Data and Computer Communications

Chapter 7 – Data Link Control Protocols

Eighth Edition

by William Stallings

Lecture slides by Lawrie Brown

Data Link Control Protocols

"Great and enlightened one," said Ten-teh, as soon as his stupor was lifted, "has this person delivered his message competently, for his mind was still a seared vision of snow and sand and perchance his tongue has stumbled?"

"Bend your ears to the wall," replied the Emperor, "and be assured."

*—Kai Lung's Golden Hours,* Earnest Bramah

Data Link Control Protocols

➤ need layer of logic above Physical

➤ to manage exchange of data over a link
  - frame synchronization
  - flow control
  - error control
  - addressing
  - control and data
  - link management

Flow Control

➢ ensure sending entity does not overwhelm receiving entity
  • by preventing buffer overflow
➢ influenced by:
  • transmission time
    • time taken to emit all bits into medium
  • propagation time
    • time for a bit to traverse the link
➢ assume here no errors but varying delays

Model of Frame Transmission

(a) Error-free transmission

(b) Transmission with losses and errors
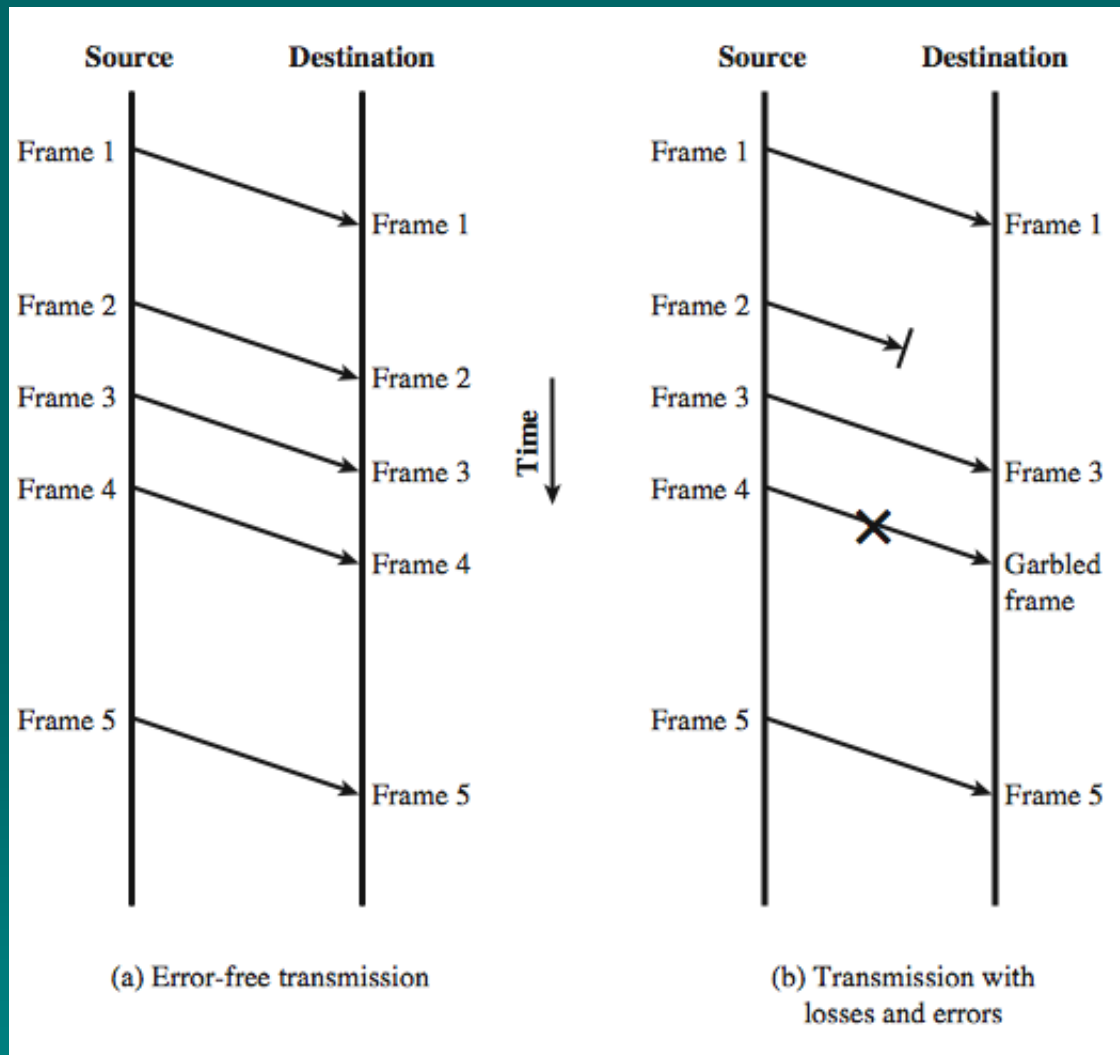
Stop and Wait

➢ source transmits frame (no need for seq. #)

➢ destination receives frame and replies with acknowledgement (ACK)

➢ source waits for ACK before sending next

➢ destination can stop flow by not send ACK

➢ works well for a few large frames

➢ Stop and wait becomes inadequate if large block of data is split into small frames

