Chapter 3: Data Link layer

Lecturer: Lam Nhut Khang

9/2020

Slides are adapted from

[1] Computer Networks – An Open Source Approach. Ying-dar Lin, Ren-hung Hwang, Fred Baker

[2]Computer Networking: A Top-Down Approach. 8th Edition. Jim Kurose, Keith Ross, Pearson, 2020

[3] Sami Rollins, Computer network's slides, University of San Francisco, www.cs.usfca.edu

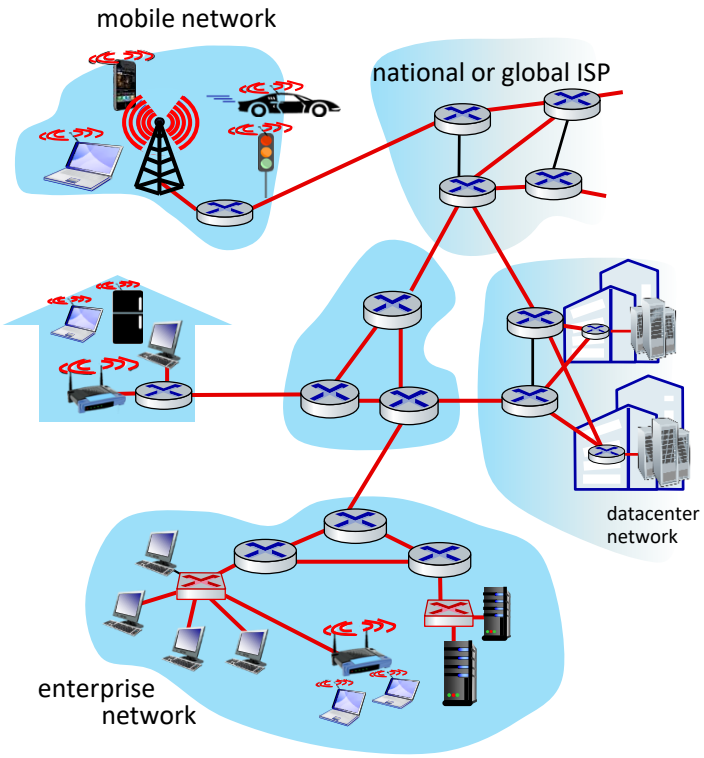
[4]Ajit Pal, CSE IIT, Kharagpur https://nptel.ac.in/course.html

Link layer: introduction

terminology:

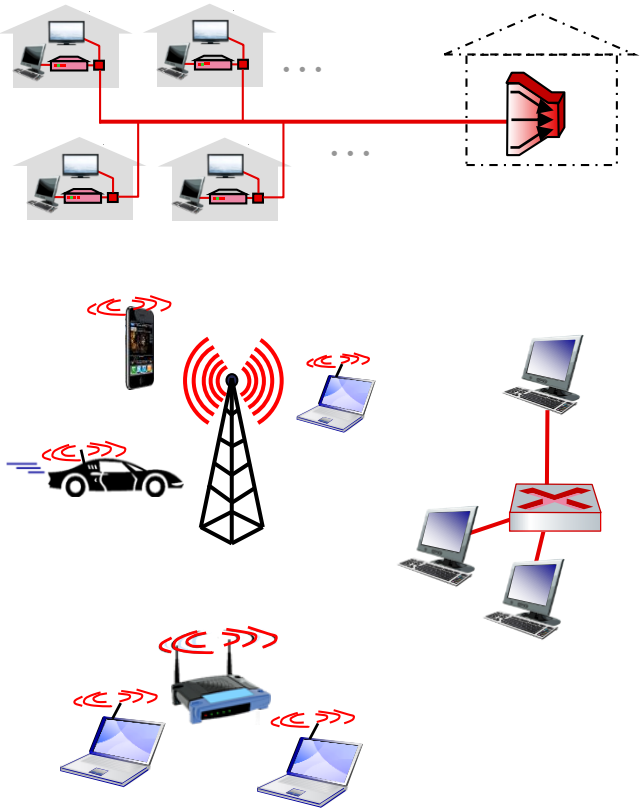
- hosts and routers: nodes
- communication channels that connect adjacent nodes along communication path: links
  - wired
  - wireless
  - LANs
- layer-2 packet: *frame*, encapsulates datagram

link layer has responsibility of transferring datagram from one node to physically adjacent node over a link



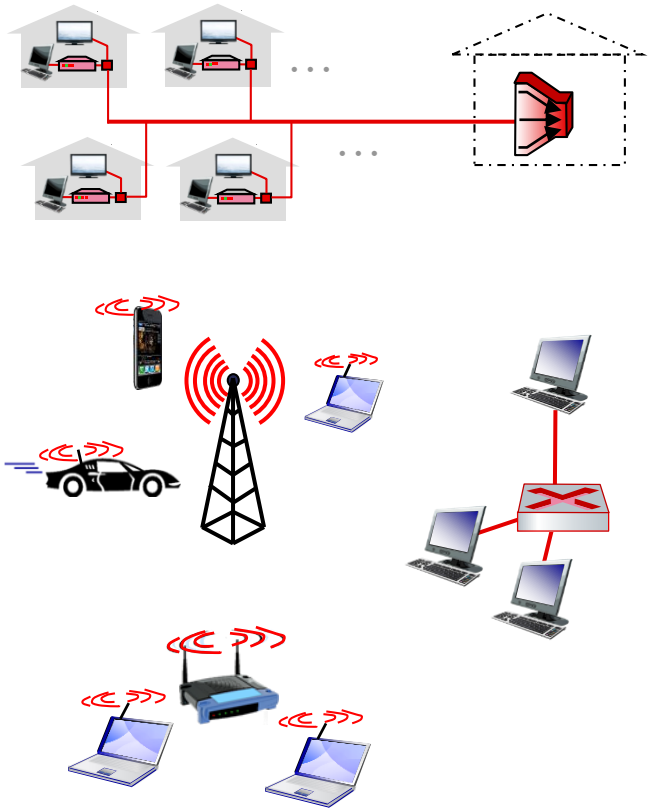
Link layer: services

- framing, link access:
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - "MAC" addresses in frame headers identify source, destination (different from IP address!)
- reliable delivery between adjacent nodes
  - we already know how to do this!
  - seldom used on low bit-error links
  - wireless links: high error rates
    - *Q:* why both link-level and end-end reliability?



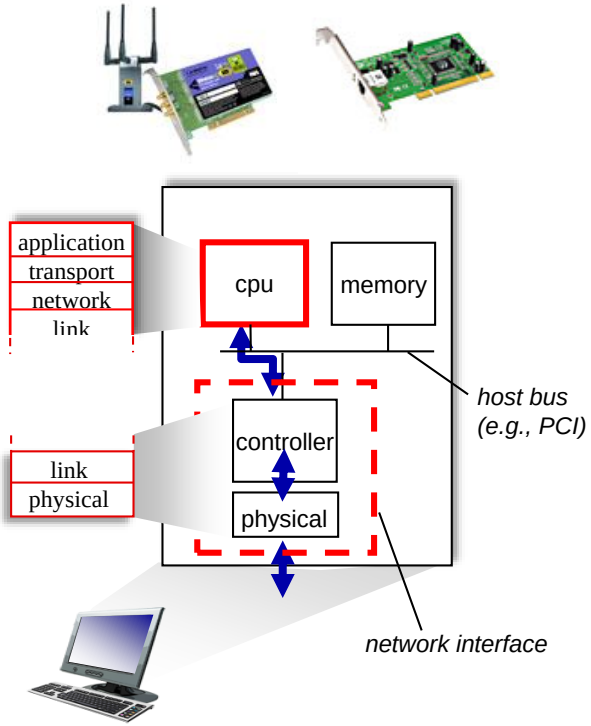
Link layer: services (more)

- flow control:
  - pacing between adjacent sending and receiving nodes
- error detection:
  - errors caused by signal attenuation, noise.
  - receiver detects errors, signals retransmission, or drops frame
- error correction:
  - receiver identifies *and corrects* bit error(s) without retransmission
- half-duplex and full-duplex:
  - with half duplex, nodes at both ends of link can transmit, but not at same time

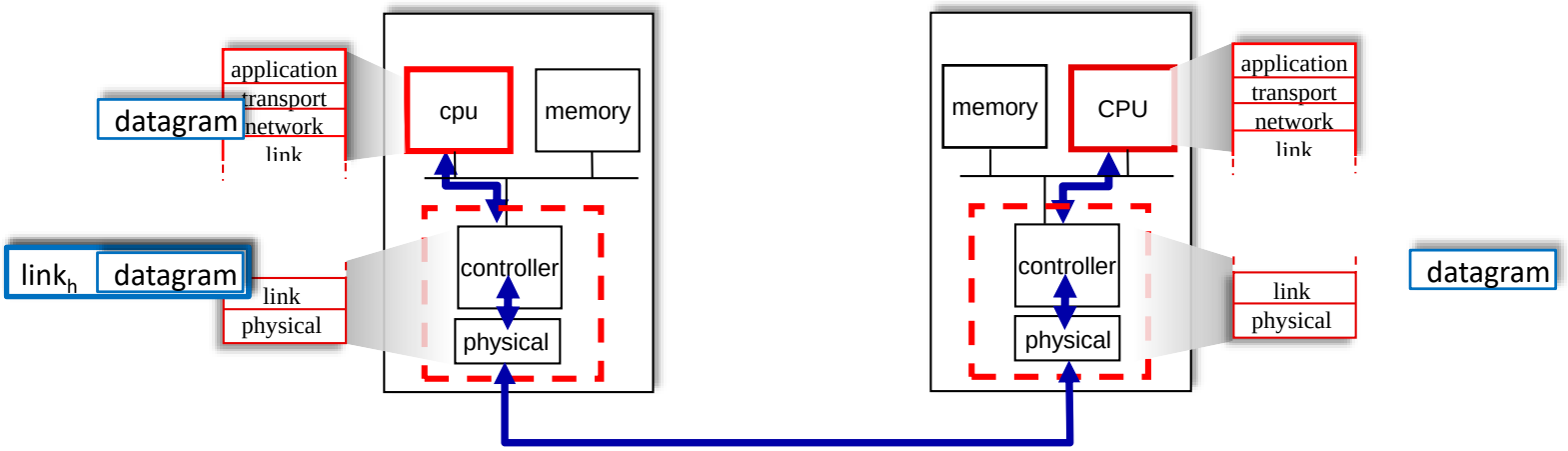


Where is the link layer implemented?

- in each-and-every host
- link layer implemented in *network interface card* (NIC) or on a chip
  - Ethernet, WiFi card or chip
  - implements link, physical layer
- attaches into host's system buses
- combination of hardware, software, firmware



Interfaces communicating



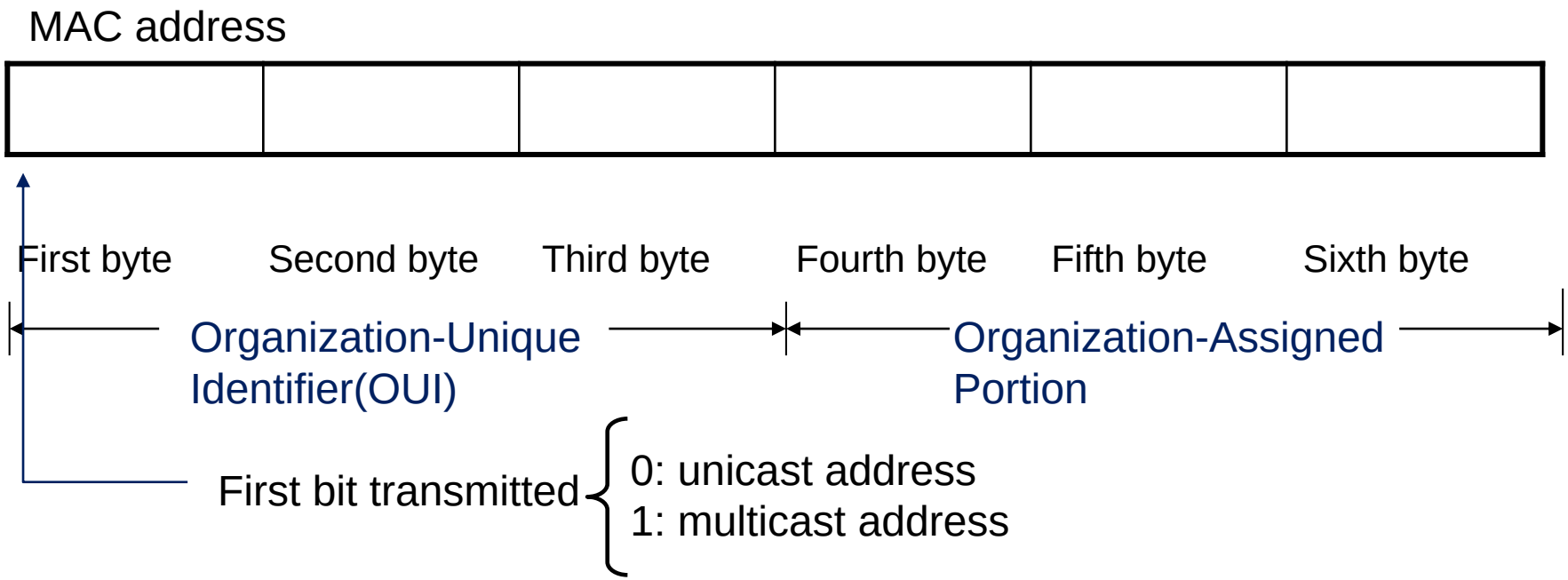
sending side:

- encapsulates datagram in frame
- adds error checking bits, reliable data transfer, flow control, etc.

receiving side:

- looks for errors, reliable data transfer, flow control, etc.
- extracts datagram, passes to upper layer at receiving side

IEEE 802 MAC Address



Transmission order of bits in each byte

Little-Endian: e.g., Ethernet

Big-Endian: e.g., FDDI, Token Ring

Framing

Framing

Typical fields in the frame format

- address
- length
- type of upper layer protocol
- payload
- error detection code

Basic unit of a frame

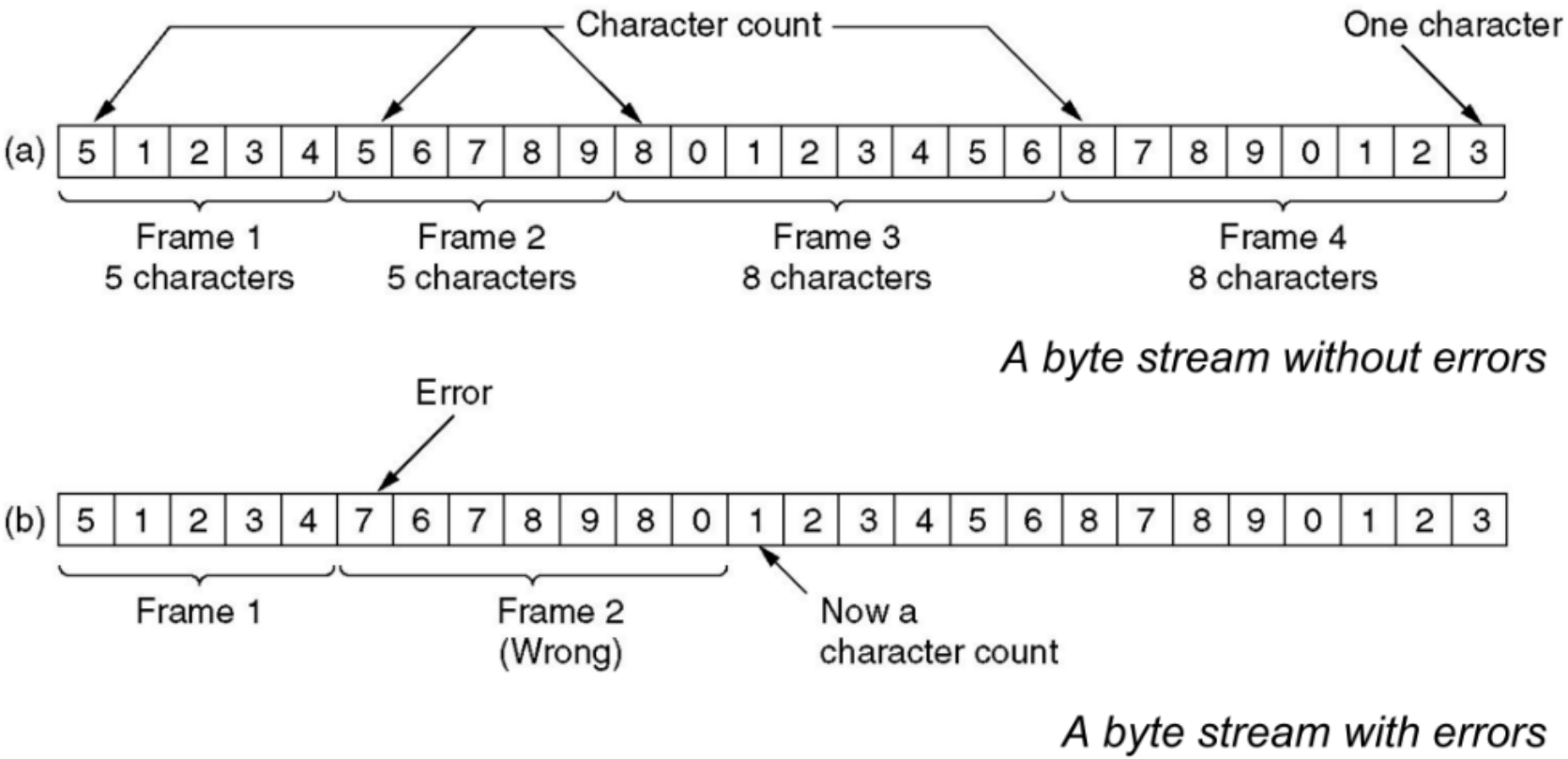
- byte (e.g., Ethernet frame) → *byte-oriented*
- bit (e.g., HDLC frame) → *bit-oriented*

Framing methods

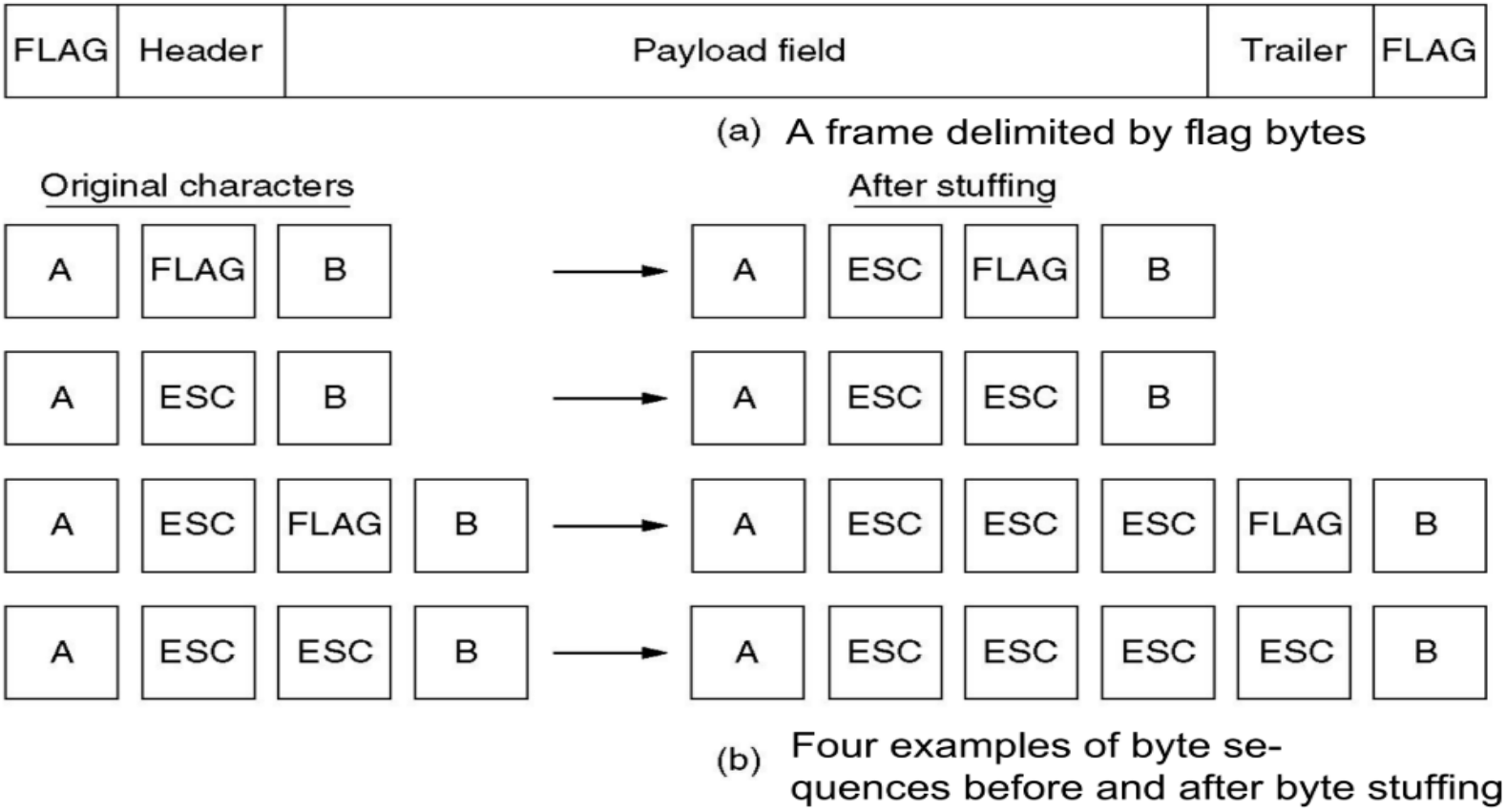
- Counting
- Flag bytes with byte stuffing
- Flag bits with bit stuffing

Note: Bit (or byte) stuffing to avoid ambiguity

Framing methods
counting



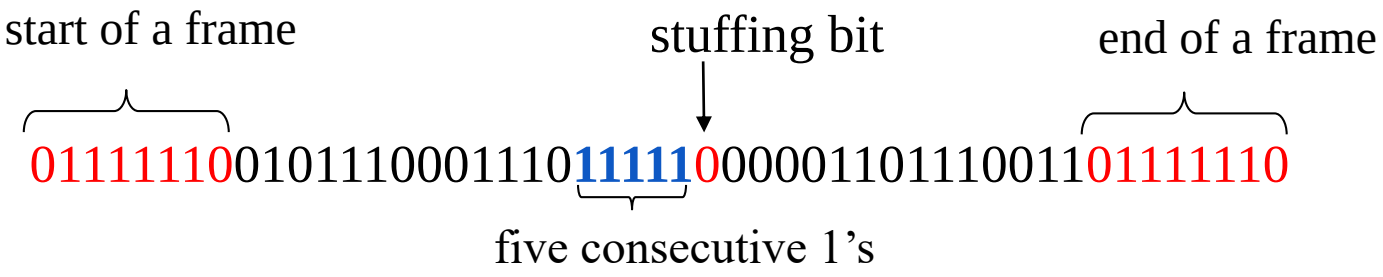
Framing methods
Flag bytes with byte stuffing



Framing methods
Flag bits with bit stuffing

Using Special bit pattern

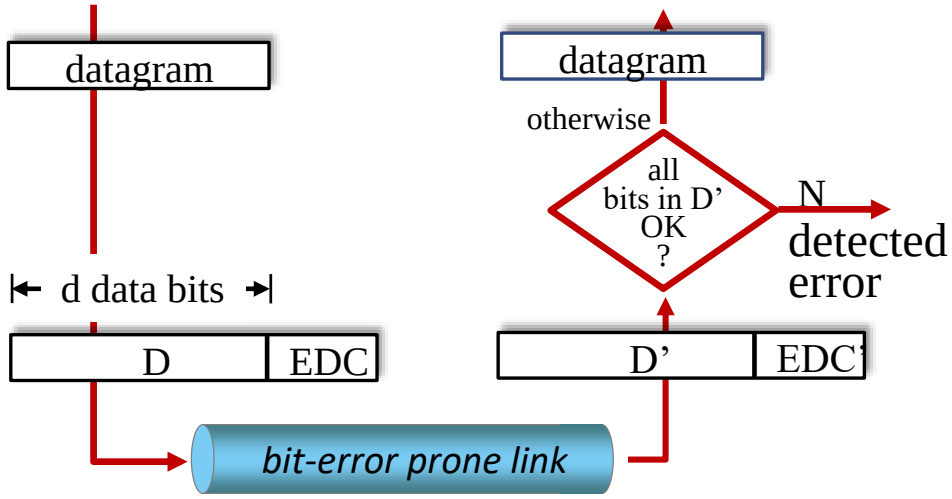
e.g. a bit pattern 01111110



Error detection

Error detection

EDC: error detection and correction bits (e.g., redundancy)
D: data protected by error checking, may include header fields



Error detection not 100% reliable!

