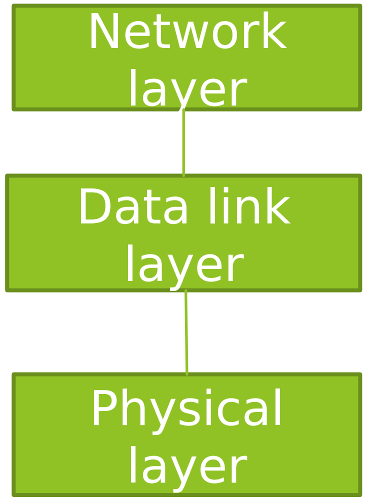
Unit II
Data Link Layer

Introduction

-A reliable & effective communication between two machines can be achieved via Data link layer

-The layer basically deals with frame formation, flow control, error control, addressing & link management.



Network layer

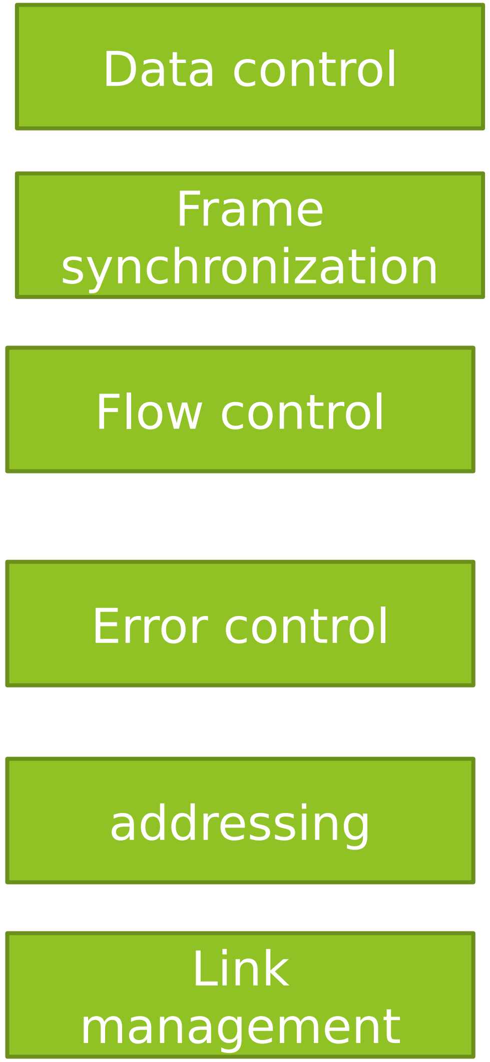
Data link layer

Physical layer

DLL provide services to the network layer

DLL takes services from the physical layer

Data link layer design issue



Data control

Frame synchronization

Flow control

Error control

addressing

Link management

1)Frame synchronization

The source machine send data in the form of frame to the Destination machine.

2)Flow control

The source machine cant send frames at a rate faster than The capacity of destination machine

3)Error Control

The error included during transmission from source to destination .must be detected & corrected at the destination machine

4)Addressing:

When multiple machines are connected together identity of individual machines must be specified with data frame .this is known as addressing

5)Control and data on same link:

Data & control information is combined in a frame & transmitted from source to destination.

6)Link management:

Communication link between source & destination initiated, maintained,& terminated.

- **1)Unacknowledge connectionless service:**
- It is a connectionless service.
- Connection can not established between sender & receiver.& can not release after receiving a frame
- Frame is transfer from sender to receiver.
- Destination machine does not give any acknowledgement after receiving frames.

Services provided to network layer

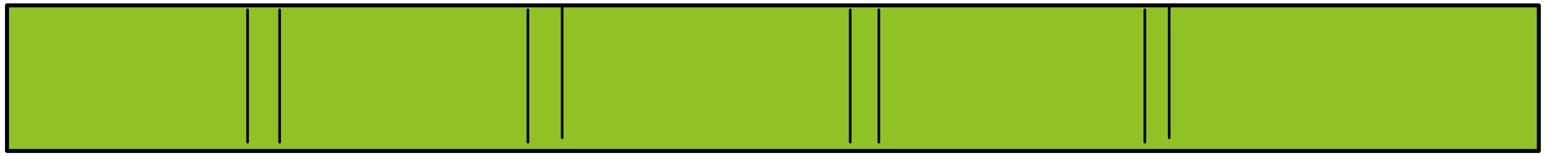
- 1)Unacknowledge connectionless service
- 2) Acknowledge connectionless service
- 3) Acknowledge connection oriented service

- **2) Acknowledge connectionless service**
- It is a connectionless service.
- Connection can not established between sender & receiver.& can not release after receiving a frame
- Frame is transfer from sender to receiver.
- Destination machine give acknowledgement after receiving frames.
- If frame is not received within specified time the sender can retransmit it.