Stop and Wait Link Utilization

(a) $a < 1$

(b) $a > 1$
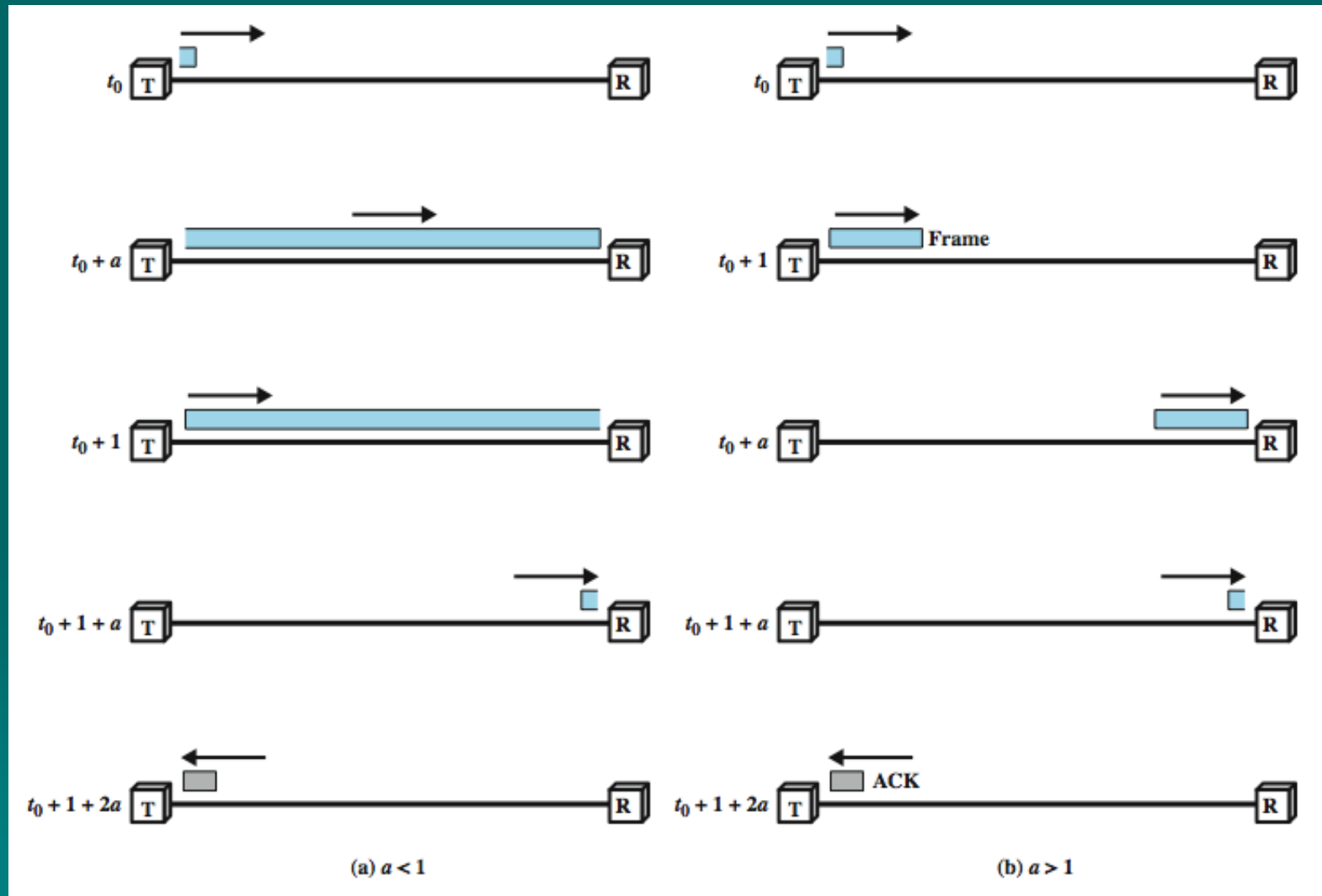
Stop and Wait Utilization (cont)

- *B = R x (d/V)*

  - **B**: length of the link in bits

  - **R**: data rate of the link, in bps

  - **d**: length or distance of the link in meters

  - **D**: velocity of propagation, in m/s

- *a = B/L*

  - **a**: the propagation delay (with frame tx time = 1)
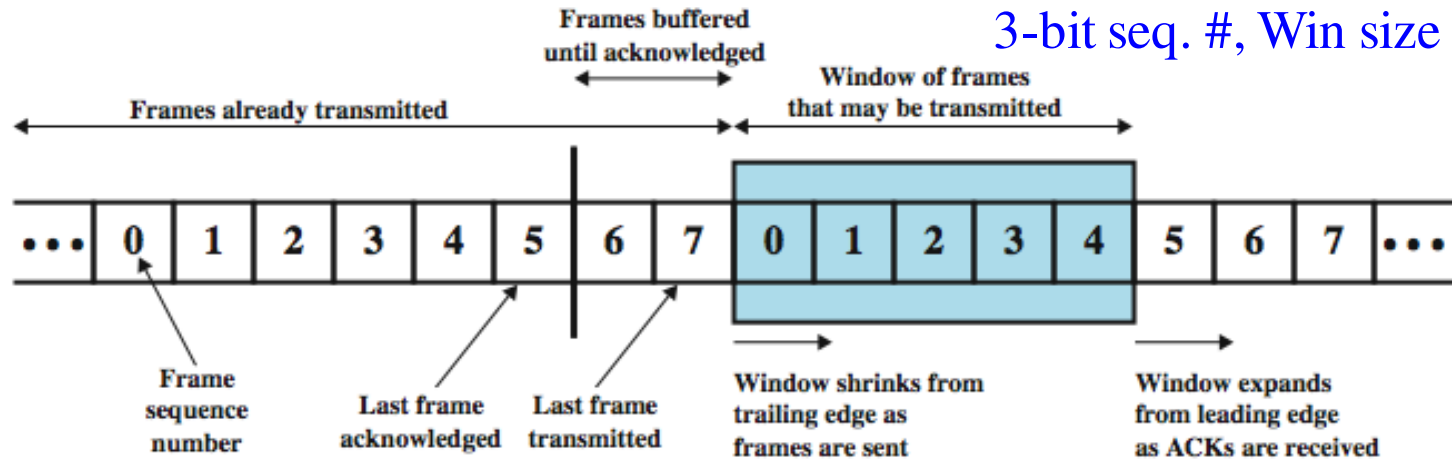
  - **L**: the number of bits in the frame

