 + 1$ – **CRC-CCITT** = $x^{16} + x^{12} + x^5 + 1$
 - **CRC-32** = $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + 1$

g. babic

Presentation D

20

Error Correction Codes

- If an error in a frame is detected, receiver simply discards the frame. Thus, correction of detected errors usually requires data block to be retransmitted
- Retransmissions are not appropriate for wireless applications
 - Bit error rate is high
 - Lots of retransmissions
- Propagation delay can be long (satellite) compared with frame transmission time
 - Would result in retransmission of frame in error plus many subsequent frames
- Thus, there is a need to correct errors on basis of bits received → **error correction techniques**

g. babic

Presentation D

21

Error Correction Process Diagram

FEC – Forward Error Correcting

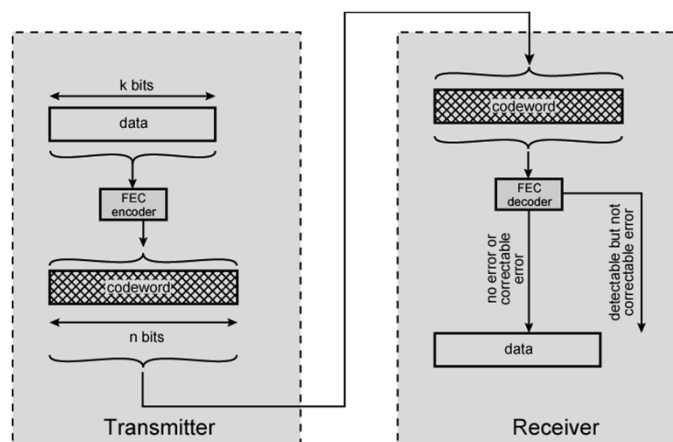


Figure 6.7

g. babic

Presentation D

22

Error Correction Process

- Each k -bit block data mapped to an n -bit ($n > k$) codeword
 - Forward error correction (FEC) encoder
 - codeword, it may have no resemblance to k -bit data
 - block error correction code adds $(n - k)$ bits
- Codeword sent
- Received codeword passed to FEC decoder; it can deduce original in face of certain level of error rate:
 - If no errors, original data block output
 - Some error patterns can be detected and corrected
 - Some error patterns can be detected but not corrected
 - Some (rare) error patterns are not detected and incorrect data output from FEC decoder

g. babic

Presentation D

23

Frame Transfer Time

- Find time T needed to transfer 2,000byte message from one device to another one connected over a 100 mile fiber link with capacity of 10Mbps.
- Transfer time = Transmission time + Propagation time
- Transmission time – time taken to emit all bits into medium:

$$t_{\text{frame}} = \text{message size} / \text{channel capacity}$$
- Propagation time: time for a bit to traverse the link:

$$t_{\text{prop}} = \text{distance} / \text{signal speed}$$
 - signal speed = 300 meters/ μsec for light in vacuum
 - = 200 meters/ μsec for light in fiber
 - = 250 meters/ μsec for electricity over copper
- $T = t_{\text{frame}} + t_{\text{prop}} = 2000 \times 8 / 10 \times 10^6 + 100 \times 1600 / 200 \times 10^6$