- **3) Acknowledge connection oriented service**
- It is a connection oriented service.
- Connection can established in between sender & receiver. before transferring a data.
- All the frames are received in the manner as they are transferred .
- Connection is released after receiving a data.

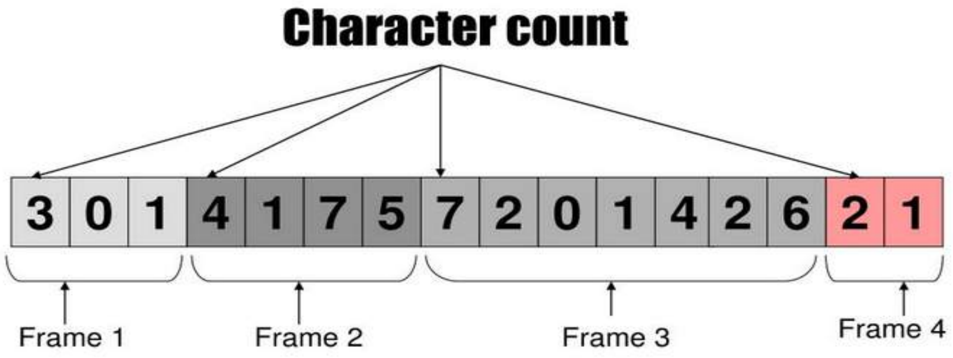
Framing

- Breaking the bit stream into frames is called as framing.
- We have just insert the time gap between two frames.



Framing Methods

- **Character count method**
- In this header is used to specify the number of character in the frame.
- This help the receiver to know the number of character present in the frame.