- **or** $$a = (\frac{d}{V}) \, / \, (\frac{L}{R})$$
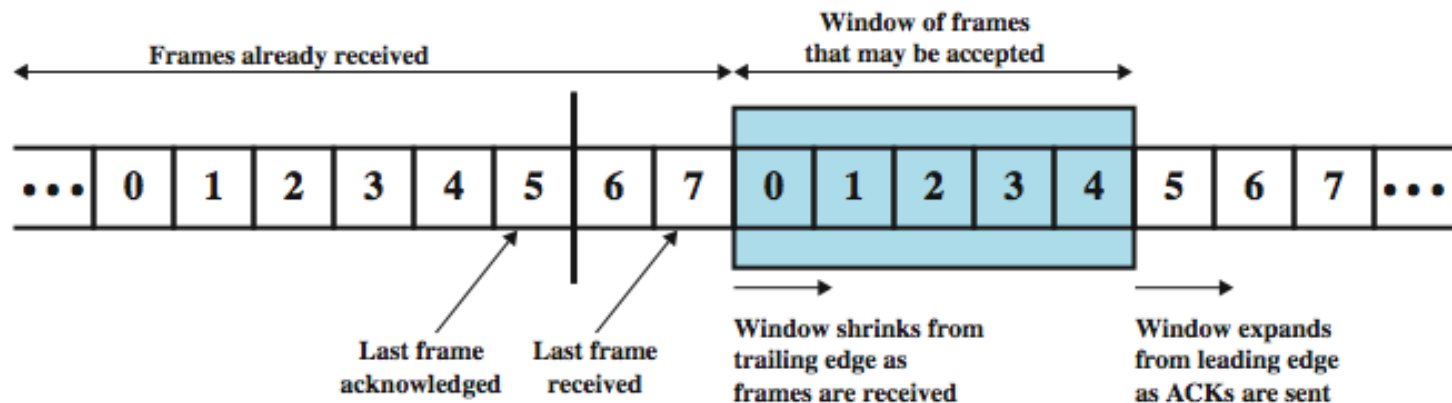
Sliding Windows Flow Control

➢ allows multiple numbered frames to be in transit
➢ receiver has buffer W long
➢ transmitter sends up to W frames without ACK
➢ ACK includes number of next frame expected
➢ sequence number is bounded by size of field (k)
  • frames are numbered modulo 2^k
  • giving max window size of up to $2^k - 1$ (why?)
➢ receiver can ack frames without permitting further transmission (Receive Not Ready)
➢ must send a normal acknowledge to resume
➢ if have full-duplex link, can piggyback ACKs