$$= 1.6 \text{ millisec} + 0.8 \text{ millisec} = 2.4 \text{ millisec}$$

g. babic

Presentation D

24

Model of Frame Transfer

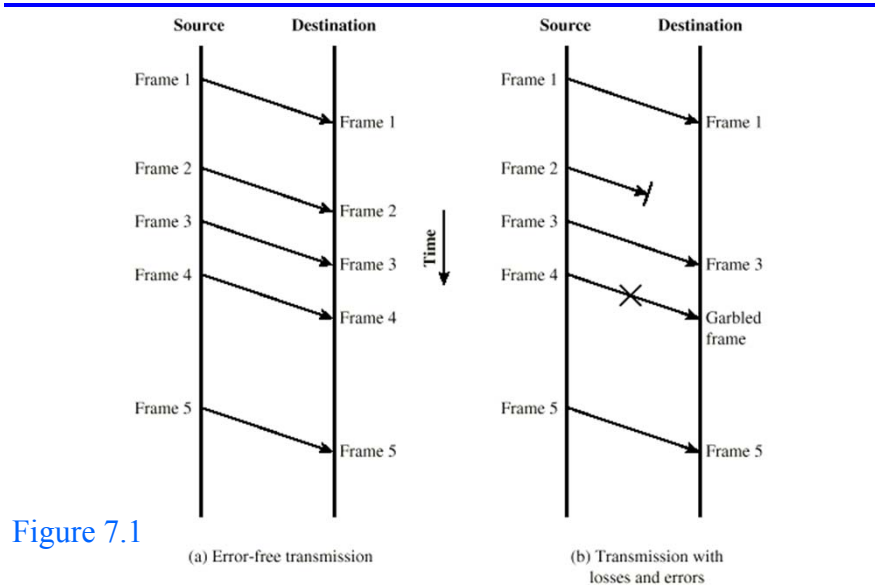


Figure 7.1

(a) Error-free transmission

(b) Transmission with losses and errors

g. babic

Presentation D

25

Stop and Wait Flow Control

- Basic functioning:
 - Source transmits info-frame (a frame with data)
 - Destination receives frame and replies with acknowledgement (ACK-frame)
 - Source waits for ACK-frame before sending next frame
 - Destination can stop flow by not sending ACK-frame
- In summary:

A sender may send the next info-frame only after an acknowledgment from a receiver is received for the previously sent info-frame. The acknowledgment is also a frame (control one), but short and sent by a receiver.

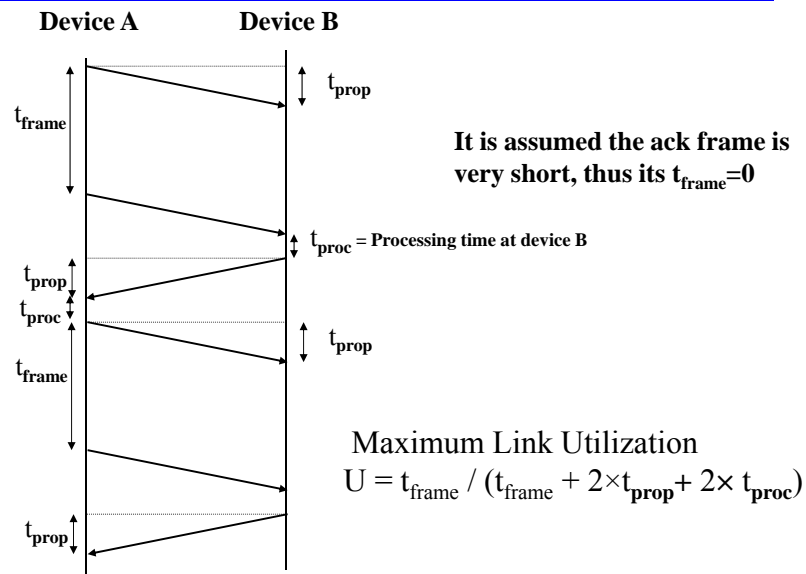
The receiver of an info-frame sends an acknowledgment back when it is ready to receive the next info-frame.
- Note: there is always a frame processing time (in addition).

g. babic

Presentation D

26

Stop and Wait Flow Control: Example



g. babic

Presentation D

27

Stop and Wait Link Utilization

$$U = t_{\text{frame}} / (t_{\text{frame}} + 2t_{\text{prop}} + 2t_{\text{proc}}) = 1/(2a + 1),$$

where $a = t_{\text{prop}}/t_{\text{frame}}$, and assuming $t_{\text{proc}}=0$

Examples of calculating U:

- Satellite link: Known $t_{\text{prop}} = 270$ millisecc, assume 4000b frame and bit rate = 56kbps;
 $t_{\text{frame}} = 4000/56000 = 71$ millisecc;
 $a = 270/71 = 3.8$;
 $U = 1/(2 \times 3.8 + 1) = 0.12$
- LAN link: Assuming 1000 meter fiber $\rightarrow t_{\text{prop}} = 5 \mu\text{sec}$, also assume Frame size = 4000 bits and bit rate = 10Mbps
 $t_{\text{frame}} = 4000/10\text{M} = 400\mu\text{sec}$;
 $a = 5/400 = 0.012$;
 $U = 1/(2 \times 0.012 + 1) = 0.98$

g. babic

Presentation D

28

Stop and Wait Link Utilization (continued)