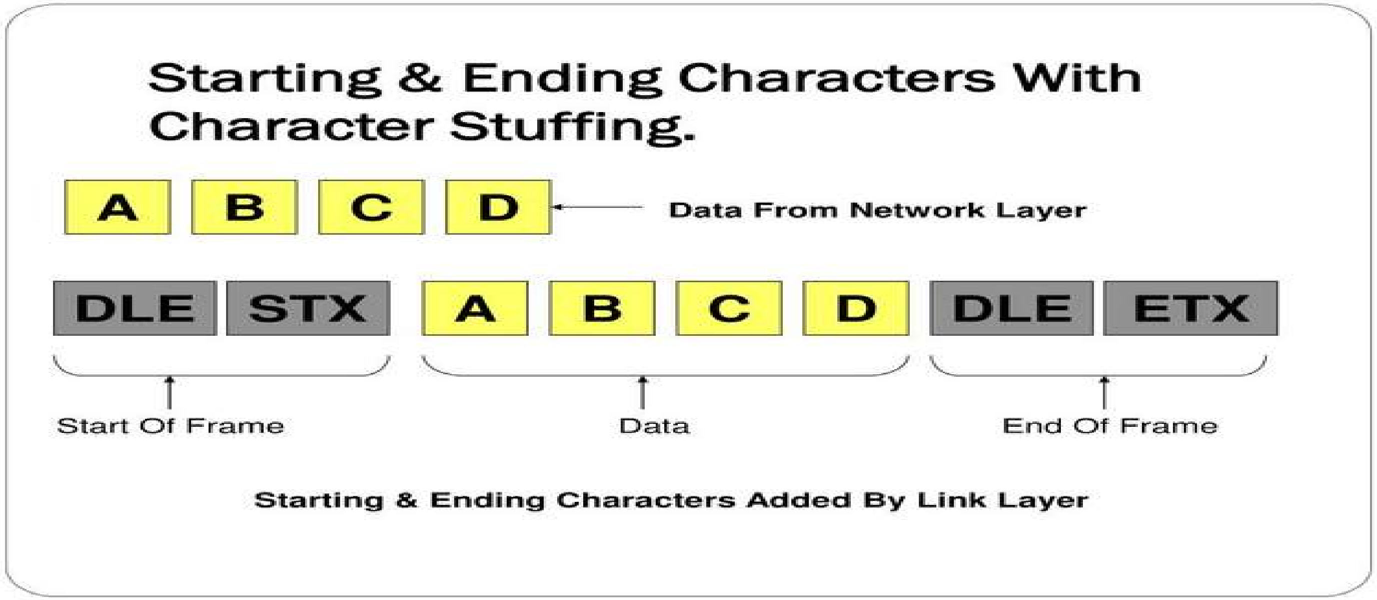


Character Count Method

- Disadvantage:
- Error can change the character count itself
- If wrong character count can received by receiver then reciver get out of synchronization.
- Will not be able to understand the starting of next frame.

2)Starting and ending character with character stuffing

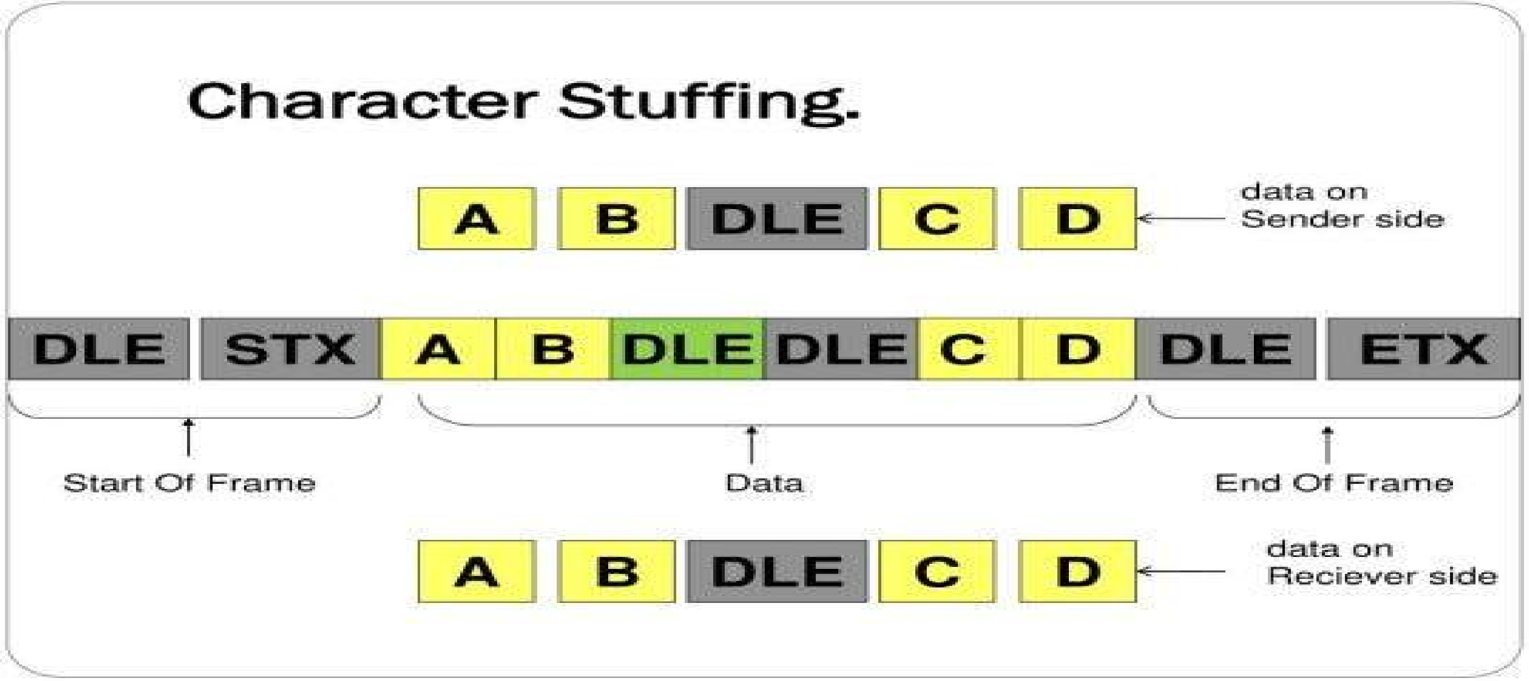
- The problem of character count solve here we use starting character at starting of frame & ending character at the end of frame.
- Here at starting of frame we put STX,DLE
- Here at ending of frame we put DLE,ETX
- If receiver has lost synchronization then receiver just search STX,DLE& ETX.