Sliding Window Diagram
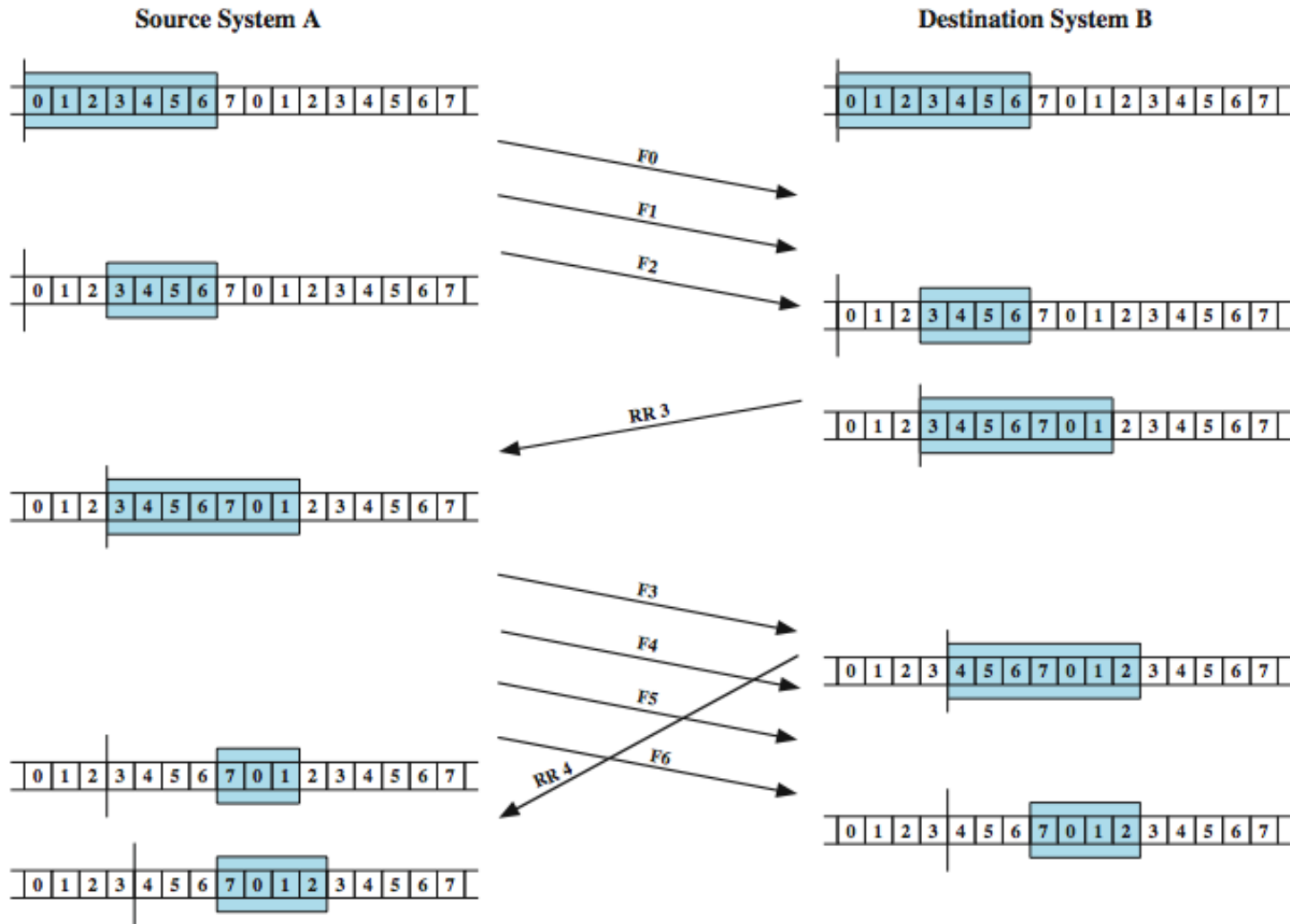

3-bit seq. #, Win size = 7

Frames buffered until acknowledged

Window of frames that may be transmitted

Frames already transmitted

... 0 1 2 3 4 5 6 7 | 0 1 2 3 4 | 5 6 7 ...

Frame sequence number

Last frame acknowledged

Last frame transmitted

Window shrinks from trailing edge as frames are sent

Window expands from leading edge as ACKs are received

(a) Sender's perspective

Window of frames that may be accepted

Frames already received

... 0 1 2 3 4 5 6 7 | 0 1 2 3 4 | 5 6 7 ...

Last frame acknowledged

Last frame received

Window shrinks from trailing edge as frames are received

Window expands from leading edge as ACKs are sent

(b) Receiver's perspective

Sliding Window Example

Sliding Window Utilization

➤ **Window size W, transmission time = 1, propagation time = a**

➤ **Case 1: W >= 2a + 1**

  ● **Sender A can transmit continuously with no pause and normalized throughput is 1.0**

➤ **Case 2: W < 2a + 1**

  ● **Sender A exhausts its window at t = W and cannot send additional frames until t = 2a + 1.**

  ● **Normalized throughput is W / (2a+1)**

Error Control

- detection and correction of errors such as:
  - lost frames
  - damaged frames
- common techniques use:
  - error detection
  - positive acknowledgment
  - retransmission after timeout
  - negative acknowledgement & retransmission

Automatic Repeat Request (ARQ)

➤ collective name for such error control mechanisms, including:

➤ stop and wait

➤ go back N

➤ selective reject (selective retransmission)

Stop and Wait

➤ source transmits single frame

➤ wait for ACK

➤ if received frame damaged, discard it
  - transmitter has timeout
  - if no ACK within timeout, retransmit

➤ if ACK damaged,transmitter will not recognize it
  - transmitter will retransmit
  - receive gets two copies of frame
  - use alternate numbering and ACK0 / ACK1

Stop and Wait

➢ see example with both types of errors

➢ pros and cons
- simple
- inefficient
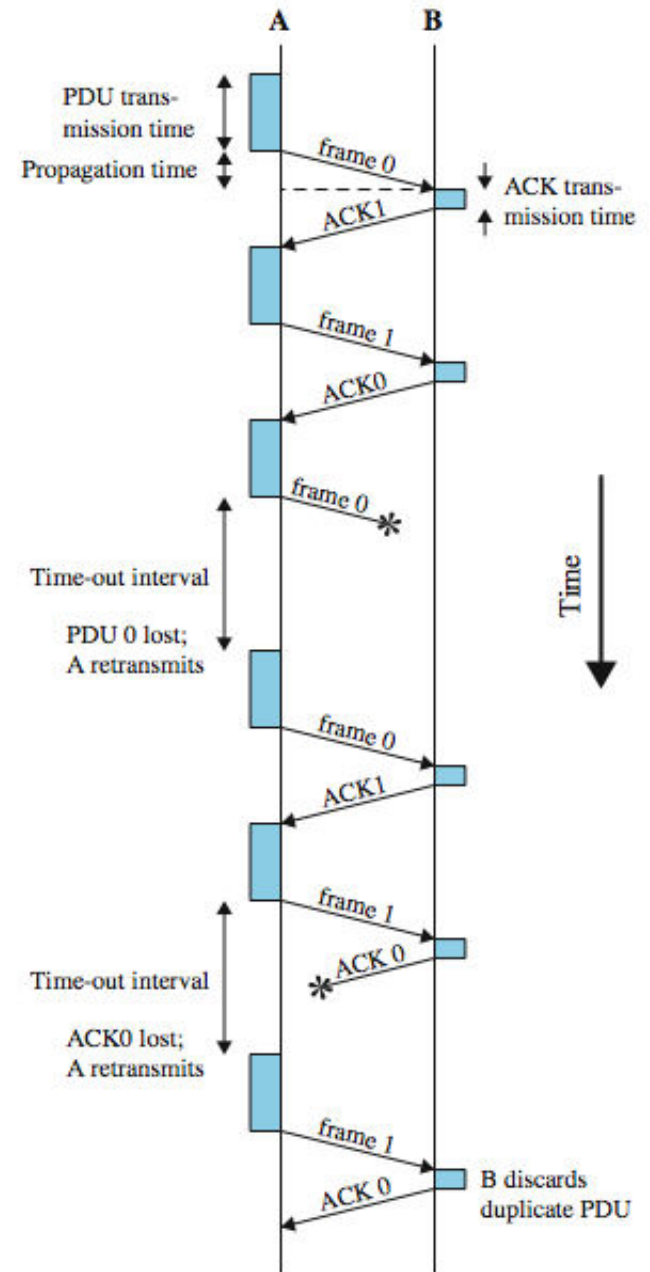
Go Back N