Taken that $t_{\text{frame}} = 1 \rightarrow a = t_{\text{prop}}$

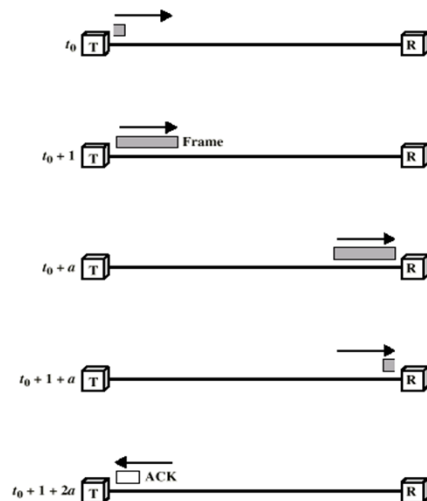


Figure 7.2
g. babic

(a) $a > 1$
Satellite link

Presentation D

LAN link

29

Stop and Wait Flow Control: Summary

- This technique is appropriate for short distances, long frames and short processing times (at receiver). But, a longer frame implies higher probability of error in transmission.
- Thus, large block of data are required to split into smaller frames and:
 - Errors detected sooner (when a frame is received)
 - On error, retransmission of smaller frames is needed
 - Prevents one station occupying medium for long periods
 - Problem of limited buffer size at receiver resolved.
- But, the stop and wait flow control is very often inadequate.

g. babic

Presentation D

30

Sliding Windows Flow Control

- Allows multiple info-frames to be in transit, in one direction
- Transmitter can send up to W unacknowledged info-frames
- W = maximum window size (agreed upon in advance)
- Receiver has buffer for W info-frames
- Each info-frame is numbered
- A sequence number of the first info-frame sent is 0, and for each next info-frame sent this number is incremented by one.
- Sequence numbers are bounded by size of send a sequence number field (=N bits) in the header of info-frame; Thus, info-frames are numbered modulo 2^N , i.e. 0, 1, 2, ..., 2^N-1 , 0, 1, 2, ..., 2^N-1 , 0, 1, 2, ...
- Acknowledgement frames are sent by receiver as a response

g. babic

Presentation D

31

Sliding Window Control: Details

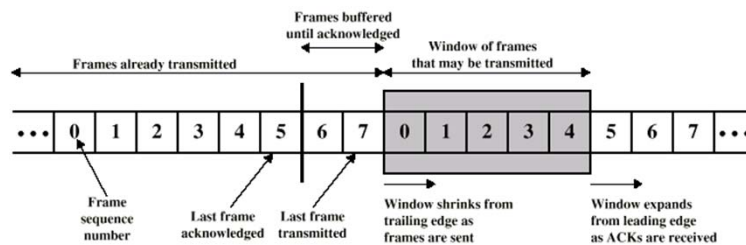
- At any moment, each a sender or a receiver maintains its window (for the given direction). A window size dynamically changes between 0 to W.
- A sender window includes sequence numbers of frames that can be sent. Each time, a sender sends an info-frame its window shrinks for a sequence number of the sent frame. When a window size is 0, a sender is not allowed to send any new info-frame.
- A receiver window includes sequence numbers of info-frames that can be received. Each time, a receiver receives an info-frame with a sequence number at the beginning of the current window, its window shrinks for that sequence number. When its window size is 0, a receiver is ignoring any new info-frame.

g. babic

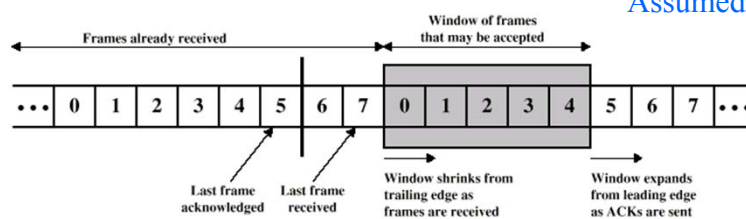
Presentation D

32

Sliding Window Diagram



(a) Sender's perspective



(b) Receiver's perspective

Figure 7.3

g. babic

Presentation D

33

Sliding Window Control: Details (continued)

- A receiver responds with acknowledgment frames. Those control frames are called **RR** (Receiver Ready) frames.
- RR frame includes a **N-bit receive sequence number field**. When receiver sends **RR(x)** to the sender, x is a content of a receive sequence field, it means that all info-frames with sequence numbers x-1, and lower are acknowledged and that a receiver is ready to receive info-frames with sequence numbers x, x+1, x+2,..., x+W-1.