- protocol may miss some errors, but rarely
- larger EDC field yields better detection and correction

Error Detection Code

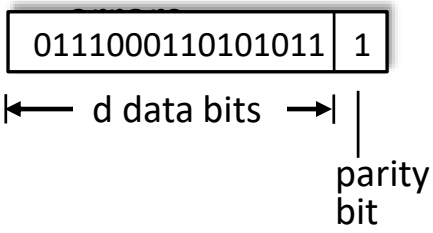
Error detection code:

- Parity check
- Checksum
- Cyclic Redundancy Check (CRC)

Parity checking

single bit parity:

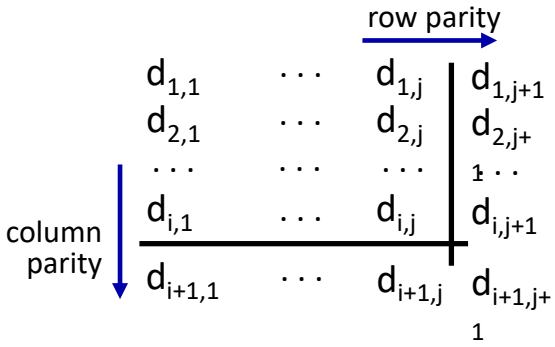
- detect single bit



Even parity: set parity bit so there is an even number of 1's

two-dimensional bit parity:

- detect *and correct* single bit errors



no errors:

1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
1	0	1	0	1	0

detected and correctable single-bit error:

1	0	1	0	1	1
1	0	1	1	0	0
0	1	1	1	0	1
1	0	1	0	1	0

parity error

parity error

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

Error Detection Code
Checksum

Checksum

- Transmitter: add all words and transmit the sum
- Receiver: add all words and check the sum

Cyclic Redundancy Check (CRC)

- Transmitter: Generate a bit sequence by modulo 2 division
- Receiver: Divide the incoming frame and check if no remainder

CRC for link layer and checksum for IP/TCP/UDP

- CRC: easy implementation in hardware, but not in software; more robust to errors
- Checksum: just a double-check against nodal errors

Error Detection Code
Checksum

Checksum

- Transmitter: add all words and transmit the sum
- Receiver: add all words and check the sum

Sender side

- Divide data into equal segments of *n* bits
- Add all segments
- The sum is complemented using 1's complement arithmetic and appended at the end of original data

Receiver side

- Divide received data into equal segments of *n* bits
- Add all segments
- The sum is complemented using 1's complement arithmetic.
  - The sum = 0 → no error occurred → accept data
  - The sum != 0 → error occurred → discard data

Error Detection Code
Checksum

Original data	10011001111000100010010010000100			
	10011001	11100010	00100100	10000100

Sender	10011001	11100010	00100100	10000100	11011010

Checksum value

10011001
11100010
101111011
1
01111100
00100100
10100000
10000100
100100100
1
00100101
11011010

1's complement

Error Detection Code
Checksum

Original data	10011001 11100010 00100100 10000100			
	10011001	11100010	00100100	10000100

Received data	10011001	11100010	00100100	10000100	11011010

10011001
11100010
101111011
1
01111100
00100100
10100000
10000100
100100100
1
00100101
11011010
11111111
1's complement → 00000000

no error occurred

Error Detection Code
Cyclic Redundancy Check (CRC)

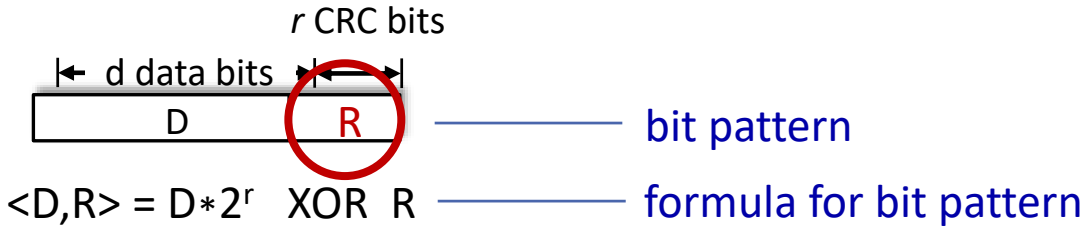
Frame content: 11010001110(11 bits)

Pattern: 101011 (6 bits)

Frame check sequence = (5 bits)

Cyclic Redundancy Check (CRC)

- more powerful error-detection coding
- D: data bits (given, think of these as a binary number)
- G: bit pattern (generator), of *r+1* bits (given)



goal: choose *r* CRC bits, R, such that <D,R> exactly divisible by G (mod 2)

- receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!
- can detect all burst errors less than r+1 bits
- widely used in practice (Ethernet, 802.11 WiFi)

Cyclic Redundancy Check: example

We want:

$D \cdot 2^r \ XOR \ R = nG$

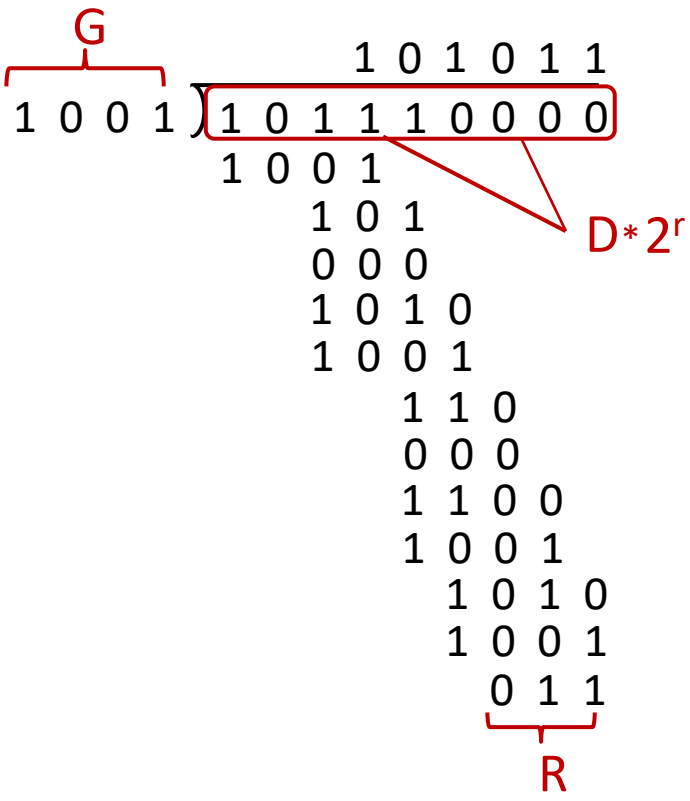
or equivalently:

$D \cdot 2^r = nG \ XOR \ R$

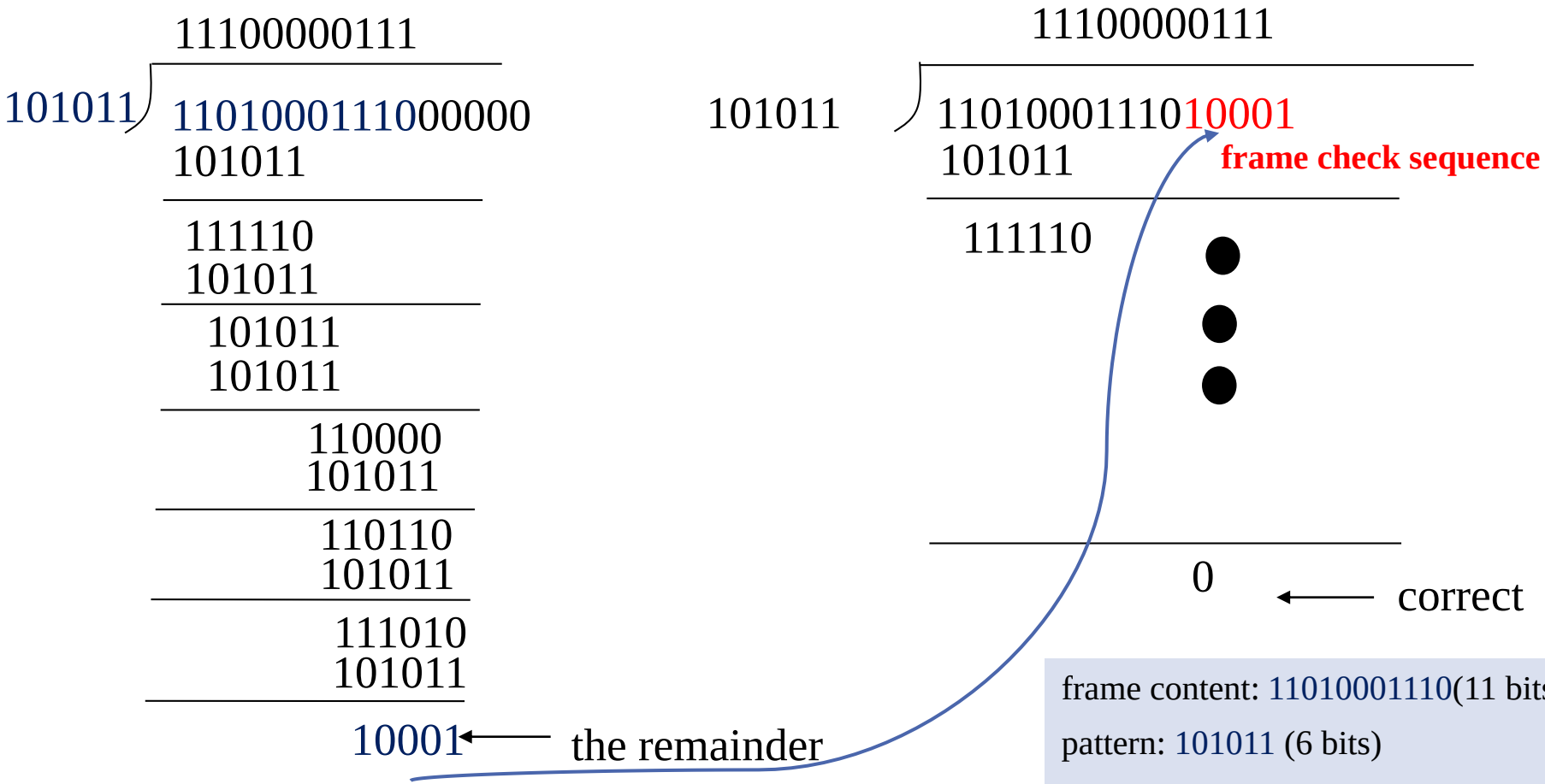
or equivalently:

if we divide $D \cdot 2^r$ by G, want remainder R to satisfy:

$$R = \text{remainder}\left[\frac{D \cdot 2^r}{G}\right]$$



Error Detection Code
Cyclic Redundancy Check



frame content: 11010001110(11 bits)

pattern: 101011 (6 bits)

frame check sequence = (5 bits)

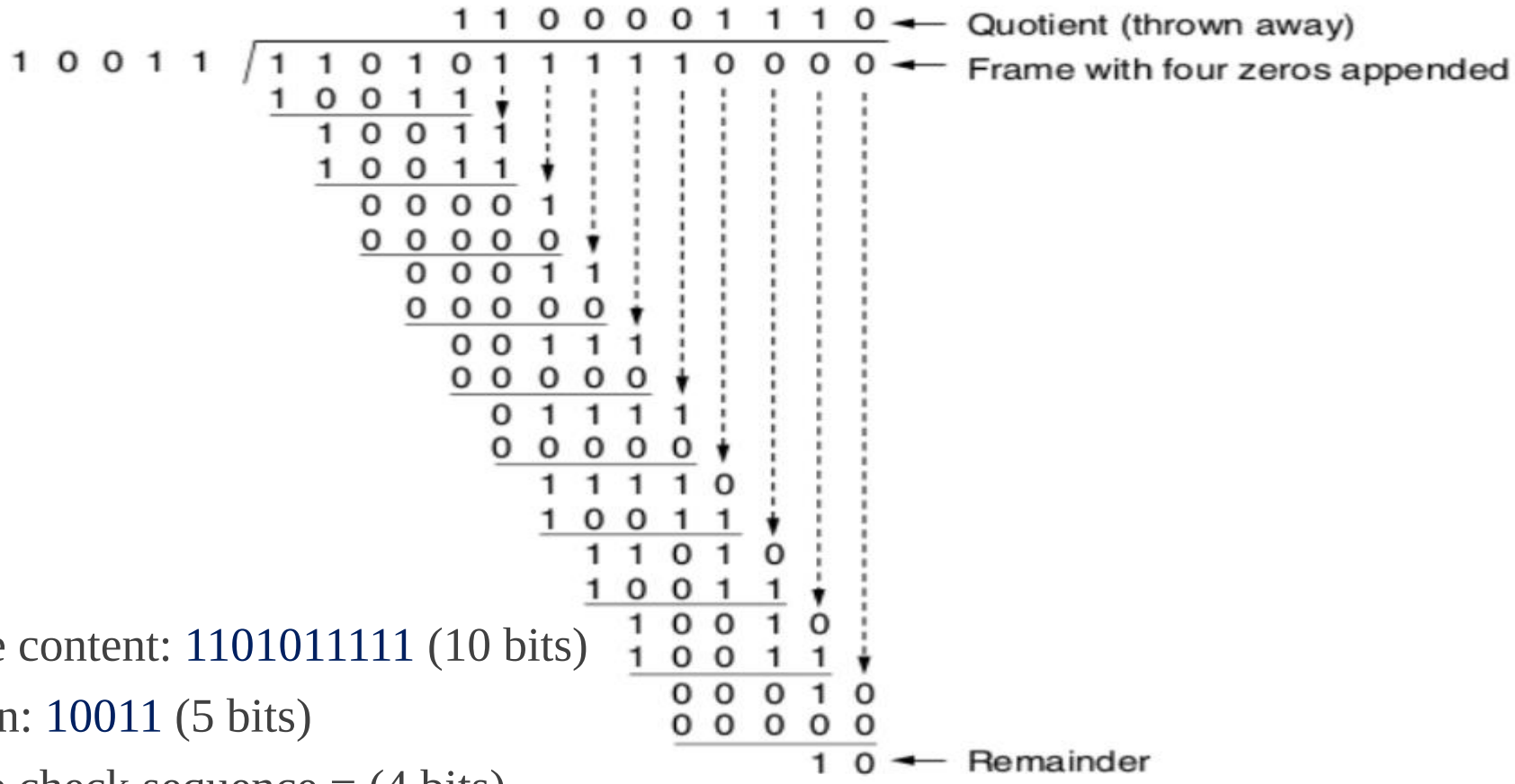
Error Detection Code
Cyclic Redundancy Check (CRC)