➢ based on sliding window

➢ if no error, ACK as usual

➢ 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

Go Back N - Handling

- Damaged Frame
  - error in frame *i* so receiver rejects frame *i*
  - transmitter retransmits frames from *i*
- Lost Frame
  - frame *i* lost and either
    - transmitter sends *i+1* and receiver gets frame *i+1* out of seq and rejects frame *i*
    - or transmitter times out and send ACK with P bit set which receiver responds to with ACK *i*
  - transmitter then retransmits frames from *i*

Go Back N - Handling

➢ Damaged Acknowledgement
- receiver gets frame *i*, sends ack (*i+1*) which is lost
- acks are cumulative, so next ack (*i+n*) may arrive before transmitter times out on frame *i*
- if transmitter times out, it sends ack with P bit set
- can be repeated a number of times before a reset procedure is initiated

➢ Damaged Rejection
- reject for damaged frame is lost
- handled as for lost frame when transmitter times out

Selective Reject

- also called selective retransmission
- only rejected frames are retransmitted
- subsequent frames are accepted by the receiver and buffered
- minimizes retransmission
- receiver must maintain large enough buffer
- more complex logic in transmitter
- hence less widely used
- useful for satellite links with long propagation delays

Go Back N vs Selective Reject

(a) Go-back-N ARQ

(b) Selective-reject ARQ
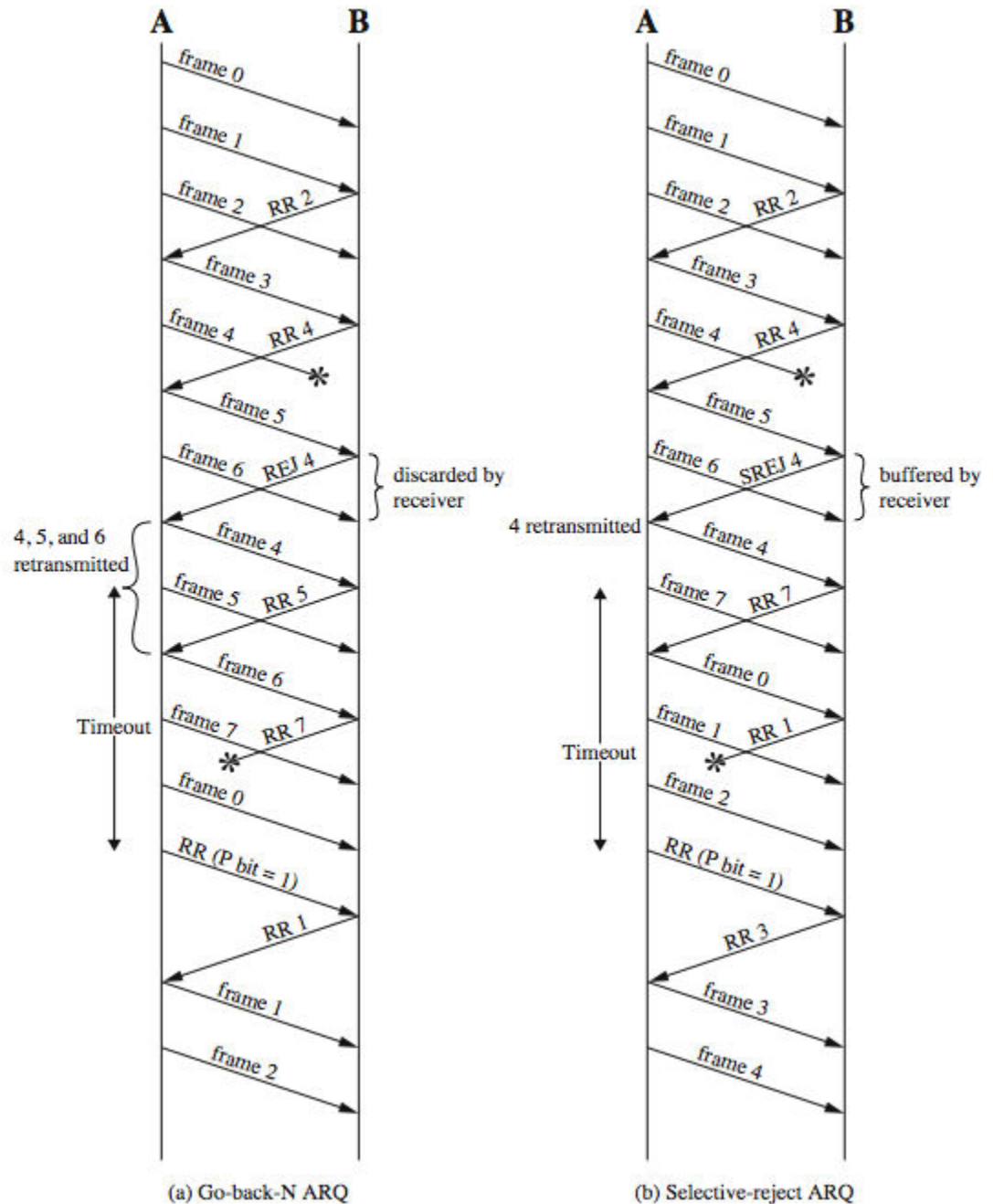
Selective-reject ARQ

➤ **Window-size limitation**

- **E.g. 3 bit sequence number, window size = 7**

- **Station A sends frames 0 ~ 6 to station B**

- **Station B receives all seven frames and** cumulatively acknowledges **with RR7**

- **If RR 7 is lost**

- **A times out and** retransmit frame 0 (w/o P-bit timer? Can P-bit mechanism be used in SR-ARQ?)