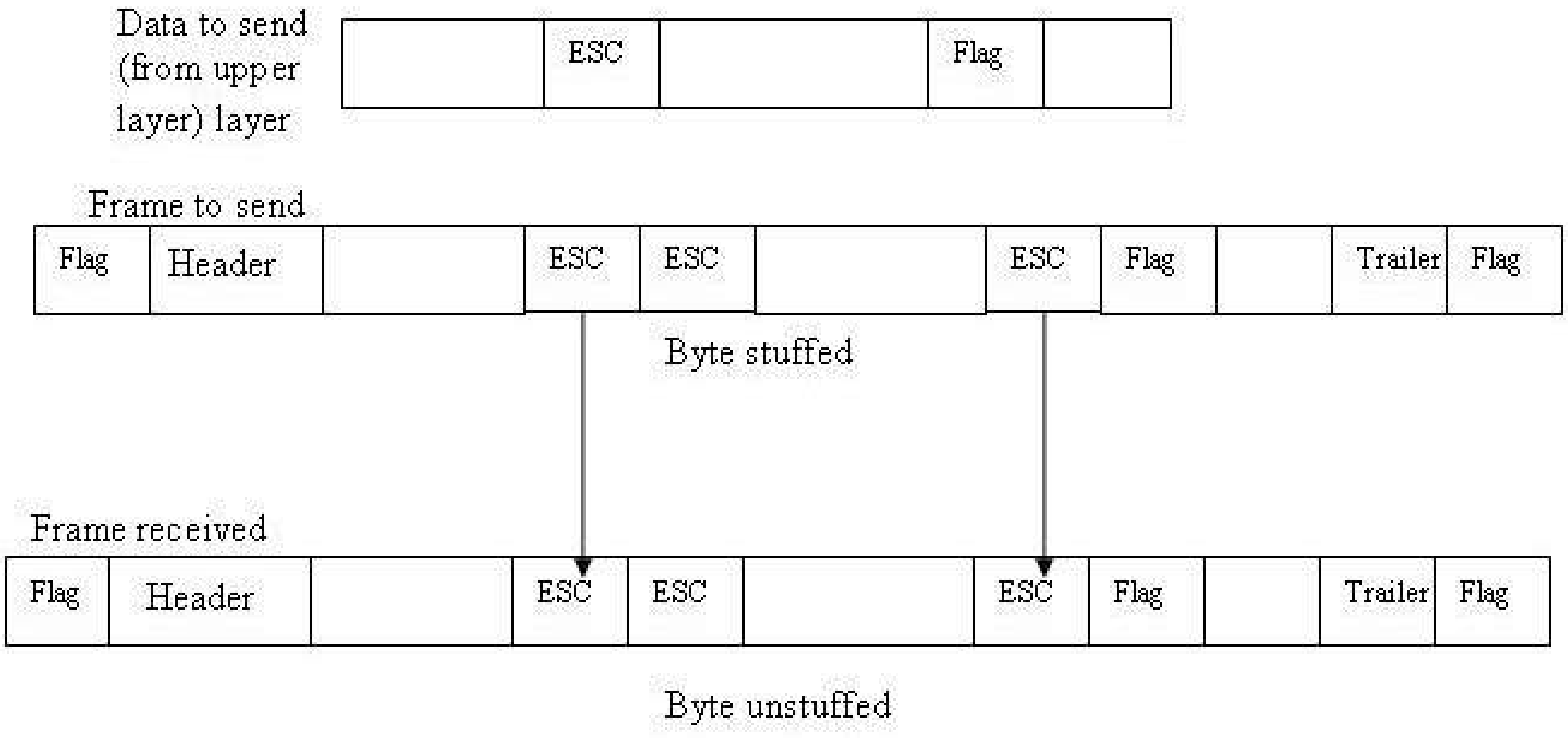
Starting & Ending Characters With Character Stuffing.