Frame content: 1101011111 (10 bits)

Pattern: 10011 (5 bits)

Frame check sequence = (4 bits)

Error Detection Code
Cyclic Redundancy Check (CRC)



Frame content: 1101011111 (10 bits)

Pattern: 10011 (5 bits)

Frame check sequence = (4 bits)

Error Detection Code
Cyclic Redundancy Check (CRC)

Common pattern

$$CRC - 12 = x^{12} + x^{11} + x^3 + x^2 + x + 1$$

$$CRC - 16 = x^{16} + x^{15} + x^2 + 1$$

$$CRC - 16 = x^{16} + x^{12} + x^5 + 1$$

$$CRC - 32 = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

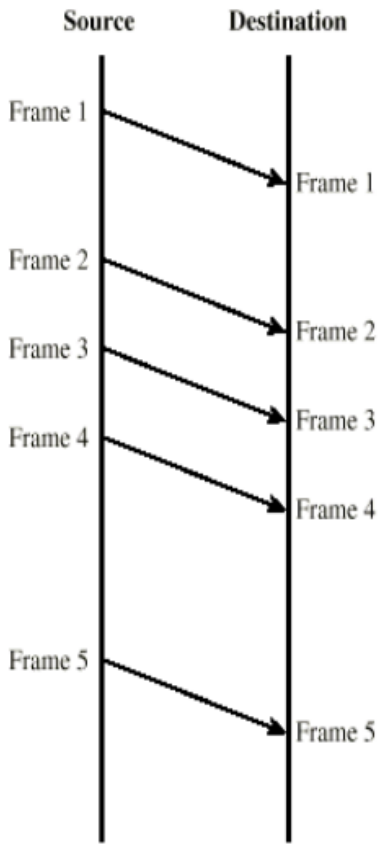
Flow control

Error Control

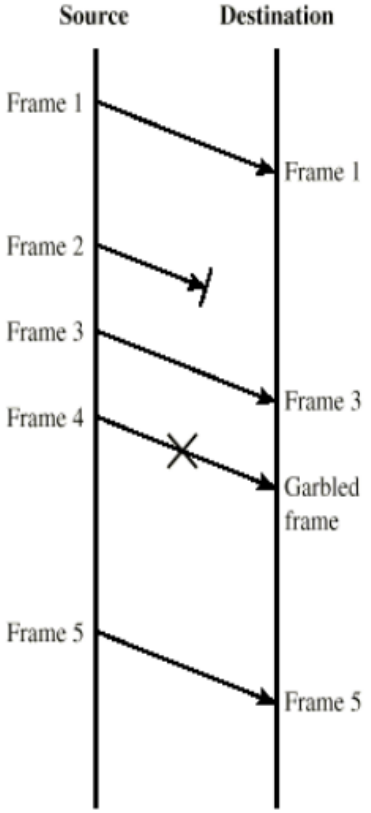
Receiver response to incoming frame

- Silently discard when the incoming frame is corrupt
- Positive acknowledgement when the incoming frame is correct
- Negative acknowledgement when the incoming frame is corrupt

Error Control



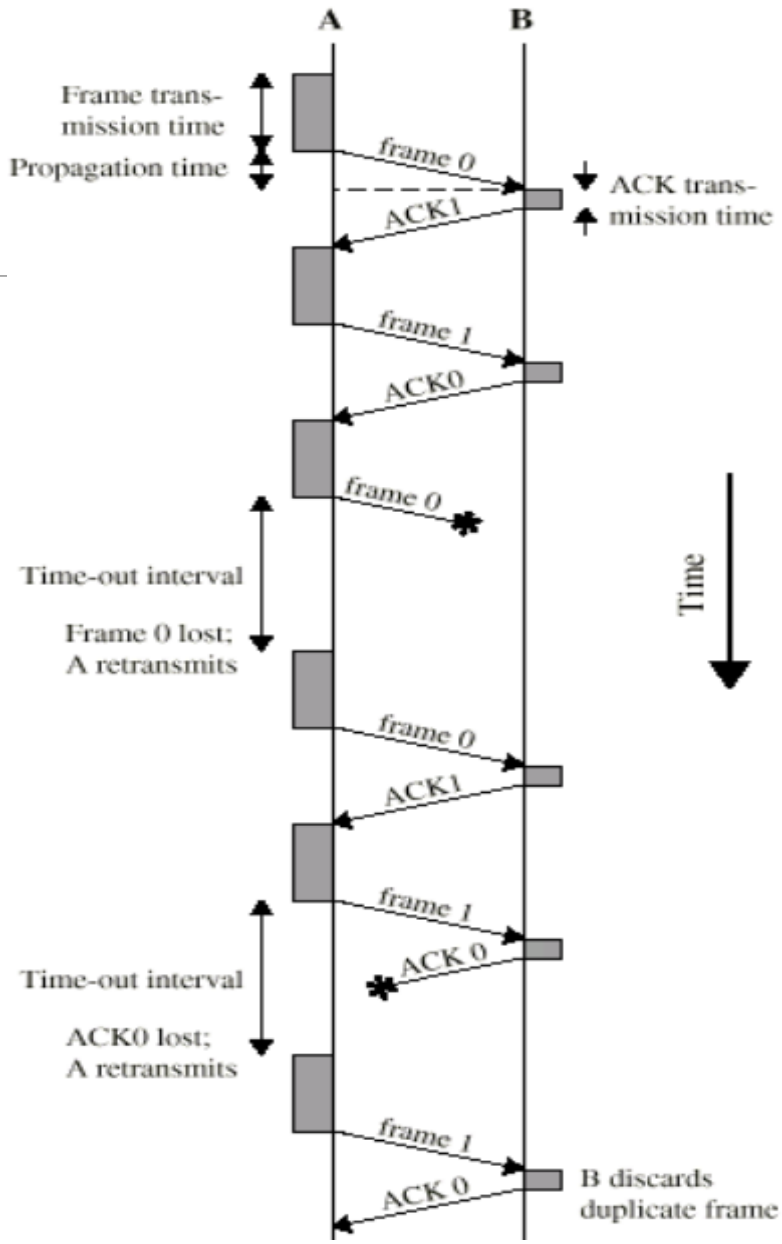
Error-free transmission



Transmission with losses and errors

Error Control

- Sender does not know if the frame was transmitted correctly
  - → ACK frame
- ACK frame could be lost
  - → Timer
- Receiver can not recognize duplicate frames
  - → Assign a sequence number to each frame



Data transmission modes

Simplex data transmission

- Unidirectional communication
- Only send data OR receive data

Half duplex data transmission

- Bidirectional communication
- Can send and receive data but one a time

Duplex data transmission

- Bidirectional communication
- Can send and receive data simultaneously
- Frame types: DATA, ACK, NACK
- Use ***piggybacking*** **technology**

Flow Control

Responsibilities of data link layer

- Flow control
- Error control

Flow control: keep fast transmitter from overwhelming slow receiver

- Rate based flow control: consists of two phases: *rate setting* by sources and network, and *rate control* by sources [https://www.nap.edu/read/5769/chapter/5]
- Feedback based flow control
  - Solutions: stop and wait and sliding window protocol

Flow Control

Stop and wait protocol

- The sender sends *one* frame and waits for acknowledgement from the receiver
- Once the receiver receives the frame, it sends an acknowledgment frame back to the sender.
- On receiving the acknowledgment frame, the sender understands that the receiver is ready to accept the next frame. So it sender the next frame in queue.

Go-back-N protocol

- The sender sends *N* frames at a time and waits for acknowledgement from the receiver
- The receiver discards all frames after the error frame. The sender retransmits all frames started from the error frame.
- Wastes lot of bandwidth if error rate high

Selective repeat protocol

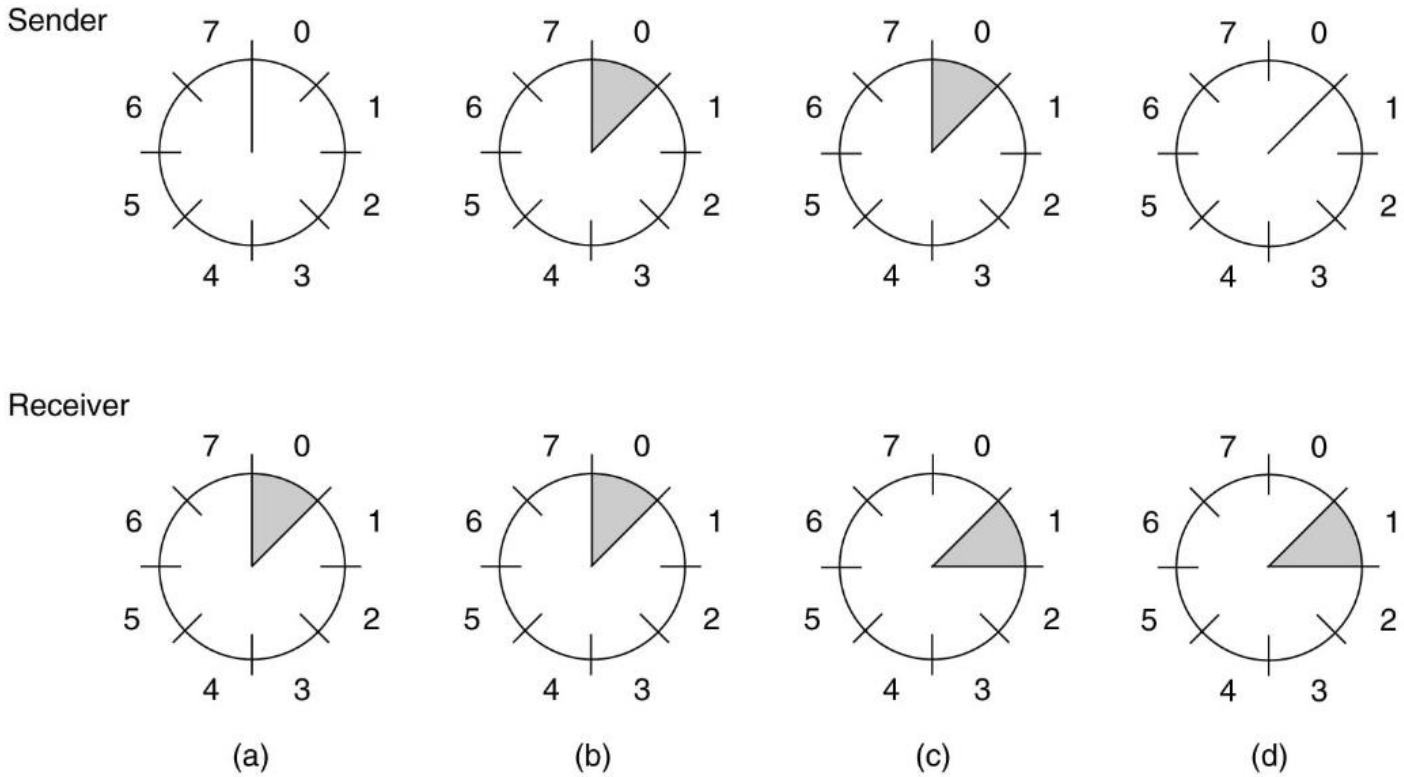
- The sender sends *N* frames at a time and waits for acknowledgement from the receiver
- The receiver only discards error frame. The sender only retransmits the error frame.