Selective-reject ARQ (cont)

- **B has already advanced its receive window to accept frames 7, 0, 1, 2, 3, 4, and 5. Thus, it assumes that frame 7 has been lost and that this is a new frame 0 which it accepts.**
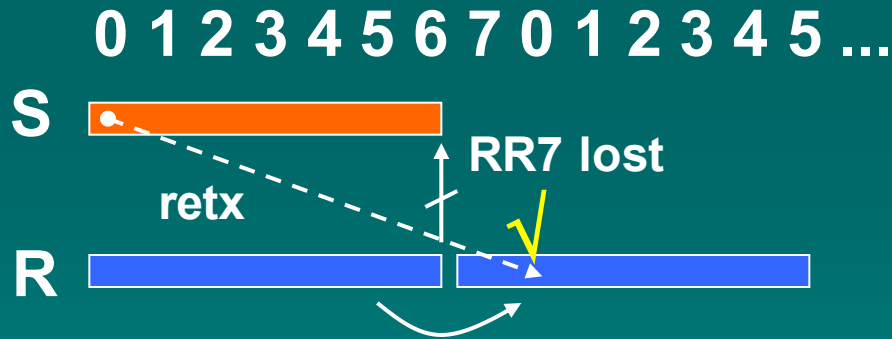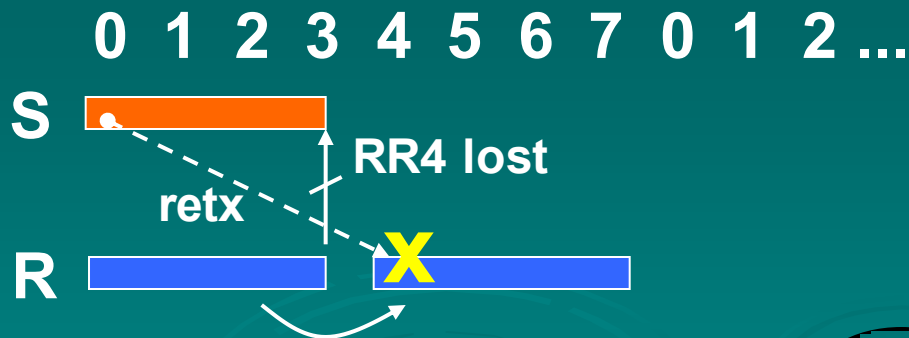
Selective-reject ARQ (cont)

➢ **Problem with the foregoing scenario**

- **overlay between the sending and receiving windows**

➢ **Solution**

- **for a k-bit sequence number, the maximum window size is limited to 2^{k-1}.**

0 1 2 3 4 5 6 7 0 1 2 ...

S

RR4 lost

retx

R

X

Performance of ARQ

> **Appendix 7A**

- **Stop-and-Wait ARQ** ☆ ☆ ☆

- **Selective Reject ARQ** ☆ ☆

- **Go-back-N ARQ** ☆

# High Level Data Link Control (HDLC)

- an important data link control protocol
- specified as ISO 33009, ISO 4335
- station types:
  - Primary - controls operation of link
  - Secondary - under control of primary station
  - Combined - issues commands and responses
- link configurations
  - Unbalanced - 1 primary, multiple secondary
  - Balanced - 2 combined stations

HDLC Transfer Modes

➢ Normal Response Mode (NRM)
  • unbalanced config, primary initiates transfer
  • used on multi-drop lines, eg host + terminals

➢ Asynchronous Balanced Mode (ABM)
  • balanced config, either station initiates transmission, has no polling overhead, widely used

➢ Asynchronous Response Mode (ARM)
  • unbalanced config, secondary may initiate transmit without permission from primary, rarely used

HDLC Frame Structure

➢ synchronous transmission of frames
➢ single frame format used

Flag	Address	Control	Information	FCS	Flag
8 bits	8 extendable	8 or 16	variable	16 or 32	8
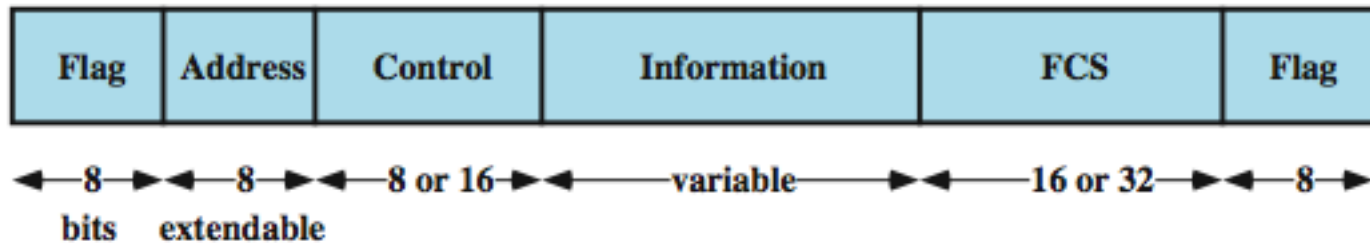
(a) Frame format