3)Character stuffing

- The problem with DLE STX & DLE ETX is that some time it is misinterpreted by receiver as a start or end of frame
- So we used technique called as character stuffing
- Data link layer insert a DLE character with data being transmitted
- & receiver side remove the DLE character.& then send to network layer



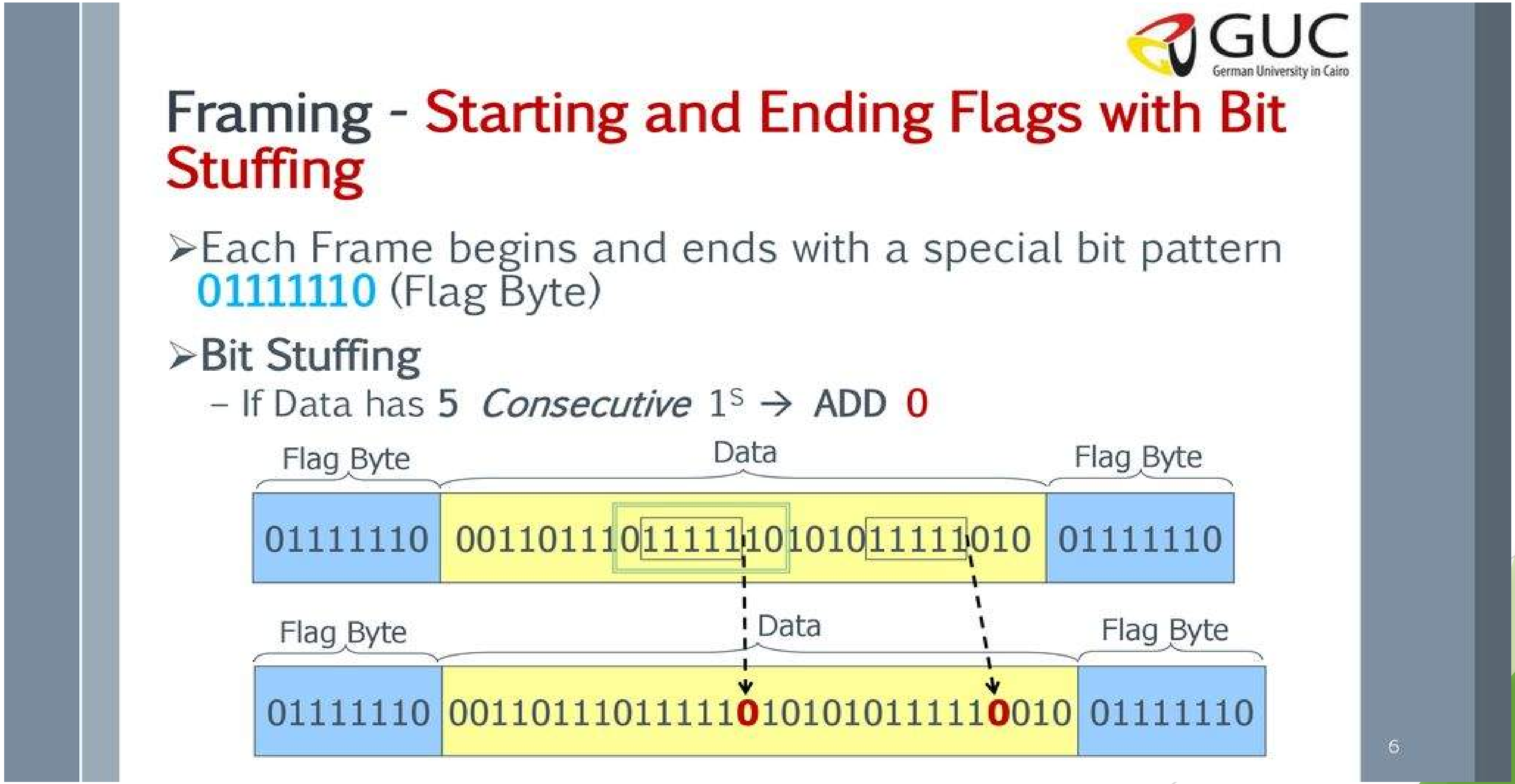
4)Byte stuffing



- In byte stuffing **special byte is added to the data section of the frame.**
- The data section is stuffed with extra byte .**this byte is called as ESC character.**
- At the **receiver side ESC byte are removed from the data** .
- Byte stuffing with escape character will allow the presence of flag.