Flow Control
Sliding window protocol

- Sender and receiver maintain *buffer space*
- Frames have sequence number from 0 to maximum 2^n-1 , where *n* is bit field
- At any moment, the sender maintains a list of sequence numbers it is permitted to send - these fall within the ***sending window***. These are frames sent-but-no-ACK and frames not-yet-sent.
  When new packet from Network layer comes in to send, it is given highest number, and upper edge of window advanced by 1.
  When ACK comes in, lower edge of window advanced by 1.
- Receiver has ***receiving window*** - the frames it is permitted to accept.

https://computing.dcu.ie/~humphrys/Notes/Networks/data.sliding.html

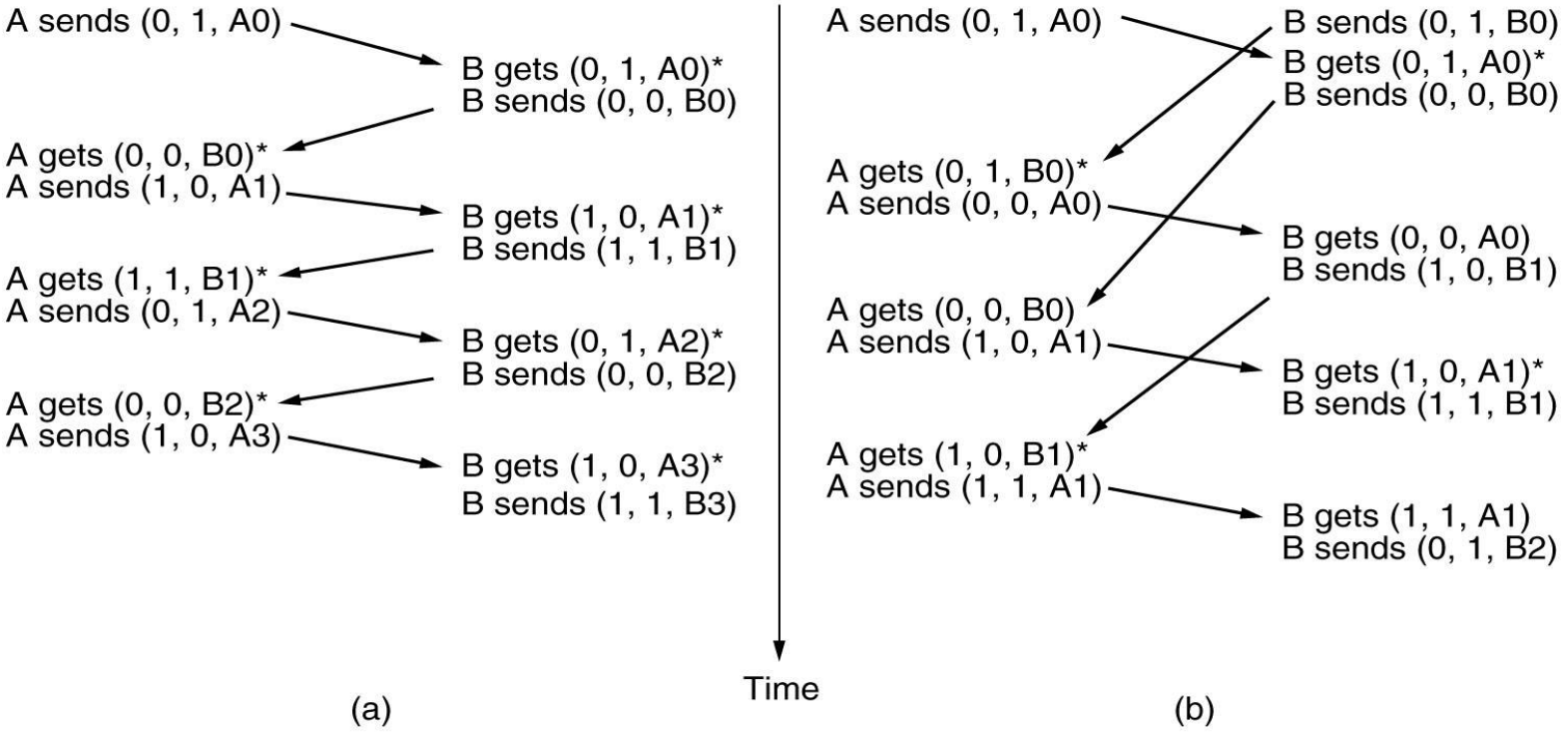
Flow Control
Sliding window protocol



https://computing.dcu.ie/~humphrys/Notes/Networks/data.sliding.html

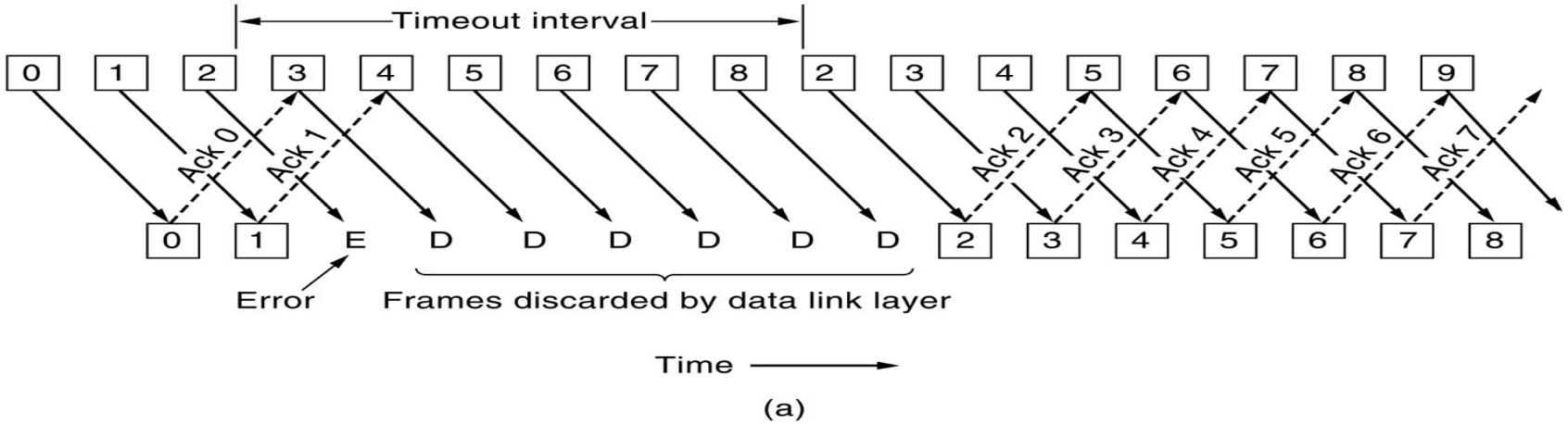
Flow Control
Sliding window protocol

Window size 1. Stop-and-wait. Must get ACK before can send next frame.

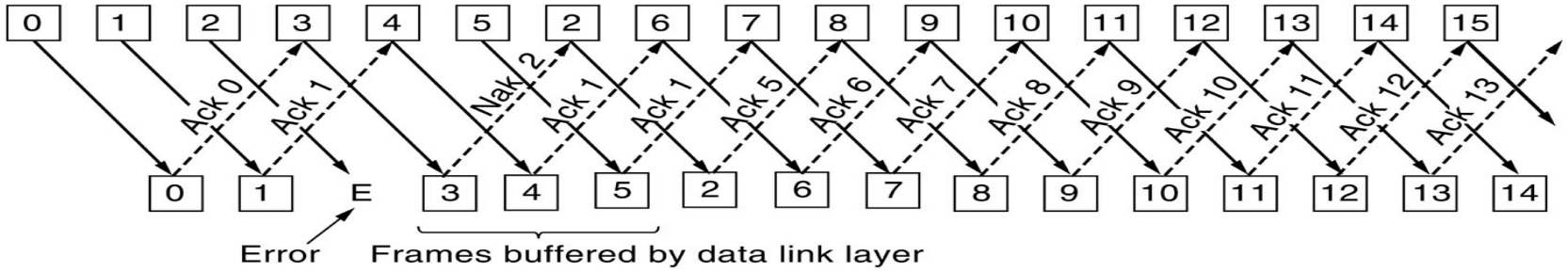


https://computing.dcu.ie/~humphrys/Notes/Networks/data.sliding.html

Flow Control
Sliding window protocol



(a)

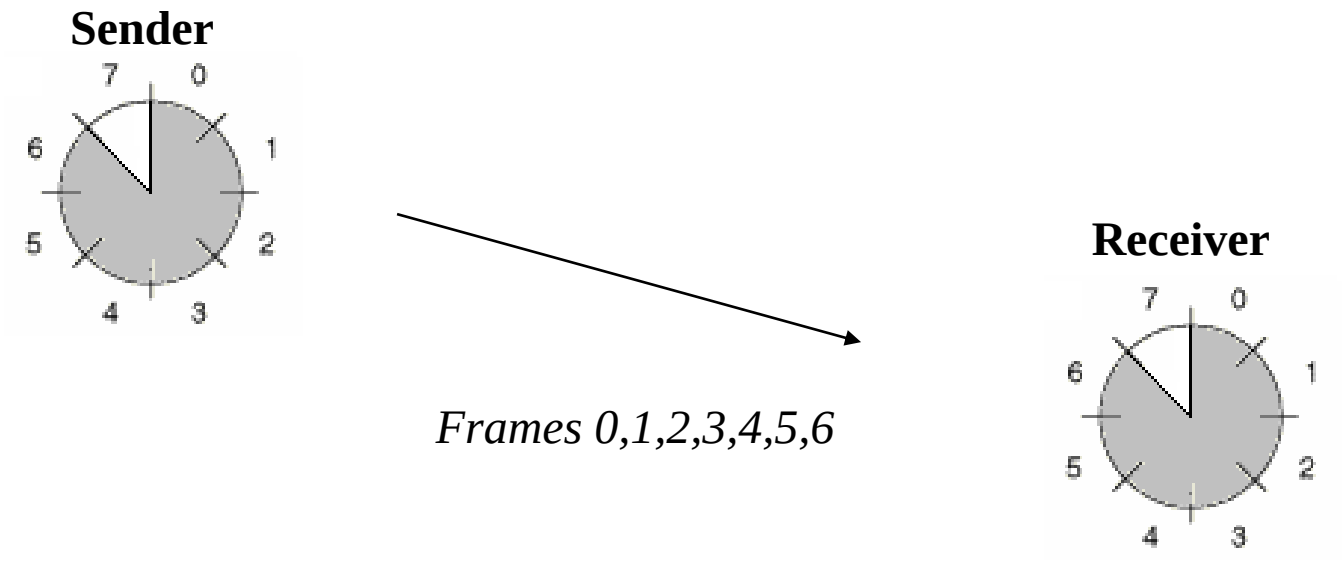


(b)

https://computing.dcu.ie/~humphrys/Notes/Networks/data.sliding.html

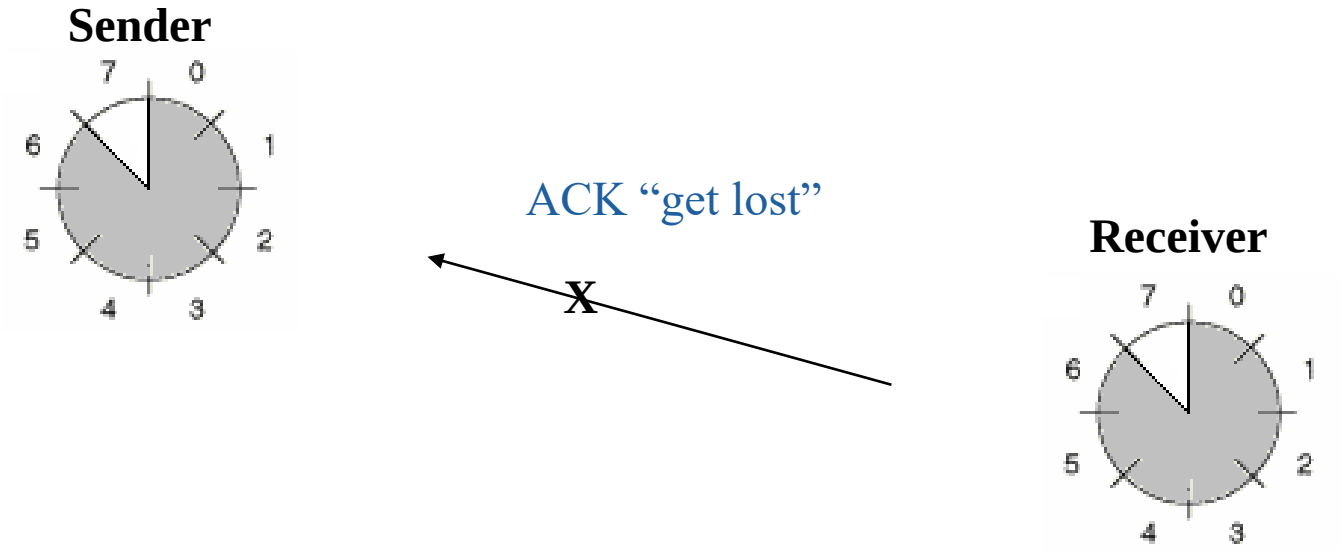
Flow Control
Sliding window protocol

- Size of the sliding window
  - Frames have sequence number from 0 to maximum 2^n-1 , where *n* is bit field
  - If $n = 3$



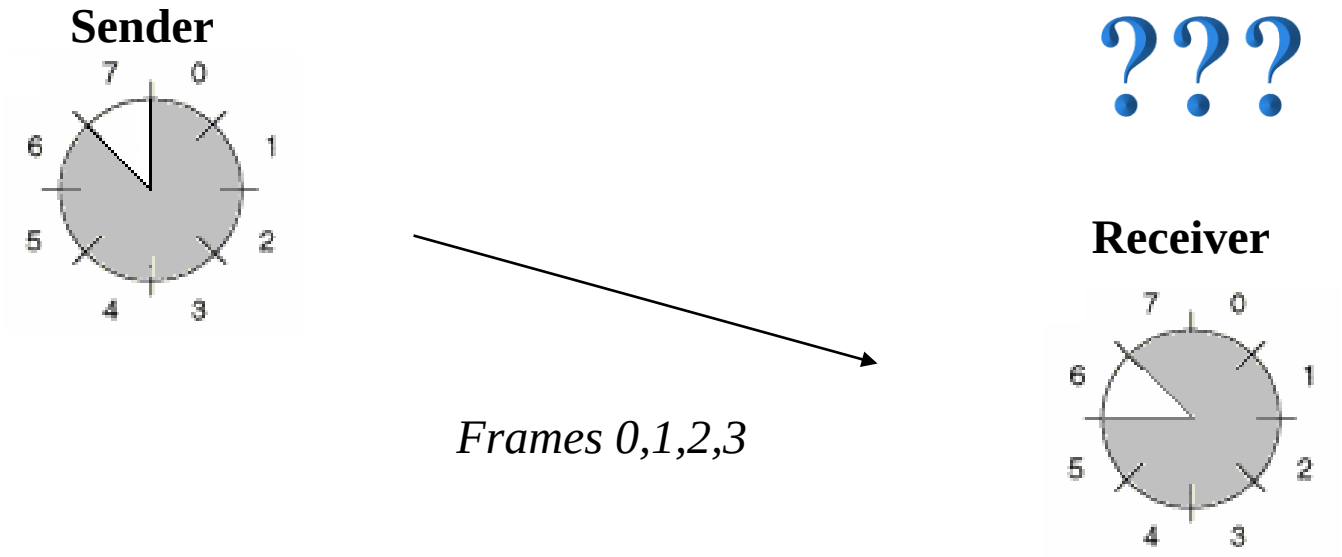
Flow Control
Sliding window protocol

- Size of the sliding window
  - Frames have sequence number from 0 to maximum 2^n-1 , where *n* is bit field
  - If $n = 3$