Since RR control frame may enlarge a window, and a receiver will send such RR frame only if it is ready to receive that many additional info-frames.

- When a sender receives **RR(x)**, it sets its window at x, x+1, ..., x+W-1.

g. babic

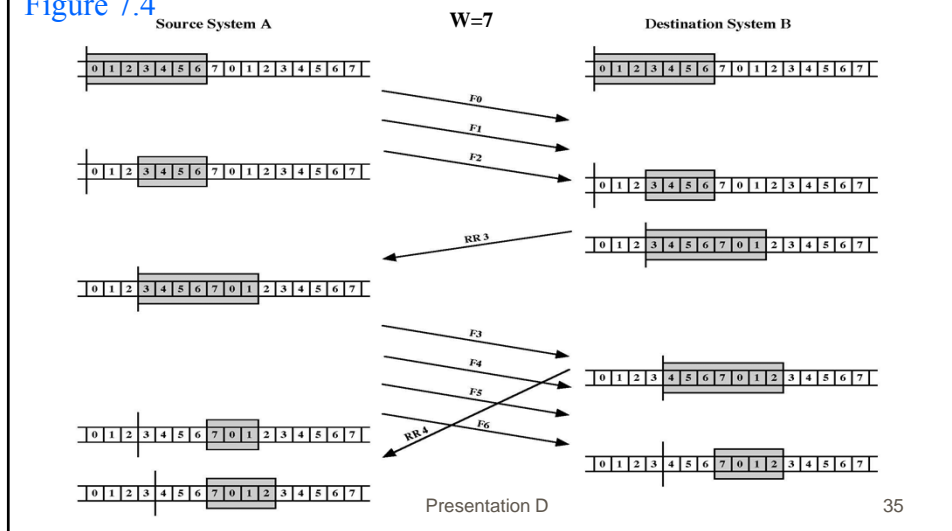
Presentation D

34

Sliding Window : Example

Note: Receiver may send RR immediately upon receiving an info-frame, or it may wait for more than one info-frame and then acknowledge all of them with only one RR.

Figure 7.4



35

Sliding Window Details (continued)

- Receiver can acknowledge frames without permitting further transmission with RNR control frame (Receive Not Ready)
 - RNR(x) acknowledges info-frames with sequence numbers x-1, and lower, but also closes a window
 - RR(x) must be sent to resume
- Note:** Normally each side is at the same time receiver and transmitter, and there is a separate window for each direction
 - Thus, RNR sender may still send info-frames, since window in another direction is not closed

g. babic

Presentation D

36

Sliding Window Control: Details (continued)

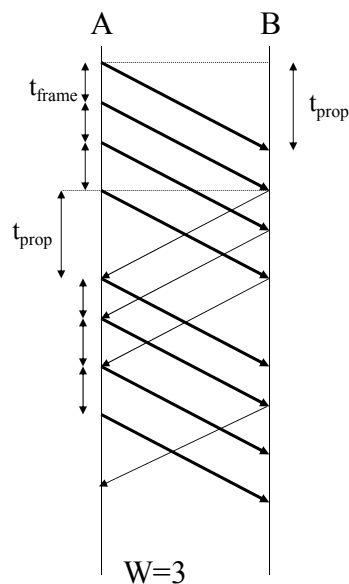
- If both sides are sending info-frames, use piggybacking, i.e. send acknowledgement in the header of info-frame
 - If no data to send, use acknowledgement frame
 - If data but no acknowledgement to send, send last acknowledgement number again
- It can be shown that for N-bit sequence number field, the largest $W=2^N-1$. For example, if $N=3$, then $W \leq 7$.
- Note: Stop and wait flow control is a special case of sliding-window flow control with $N=1$ and $W=1$.
- Note: In some protocols, W may change dynamically during a given session of message exchange.