Starting & ending flags with bit stuffing

- In this framing techniques **beginning & end of frame with specific bit pattern**
- **01111110**.called as flag byte.
- **Six consecutive 1in the flag called as bit stuffing**
- **If DLL detects the presence of five consecutive ones in a data stream it automatically put 0 bit into outgoing stream**.
- On **reciver side detects the five consecutive ones in a data stream it automatically delete the 0 bit from stream.**



Framing - Starting and Ending Flags with Bit Stuffing

- Each Frame begins and ends with a special bit pattern 01111110 (Flag Byte)
- **Bit Stuffing**
  - If Data has **5** *Consecutive* 1^S → **ADD** **0**

Flag Byte	Data	Flag Byte
01111110	0011011101111110101011111010	01111110

Flag Byte	Data	Flag Byte
01111110	00110111011111**0**10101011111**0**010	01111110

Error detection & correction

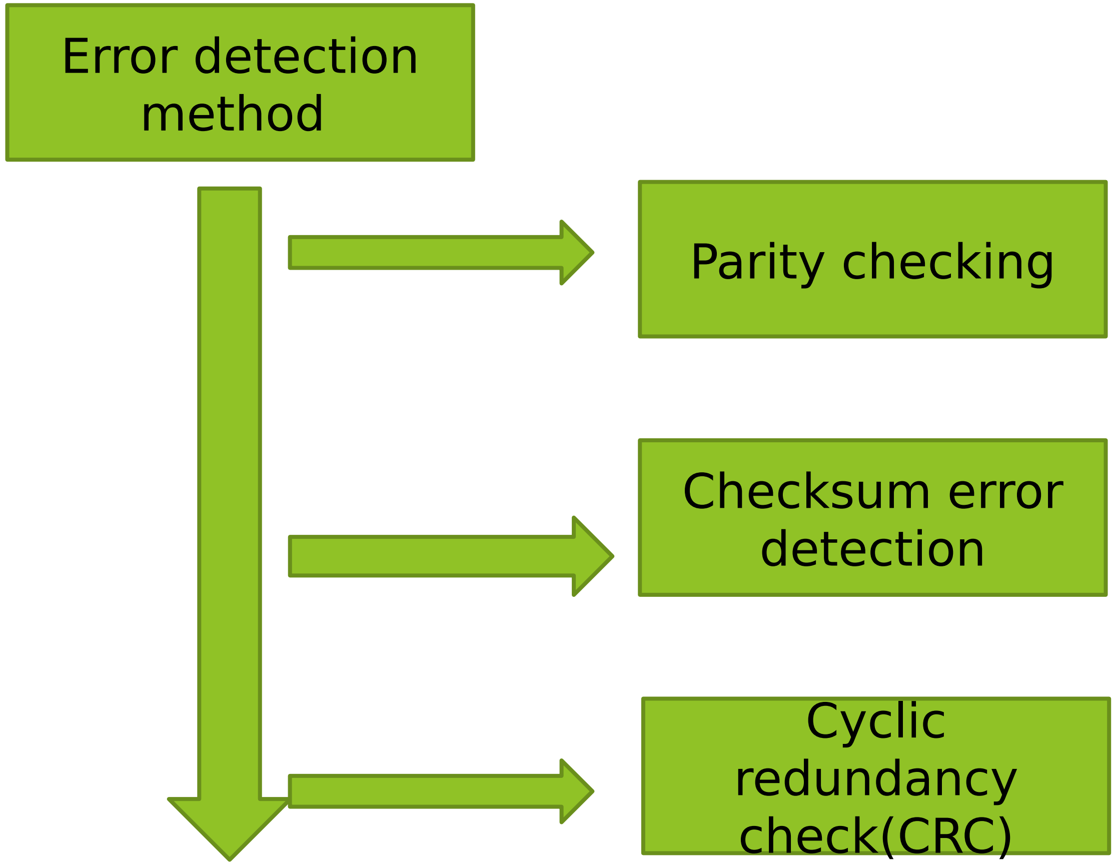
- Transmission of digital signals take place between two machines. But some time noise may occur
- Due to **noise error occur in a transmission**
- Means **0 may change to 1 . 1 may change to 0.**
- So it is **necessary to detect & correct them.**

Classification of error control techniques

- **1)Error detection techniques**
- It is only capable of detecting the errors
- They can not correct the error.