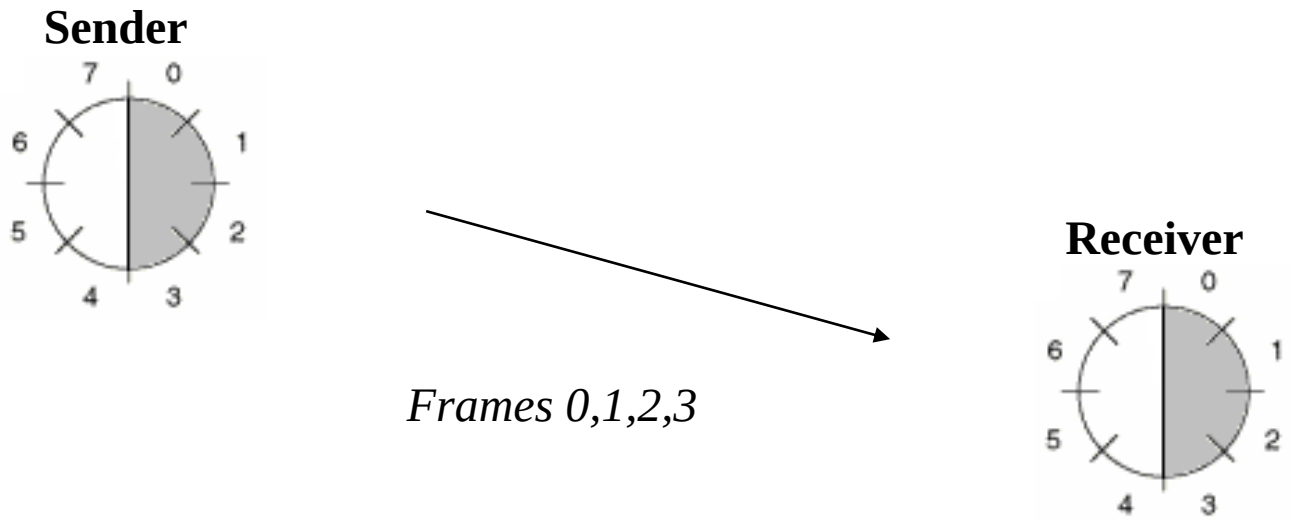
Flow Control
Sliding window protocol

- Size of the sliding window
  - Frames have sequence number from 0 to maximum 2^n-1 , where *n* is bit field
  - If $n = 3$



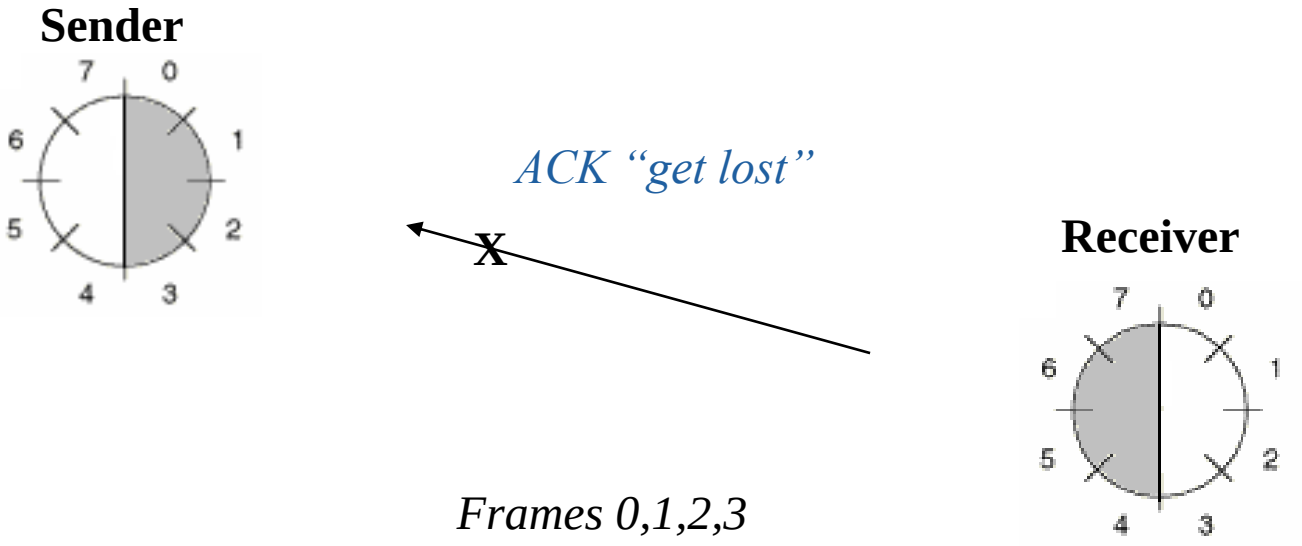
Flow Control
Sliding window protocol

- Size of the sliding window
  - Frames have sequence number from 0 to $(2^n)/2$
  - If $n = 3$



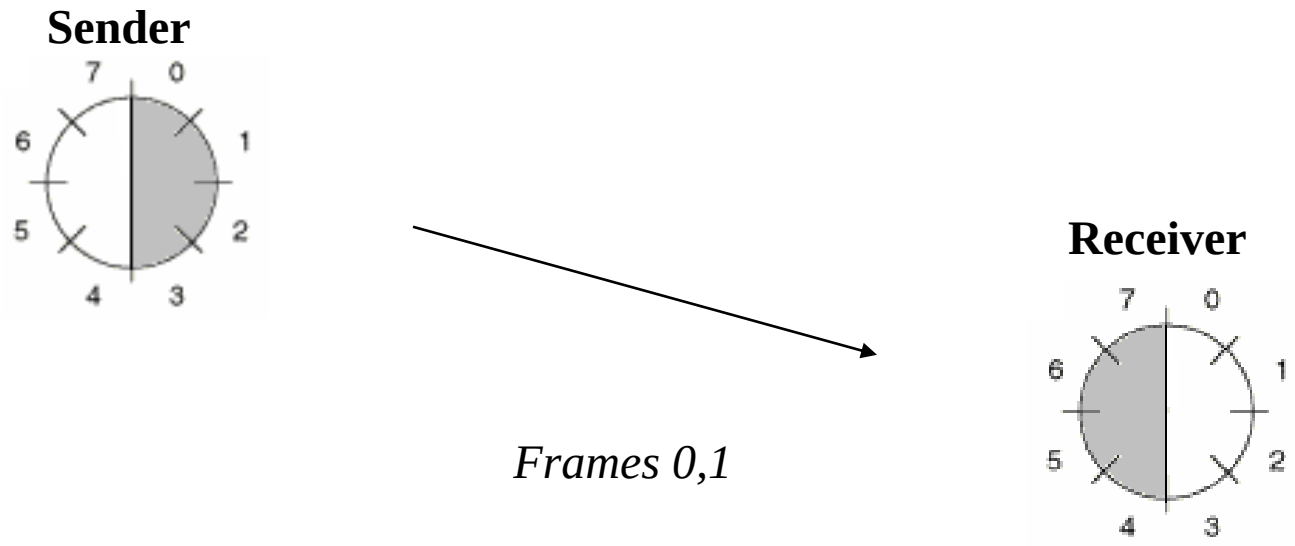
Flow Control
Sliding window protocol

- Size of the sliding window
  - Frames have sequence number from 0 to $(2^n)/2$
  - If $n = 3$

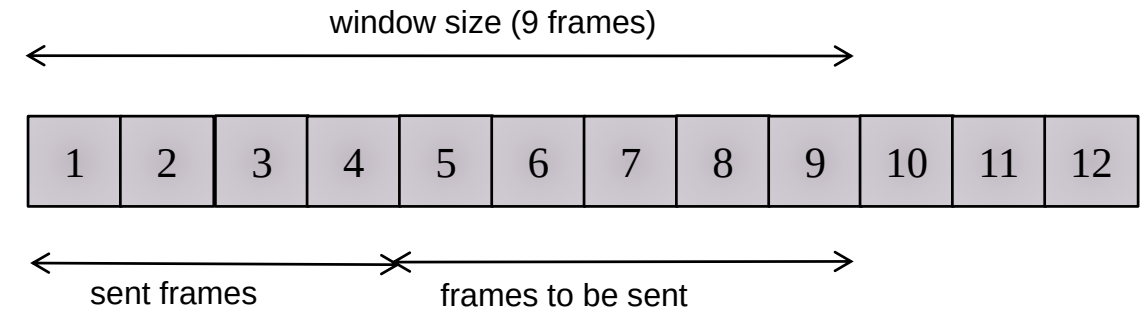


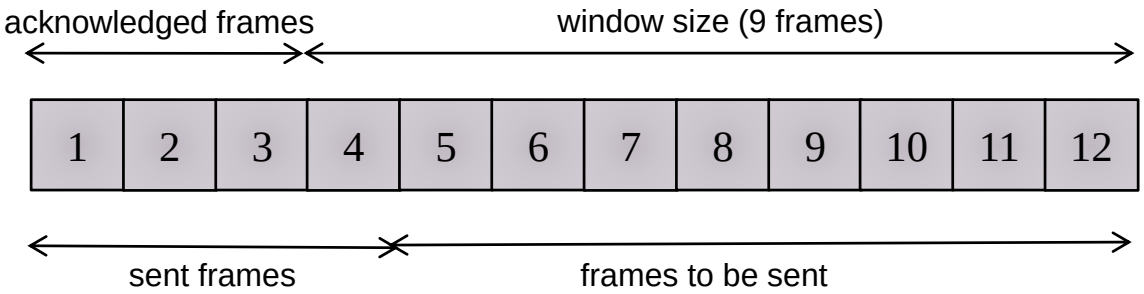
Flow Control
Sliding window protocol

- Size of the sliding window
  - Frames have sequence number from 0 to $(2^n)/2$
  - If $n = 3$



Sliding Window over Transmitted Frames





HDLC Protocol
(High Level Data Link Control)

Slides Are Adapted From

- Ajit Pal, CSE IIT, Kharagpur Https://Nptel.Ac.In/Course.Html
- Https://Www.Tutorialspoint.Com/High-level-data-link-control-hdlc

HDLC stations

Primary station

- Controls operation of link
- Issues commands (frames)
- Maintains separate logical link to each secondary station

Secondary station

- Under control of primary station
- Issues responses (frames)

Combined station

- May issue commands and responses
- Combines the features of primary and secondary stations

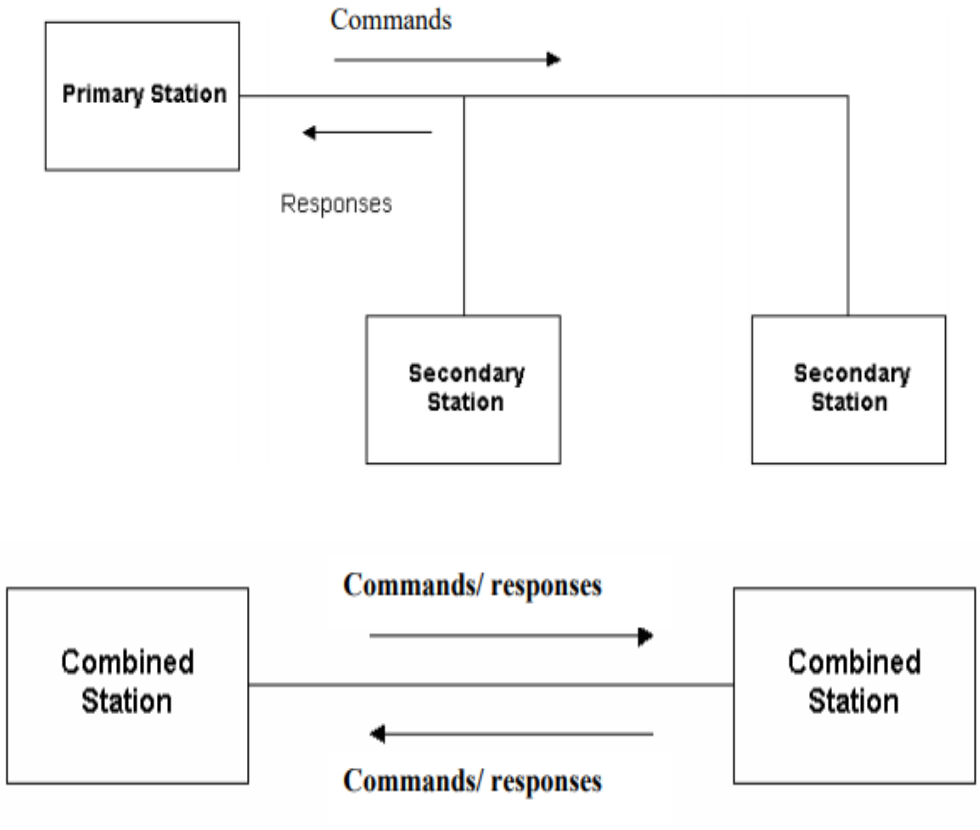
HDLC link configurations

Unbalanced

- One primary and one or more secondary stations
- Supports full duplex and half duplex