Flag Fields and Bit Stuffing

➢ delimit frame at both ends with 01111110 seq

➢ receiver hunts for flag sequence to synchronize

➢ bit stuffing used to avoid confusion with data containing flag seq 01111110

- 0 inserted after every sequence of five 1s
- if receiver detects five 1s it checks next bit
- if next bit is 0, it is deleted (was stuffed bit)
- if next bit is 1 and seventh bit is 0, accept as flag
- if sixth and seventh bits 1, sender is indicating abort
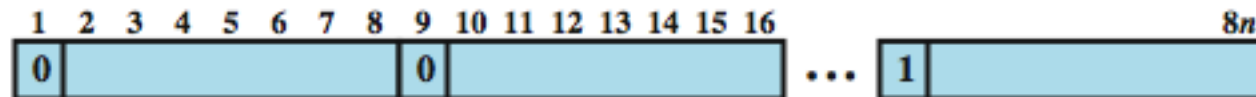
Original Pattern:

1111111111110111111101111110

After bit-stuffing

1111101111101101111101011111010

Address Field

➢ identifies secondary station that sent or will receive frame

➢ usually 8 bits long

➢ may be extended to multiples of 7 bits
  ● LSB indicates if is the last octet (1) or not (0)

➢ all ones address 11111111 is broadcast

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	...	$8n$
0								0									1

(b) Extended Address Field

Control Field

- ➤ different for different frame type
  - Information - data transmitted to user (next layer up)
    - Flow and error control piggybacked on information frames
  - Supervisory - ARQ when piggyback not used
  - Unnumbered - supplementary link control
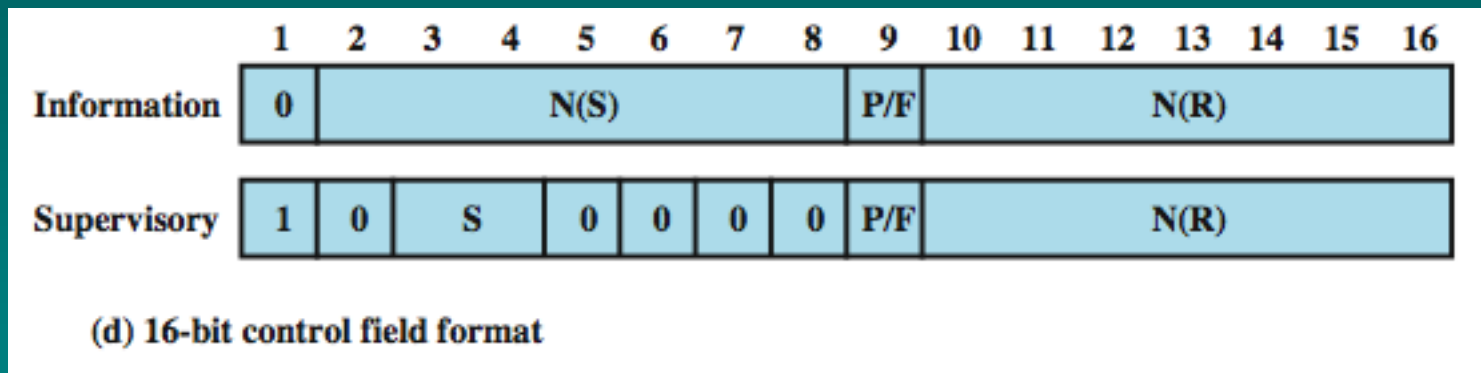- ➤ first 1-2 bits of control field identify frame type
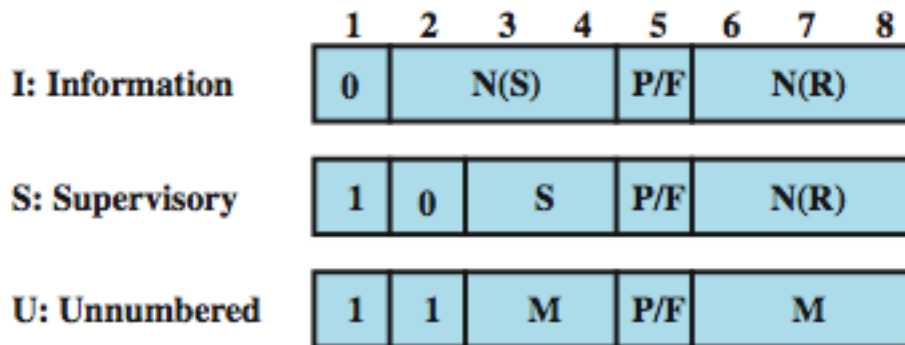
(c) 8-bit control field format

Control Field

- ➤ use of Poll/Final bit depends on context
- ➤ in command frame is P bit set to1 to solicit (poll) response from peer
- ➤ in response frame is F bit set to 1 to indicate response to soliciting command
- ➤ seq number usually 3 bits
  - can extend to 8 bits as shown below

	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)			

(d) 16-bit control field format

Information & FCS Fields

➤ Information Field
- in information and some unnumbered frames
- must contain integral number of octets
- variable length

➤ Frame Check Sequence Field (FCS)
- used for error detection
- either 16 bit CRC or 32 bit CRC

HDLC Operation

➤ consists of exchange of information, supervisory and unnumbered frames

➤ have three phases

- initialization

  - by either side, set mode & seq

- data transfer

  - with flow and error control
  - using both I & S-frames (RR, RNR, REJ, SREJ)