- **2)Error correction techniques**
- They are capable of detection of error & correction of error.
- In error correction multiple process are involved such as detecting the error,knowing their numbers,& location of error. & then correct the error bit

Error detection methods



Error detection method

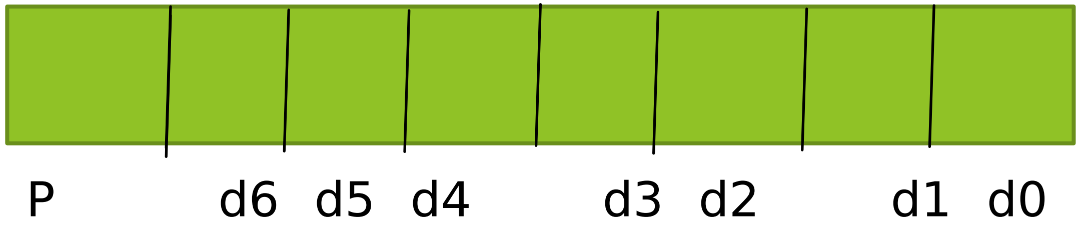
Parity checking

Checksum error detection

Cyclic redundancy check(CRC)

1)Parity checking

- Parity bit or check bit is added to a string to ensure the total number of **1 bit in a string whether even or odd.**
- Addition of parity bit
- The **simplest method of detecting the error** is add extra bit in a data known as **parity bit.**



- Here after **adding the 7 bits as a data .we put P bit as a parity bit.**
- Number of 1s represent even & odd parity
- Even means 2,4,6
- Odd means 1,3,5

How does error detection take place

- Parity can check at receiver side & detect whether error occur or not.
- Parity of transmitted signals is going to be EVEN but at receiver side it shows ODD parity .this means error has to be occur in a transmission.

Transmitted code	0	1 0 0 1 0 1 1 0	even parity
Received code	0	0 0 0 1 0 1 1 0	odd parity
Received code	0	0 0 1 0 0 1 1 0	odd parity

- **If presence of error detected the receiver will ignored the received bytes and request for retransmission.**

Checksum error detection

- Checksum is a error detection scheme ,the data is divided into k segment of m bits
- In sender ends the segments are added using 1s complement .to get the sum .the sum is complemented to get the checksum.
- Now checksum segment is sent along with data segment
- At receiver side receiver segment added using 1s complement arithmetic to get the sum.again sum is complemented.
- If the result is 0.data accepted otherwise rejected.

Sender side

- 10011001
- 11100010
- ________

1 01111011
1
2 1
3 ________
4 01111100
5 00100100
6 ________
7 10100000
8 10000100
9 ________
10 00100100
11 1
12 1
____
13 00100101 final sum 11011010 checksum

100110 01	1110001 0	0010010 0	1000100

K=4,M=8

reciver side

- 10011001
- 11100010
- ________

1 01111011
2 1　　　　1
3 ________
4 01111100
5 00100100
6 ________
7 10100000
8 10000100
9 ________
10 00100100
11 　　　　1
12 ________
13 00100101 final sum
14 11011010 checksum
15 ________
16 11111111