g. babic

Presentation D

37

Sliding Window: Link Utilization



$t_{\text{proc}} = 0$ assumed

$$U = \frac{W \times t_{\text{frame}}}{2 \times t_{\text{prop}} + t_{\text{frame}}}$$

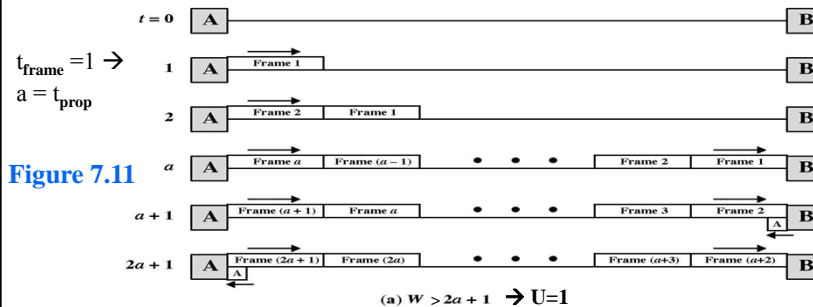
$$U = \begin{cases} W/(2a+1) & \text{if } W \leq 2a+1 \\ 1 & \text{if } W \geq 2a+1 \end{cases}$$

g. babic

Presentation D

38

Sliding Window: Link Utilization (continued)



g. babic

$$\rightarrow U=W/(2a+1)$$

39

Error Control

- **Error control** refers to mechanisms to detect and correct errors that occur in exchanging frames. Besides errors introduced by transmission system, it is possible that a complete frame gets lost when receiver buffers are full. Receiving DLC should be delivering to the upper layer received frames in the same order in which they are sent and without duplications.
- Error control is based on some of the following mechanisms:
 - error detection (and/or error correction)
 - sequence numbers for info-frames
 - positive acknowledgment
 - retransmission after time-out
 - negative acknowledgment and retransmission
- Error control mechanisms are referred as automatic repeat request – ARQ and we study three ARQs.

g. babic

Presentation D

40

Stop and Wait ARQ

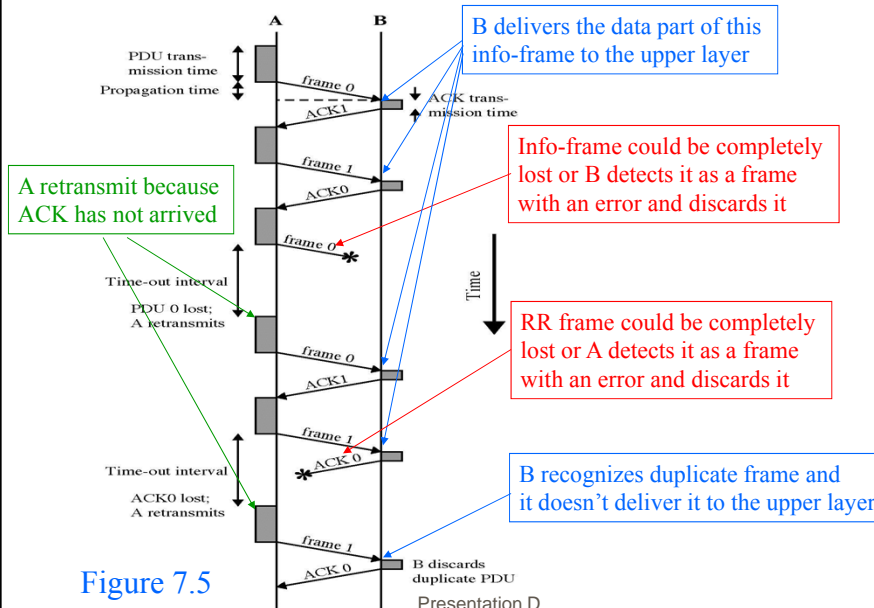
- Sender transmits only one unacknowledged info-frame, i.e. $W=1$
- Sender waits for RR, i.e. ACK, before sending the next info-frame
- In principle, if a received frame is damaged, receiver simple discards it
- If info-frame damaged
 - Sender has timeout for each info-frame
 - If no ACK within timeout, retransmit
- If ACK damaged, info-frame sender will not recognize it
 - Sender will retransmit info-frame
 - Receiver gets two copies of the same info-frame
- Thus, info-frames still has to be numbered
- Use ACK0 and ACK1, i.e. RR with one bit field