- disconnect

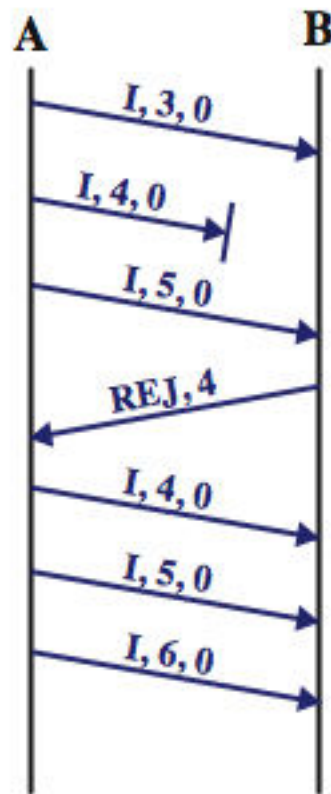 - when requested or fault noted

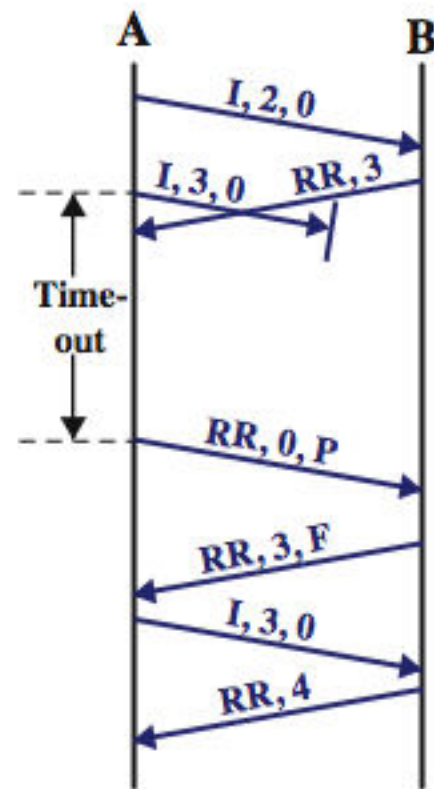# HDLC Operation Example
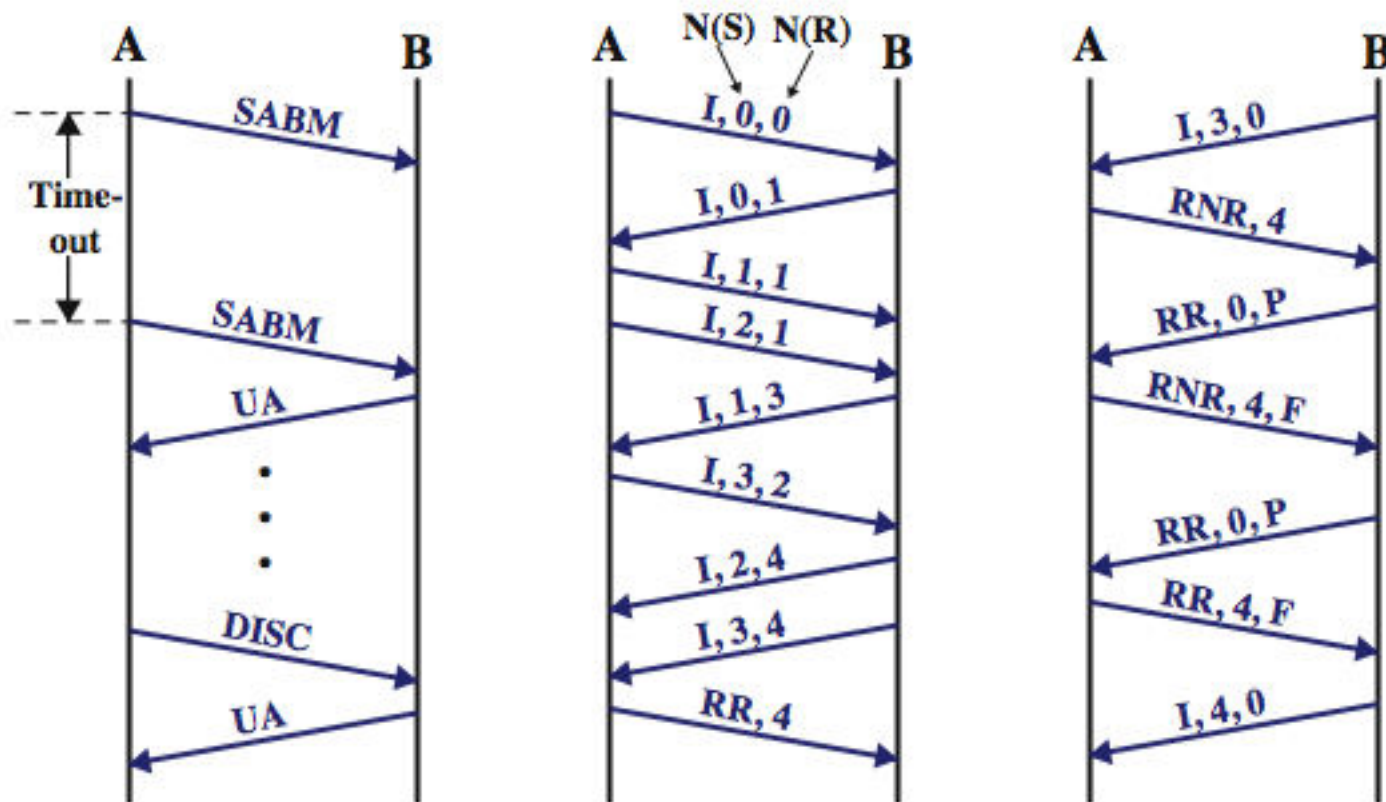
(a) Link setup and disconnect   (b) Two-way data exchange   (c) Busy condition

HDLC Operation Example

(d) Reject recovery

(e) Timeout recovery

Summary

➢ introduced need for data link protocols

➢ flow control

➢ error control

➢ HDLC