Balanced

- Two combined stations
- Supports full duplex and half duplex



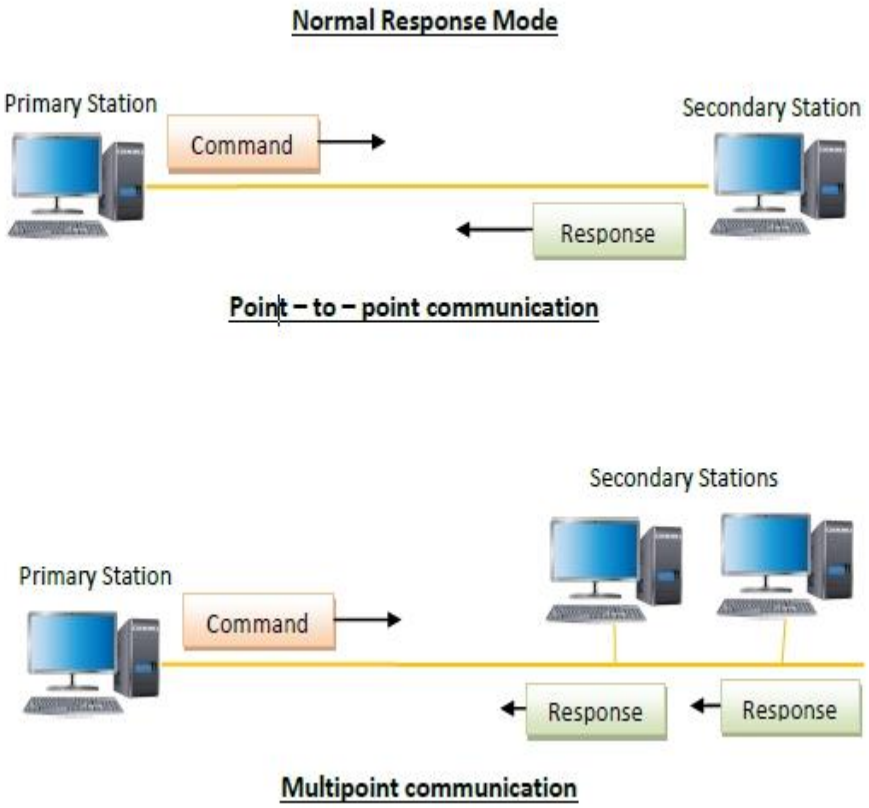
HDLC Transfer Modes

- Normal Response Mode (NRM)
- Asynchronous Balanced Mode (ABM)
- Asynchronous Response Mode (ARM)

HDLC Transfer Modes-NRM

Normal Response Mode (NRM)

- Unbalanced configuration
- Primary can only initiate transmission
- Secondary may only transmit data in response to command (poll) from primary
- Used on multi-drop lines
- Host computer as primary
- Terminals as secondary

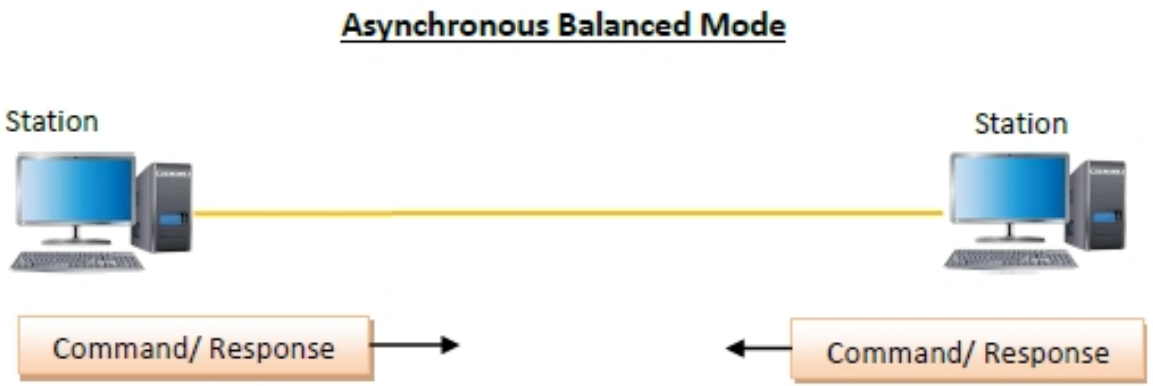


https://www.tutorialspoint.com/high-level-data-link-control-hdlc

HDLC Transfer Modes - ABM

Asynchronous Balanced Mode (ABM)

- Balanced configuration
- Either station may initiate transmission without receiving permission
- Most widely used
- No polling overhead



https://www.tutorialspoint.com/high-level-data-link-control-hdlc

HDLC Transfer Modes - ARM

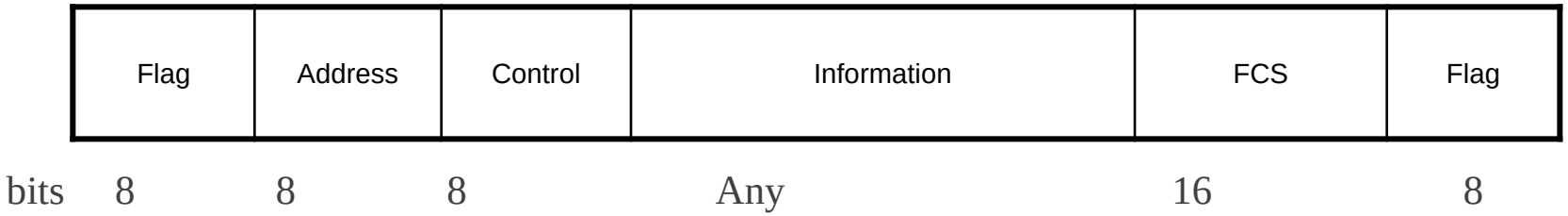
Asynchronous Response Mode (ARM)

- Unbalanced configuration
- Secondary may initiate transmission without permission form primary
- Primary is responsible for connect, disconnect, error recovery, and initialization
- rarely used

HDLC frame structure

A synchronous, reliable, full-duplex data delivery protocol

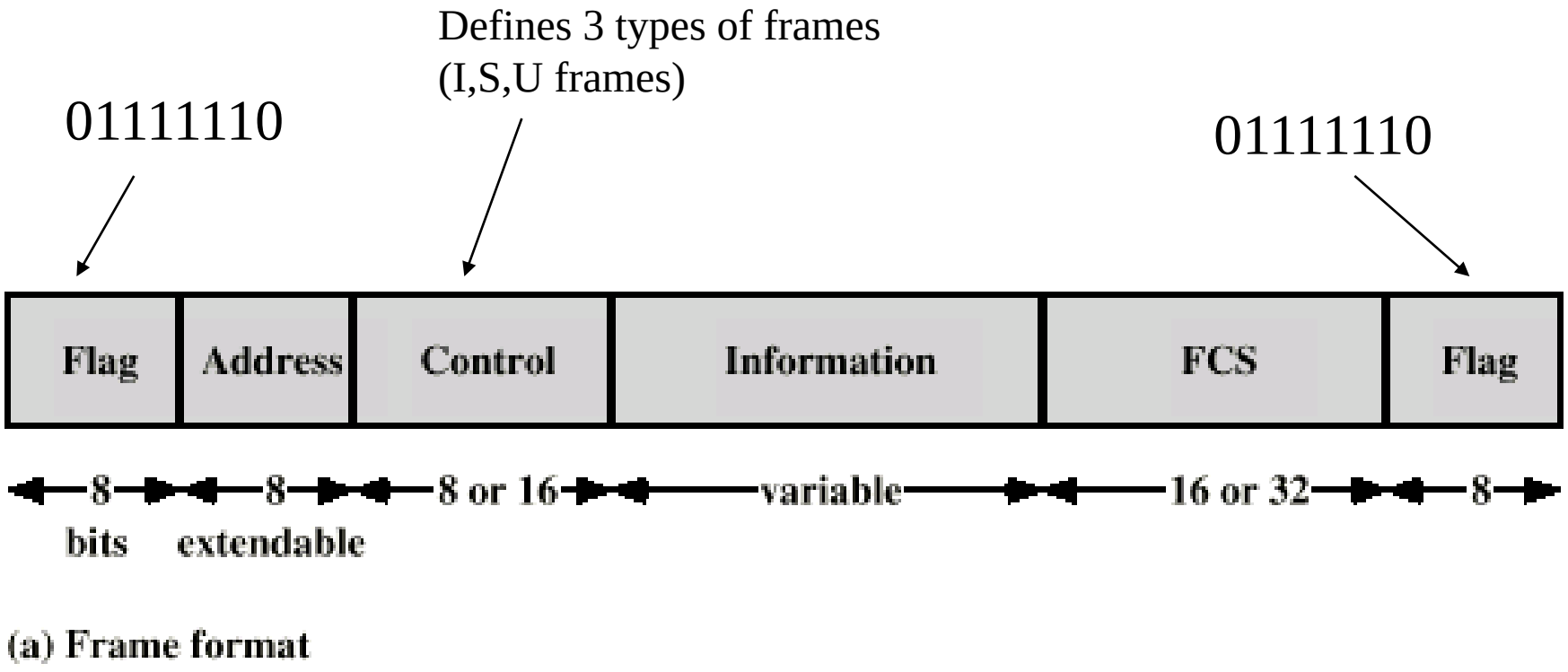
Bit-oriented frame format



Types of frames: information, supervisory, unnumbered

Used on peer-to-peer WAN link

HDLC frame structure



(a) Frame format

HDLC frame structure

Address Field

- Identifies secondary station that sent or will receive frame
- Usually 8 bits long
- All ones (11111111) is broadcast

(b) Extended Address Field

HDLC frame structure

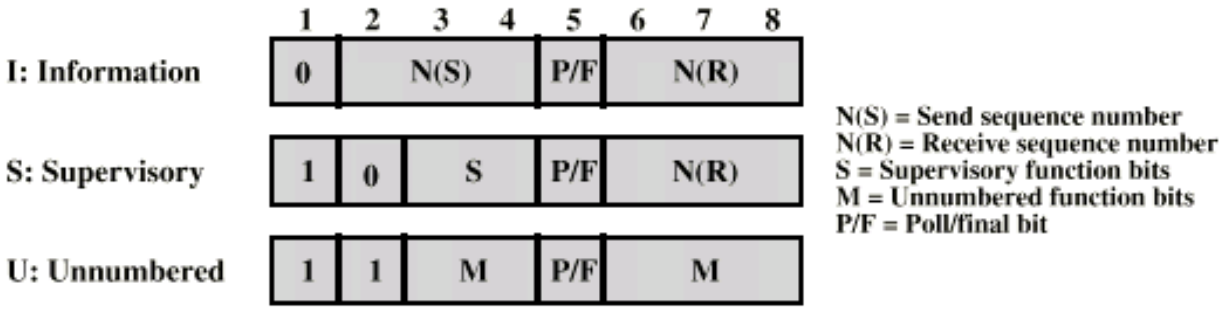
Control Field: different for different frame type

- **I-frame** (information frame)
  - data to be transmitted to user (next layer up)
  - Flow and error control piggybacked on information frames
- **S-frame** (Supervisory frame): used for flow and error control
- **U-frame** (Unnumbered frame): supplementary link control

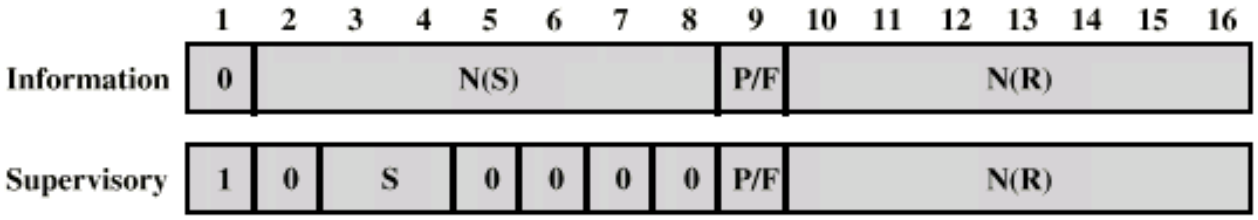
First one or two bits of control filed identify frame type

HDLC frame structure

Control Field:



(c) 8-bit control field format



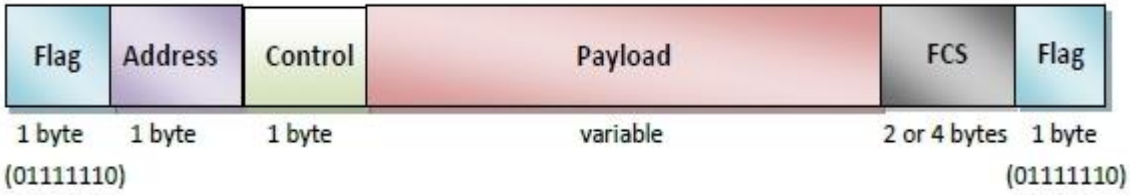
(d) 16-bit control field format

HDLC frame structure

Poll/Final Bit: use depends on context

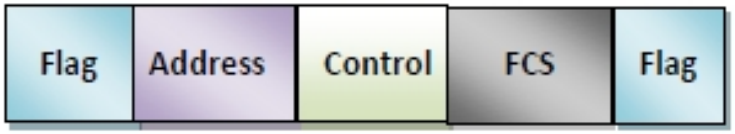
- **Command frame**
  - P bit : used for poll from primary
  - 1 to solicit (poll) response from peer
- **Response frame**
  - F bit : used for response from secondary
  - 1 indicates response to soliciting command

HDLC Frame



I – Frame

S – Frame

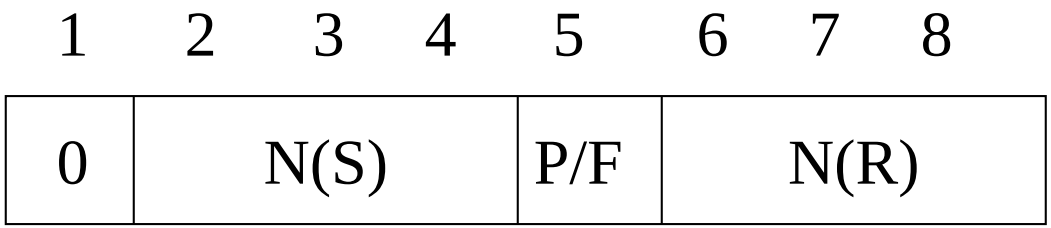


U – Frame

https://www.tutorialspoint.com/high-level-data-link-control-hdlc

HDLC frame structure

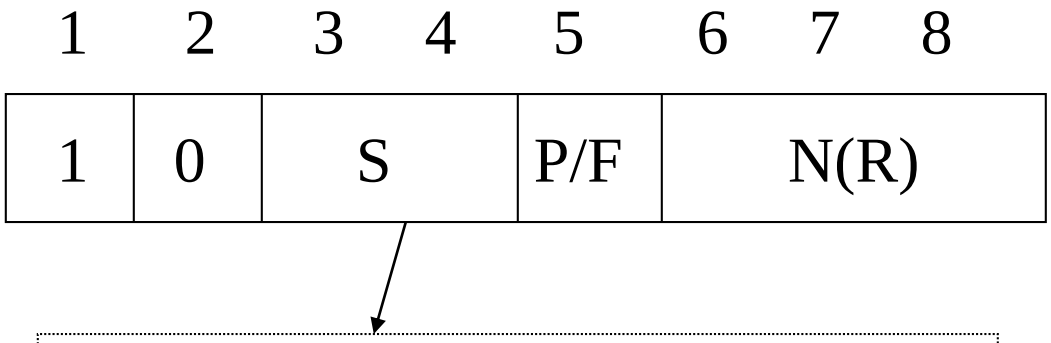
I-frame: contains the sequence number of transmitted frames and a piggybacked ACK



- I,0,0
- I,1,0
- I,2,0,P

HDLC frame structure

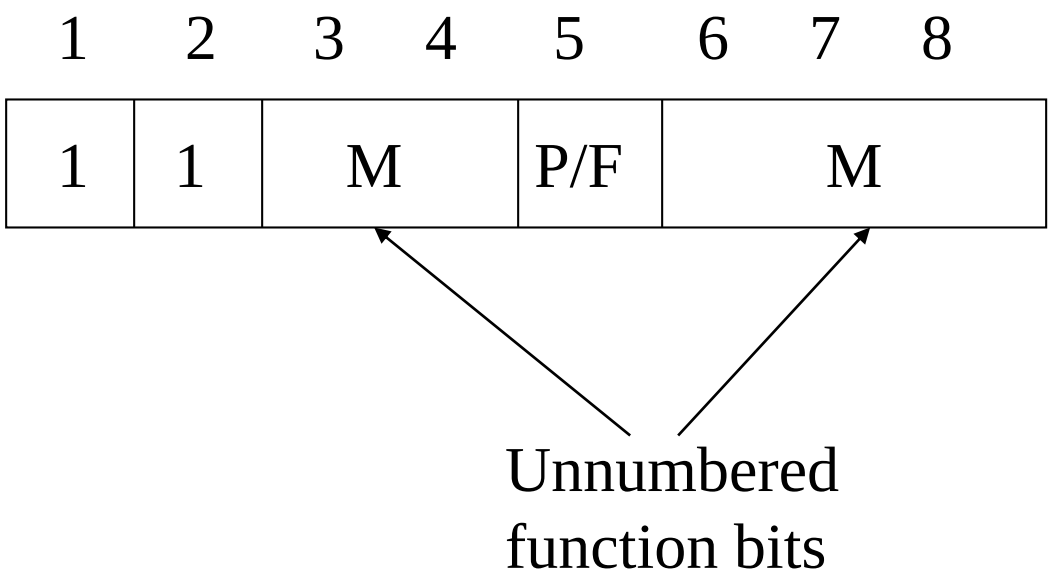
S-frame: Used for flow and error control



- RR --- receive ready
- RNR --- receive not ready
- REJ --- reject on frame N(R)
- SREJ --- selective reject on N(R)

HDLC frame structure

U-frame: Mode setting, recovery, connect/disconnect



HDLC frame structure

Unnumberred frames:

- Set normal response mode (SNRM)
- Set asynchronous response mode (SARM)
- Set asynchronous balanced mode (SABM)
- Disconnect (DISC)
- Unnumberred acknowledgement (UA)
- Disconnect mode (DM)
- Request disconnect (RD)
- Unnumberred poll (UP)
- Reset (RSET)
- Exchange identification (XID)
- Test (TEST)
- Frame reject (FRMR)

HDLC frame structure

Frame Check Sequence Field:

- FCS
- Error detection
- 16 bit CRC
- Optional 32 bit CRC

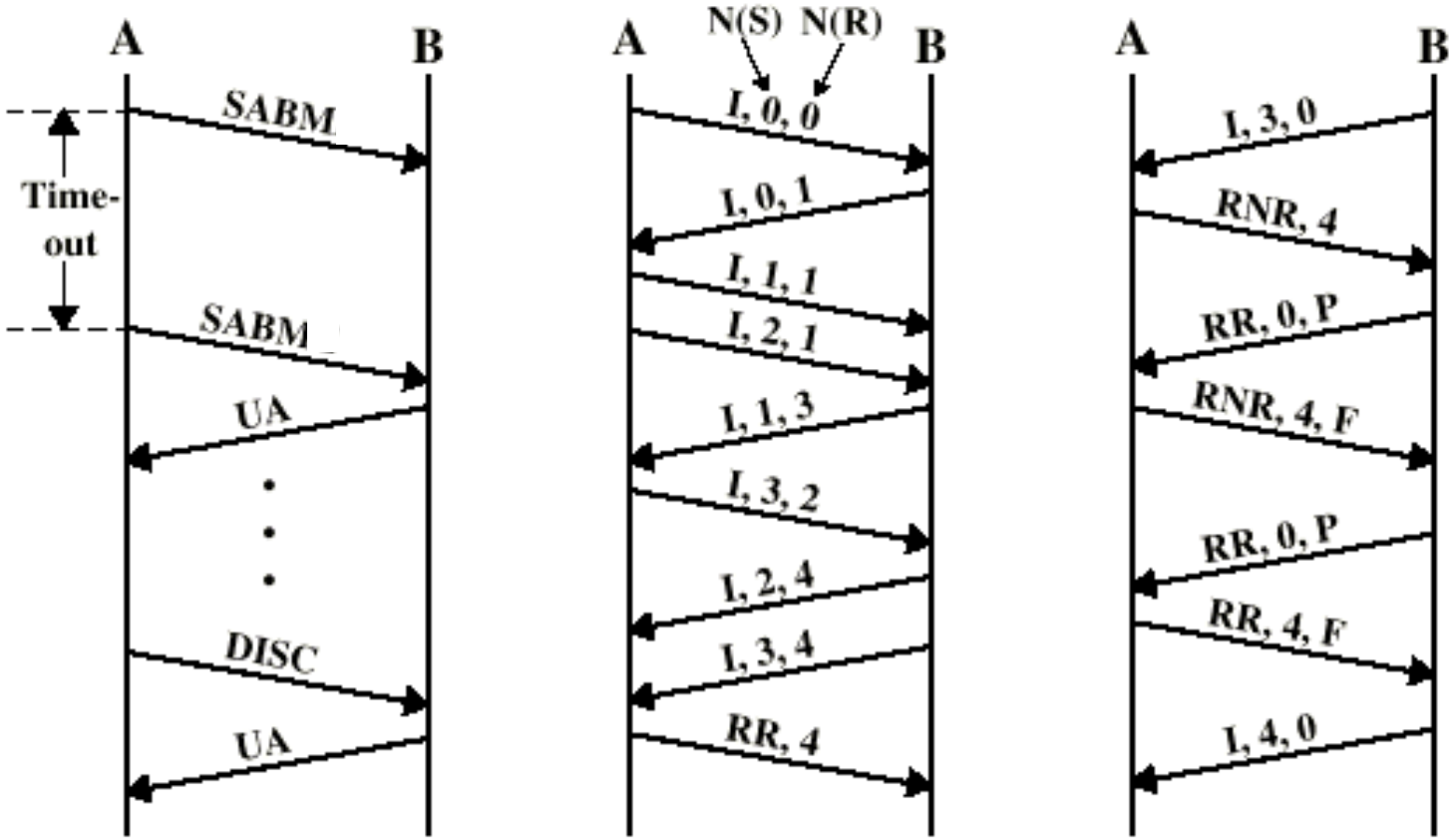
HDLC Operation

Exchange of information, supervisory and unnumbered frames

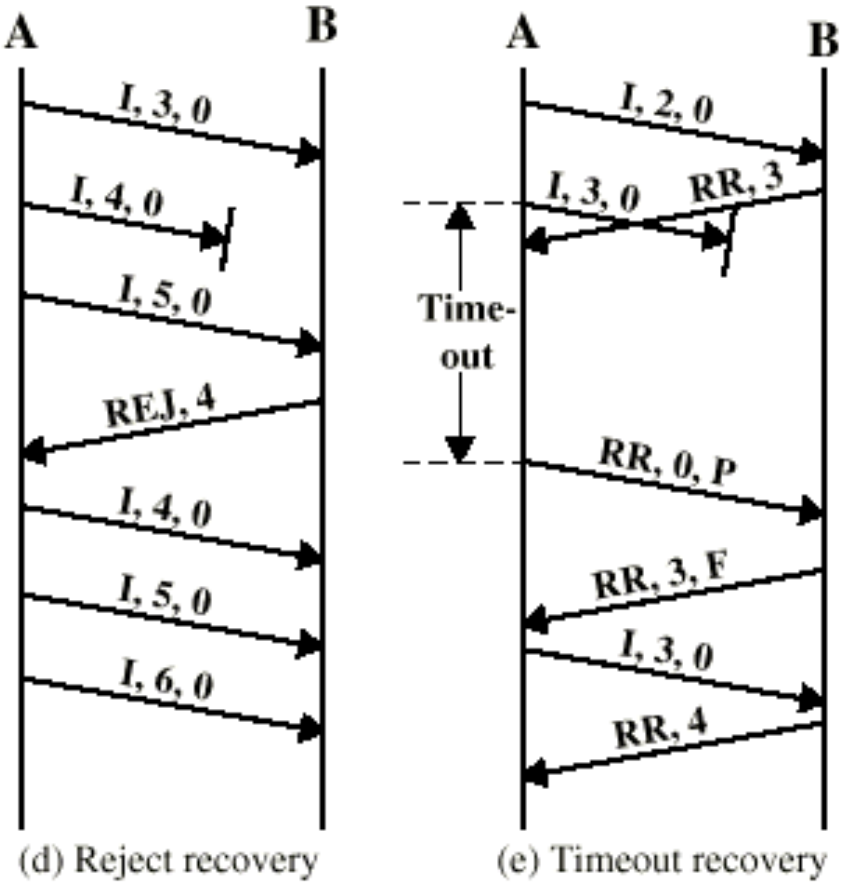
Three phases

- Initialization
- Data transfer
- Disconnect

HDLC


A
B
SABM
Time-out
SABM
UA
DISC
UA
(a) Link setup and disconnect
N(S) N(R)
I, 0, 0
I, 0, 1
I, 1, 1
I, 2, 1
I, 1, 3
I, 3, 2
I, 2, 4
I, 3, 4
RR, 4
(b) Two-way data exchange
I, 3, 0
RNR, 4
RR, 0, P
RNR, 4, F
RR, 0, P
RR, 4, F
I, 4, 0
(c) Busy condition

HDLC



(d) Reject recovery (e) Timeout recovery


THANK YOU