g. babic

Presentation D

41

Stop and Wait ARQ: Example



g. babic

Presentation D

42

Stop and Wait ARQ: Summary

- This mechanism uses sequence numbers for info-frames (1-bit send sequence number field in a header of info-frames), positive acknowledgments (1-bit receive sequence number field), info-frame retransmission after time-out and stop and wait flow control ($W=1$).
- How to determine a length of a time out? It is usually 2-4 round trip times. More advanced implementations may change it in time, if a round trip time changes.
- How many retransmission attempts?
- If after a maximum number of retransmissions, a receiver still does not respond, a sender assumes that data link is broken, i.e. non-operational.
- Simple to implement but inefficient
- Link utilization $U = (1-P)/(2a+1)$, where P is probability of frame in error

g. babic

Presentation D

43

Go-back-N ARQ

- This mechanism is based on sliding window and uses:
 - sequence numbers, N-bit send sequence number field in a header of info-frames,
 - window size $W \leq 2^N-1$ for Sliding-Window Flow Control,
 - positive acknowledgments, RR control frames with N-bit receive sequence number field,
 - **after info-frame time out, send RR with P bit set and after a response info-frame retransmission possible**
 - **negative acknowledgment, REJ control frame with N-bit receive sequence number field,**
 - piggybacking, info-frames include N-bit receive sequence number field, thus if there are info-frames to send no need for RR frames to be responded with.

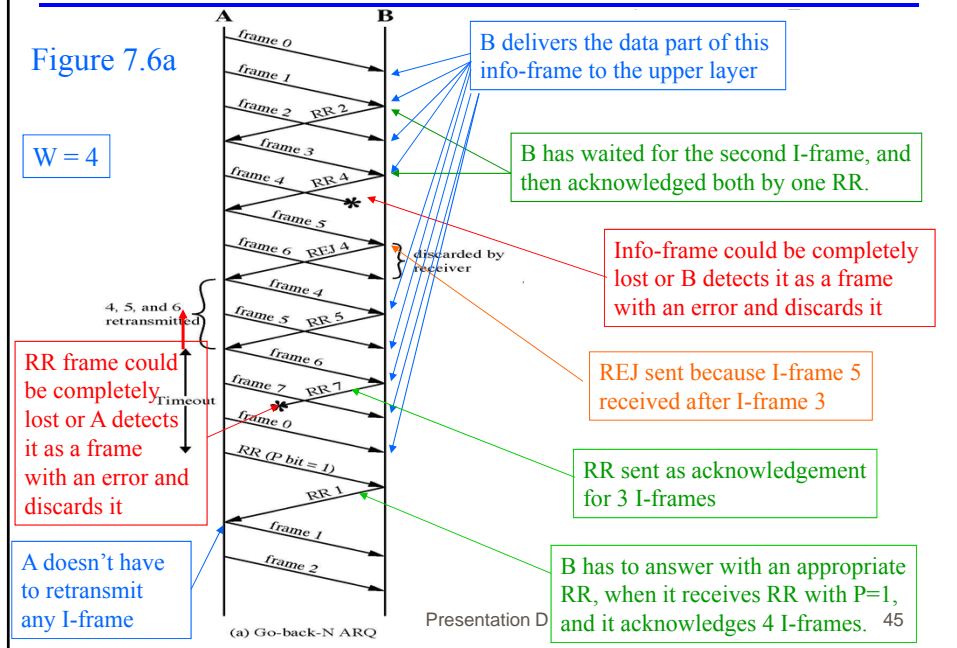
g. babic

Presentation D

44

Go-Back-N ARQ: Example

Figure 7.6a



Go-back-N ARQ: Details

- Info-frame i lost or damaged and transmitter sends info-frame $i+1$:
 - Receiver gets info-frame $i+1$ out of sequence
 - Discard that frame and all future frames until missing info-frame is received correctly,
 - Receiver send reject frame REJ(i),
 - Transmitter goes back to info-frame i and retransmits it plus all subsequent.
- Info-frame i lost and no additional frame sent:
 - Receiver gets nothing and returns neither acknowledgement nor rejection,
 - Transmitter times out and sends RR frame with P bit set
 - Receiver interprets this as command and it acknowledges with RR frame with the number of the next info-frame it expects (frame i),
 - Transmitter then retransmits frame i .

Go-back-N ARQ: Details (continued)

- Receiver gets info-frame i and sends acknowledgement $RR(i+1)$ which is lost or damaged:
 - Since acknowledgements are cumulative, next acknowledgement $RR(i+n)$ may arrive before transmitter times out on frame i ,
 - If transmitter times out, it sends acknowledgement RR with P bit set,
 - Receiver interprets this RR frame as command which it acknowledges sending an RR with the number of the next info-frame it expects (frame $i+1$),
- Only one REJ frame is sent and if REJ frame lost, same as the case Info-frame i lost and no additional frame sent.

g. babic

Presentation D

47

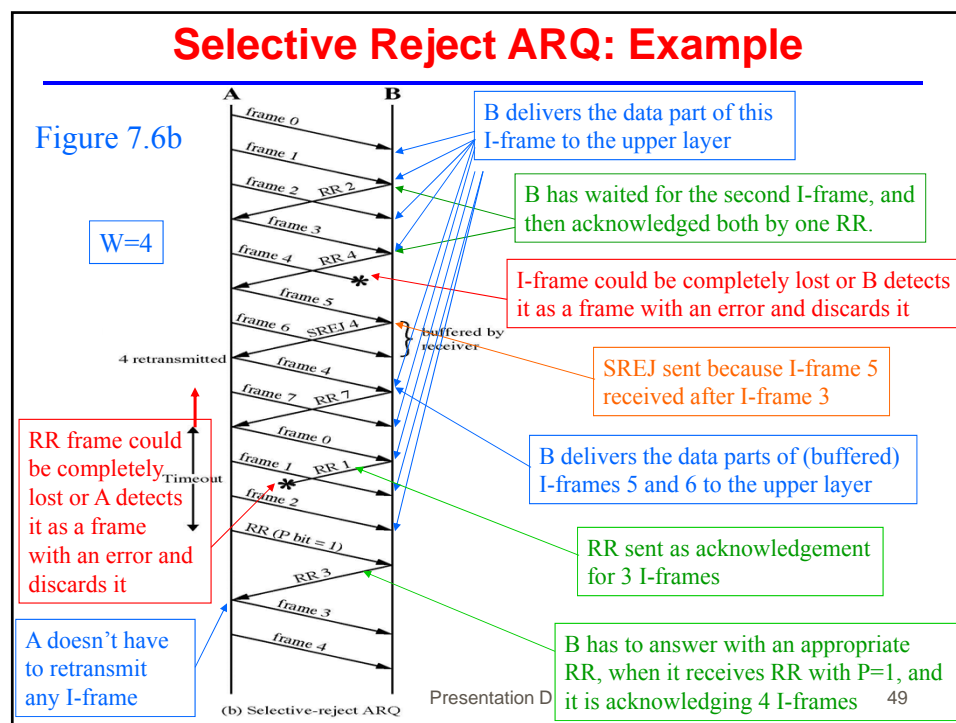
Selective Reject ARQ

- Similar to Go-back-N ARQ, except:
 - $SREJ$ used instead REJ
 - Only rejected frames are retransmitted
 - Subsequent frames are accepted by the receiver and buffered
- Minimizes retransmission
- But receiver must maintain large enough buffer
- Also, more complex at transmitter
- Link utilization
 - $U = W \times (1-P) / (2a+1)$ if $W \leq 2a+1$
 - $= 1-P$ if $W \geq 2a+1$
 - (P is probability of frame in error)

g. babic

Presentation D

48



Selective Reject ARQ and Window Size

- Window size limitation is more restrictive for selective-reject than for go-back-N.
- For N-bit sequence number field, go-back-N ARQ can have as the largest possible $W=2^N-1$. For example, if $N=3$, then $W \leq 7$.
- For N-bit sequence number field, selective-reject ARQ can have as the largest possible $W=2^{N-1}$. For example, if $N=3$, then $W \leq 4$.
- On the bottom of the page 221, there is a scenario that is suppose to prove necessity for this more restrictive W than for Go-back-N ARQ.
- Why is the scenario not correct?

High Level Data Link Control - HDLC

- HDLC uses:
 - Go-back-N and Selective reject ARQ
 - Flow and error control piggybacked on information frames
- HDLC station can be one of several types:
 - Combined station may issue commands and responses and that is most widely used.
- HDLC link can be in one of several configurations:
 - Balanced with two combined stations is most widely used.
- Used normally in synchronous transmission
- Single frame format for all data and control exchanges

g. babic

Presentation D

51

HDLC Frame Structure

- Flag = 01111110 delimits frame at both ends
- May close one frame and open another
- Receiver hunts for flag sequence to synchronize
- Bit stuffing used to avoid confusion with data containing bit sequence 01111110
- Address field, usually 8 bits long
- Frames with the address of sender are responses
- Frames with the address of receiver are commands



(a) Frame format Figure 7.7

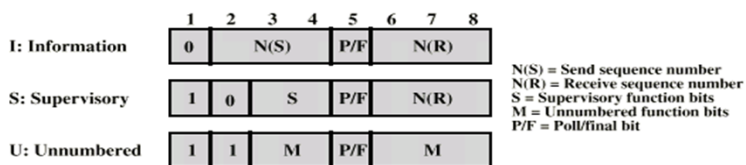
g. babic

Presentation D

52

HDLC Frame Structure (continued)

- Control field, 8 (usually) or 16 bits long
 - Defines frame type: I-frame, S-frames (RR, RNR, REJ and SREJ) and U-frames (SABM, UA, DISC, and many more)
- Info field, variable length
 - found only in I-frames (and some U-frames)
- Frame Check Sequence (FCS) field, usually 16 bits
 - Error detection with CRC



(c) 8-bit control field format

Figure 7.7

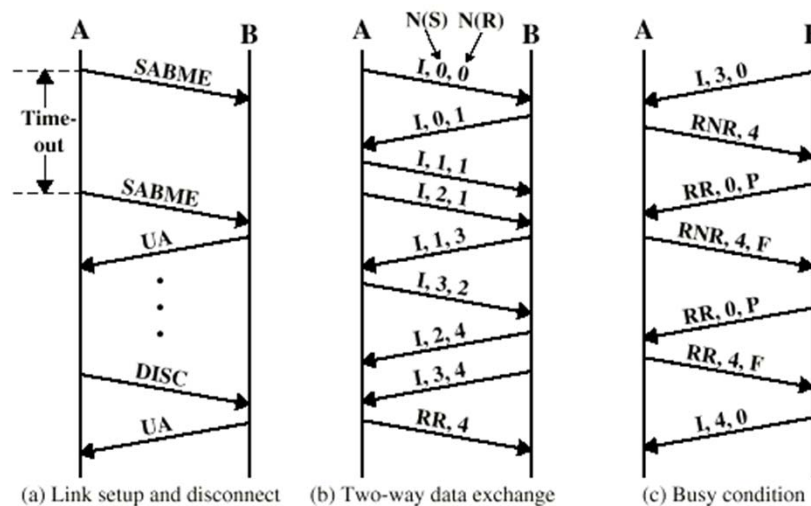
g. babic

Presentation D

53

HDLC Operations

- Three phases: Initialization (link setup), data transfer and disconnect



(a) Link setup and disconnect

(b) Two-way data exchange

(c) Busy condition

Figure 7.9

g. babic

Presentation D

54

HDLC Operations (continued)

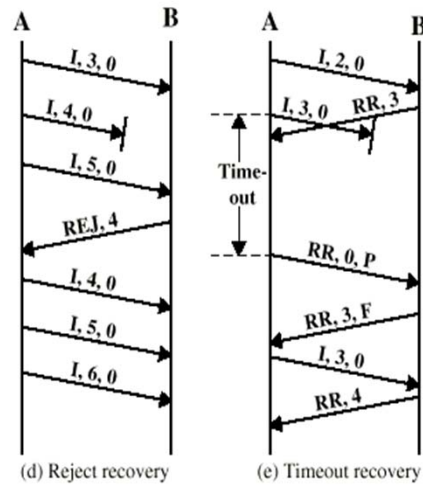


Figure 7.9

g. babic

Presentation D

55