100110 01	1110001 0	0010010 0	1000100

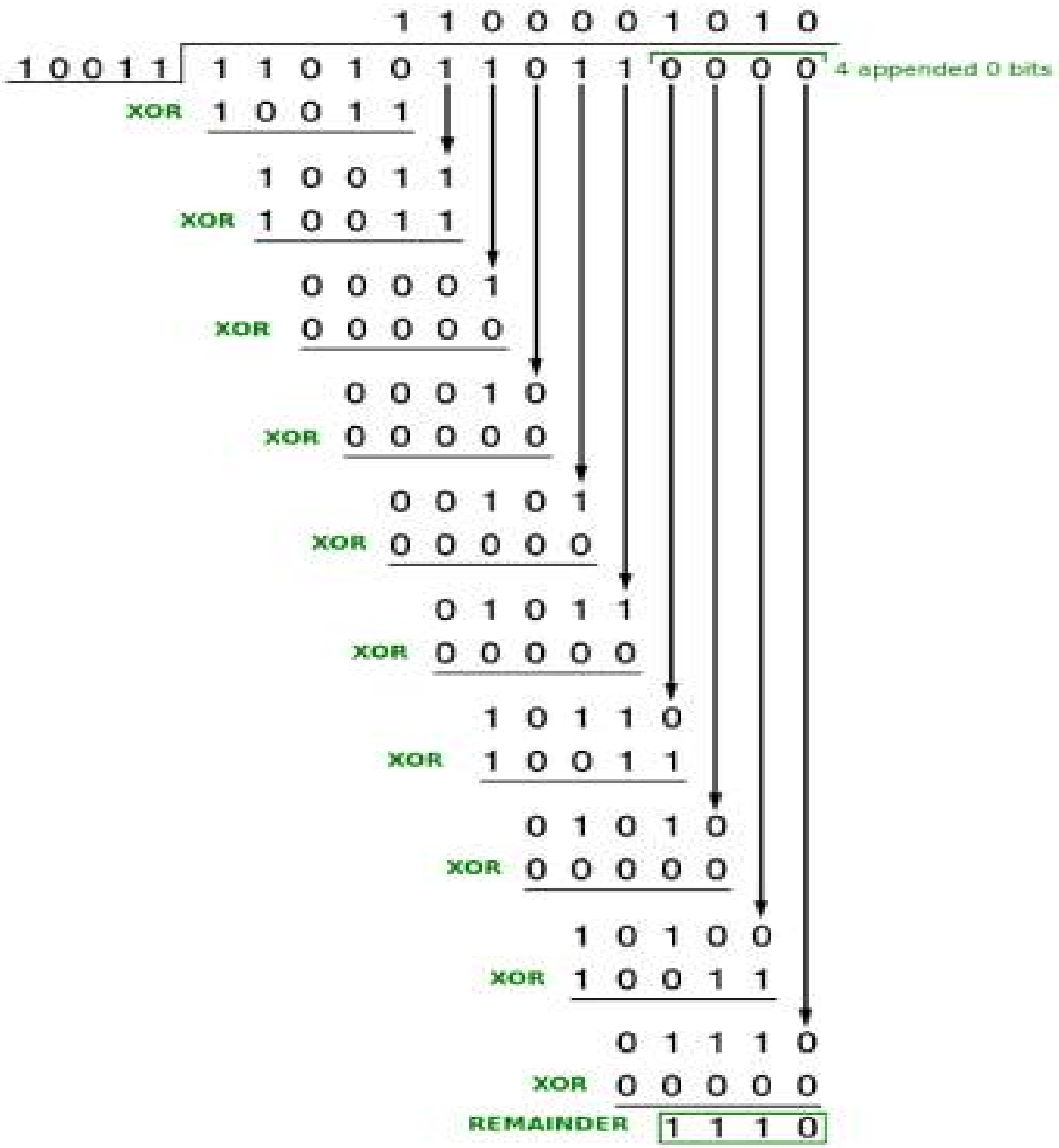
K=4,M=8

00000000=accepted if all 1 then it is rejected

CRC Cyclic redundancy code

- Step 1:
- If the frame is 11010111011 and the generator is $x^4 + x + 1$ what would be the transmitted frame
- Data word: 1101011011 m=10
- Generator:$x^4 + x + 1 = \ x^4 + 0.x^3 + 0.x^2 + x + 1$=10011=5
- N=5
- Obtain the divident=data word+(n-1)zeros
- = 1101011011+4 zeros
- =11010110110000

Step:2



```
                  1 1 0 0 0 0 1 0 1 0
        _____________________________
10011 | 1 1 0 1 0 1 1 0 1 1 0 0 0 0   4 appended 0 bits
    XOR 1 0 0 1 1
          1 0 0 1 1
      XOR 1 0 0 1 1
            0 0 0 0 1
        XOR 0 0 0 0 0
              0 0 0 1 0
          XOR 0 0 0 0 0
                0 0 1 0 1
            XOR 0 0 0 0 0
                  0 1 0 1 1
              XOR 0 0 0 0 0
                    1 0 1 1 0
                XOR 1 0 0 1 1
                      0 1 0 1 0
                  XOR 0 0 0 0 0
                        1 0 1 0 0
                    XOR 1 0 0 1 1
                          0 1 1 1 0
                      XOR 0 0 0 0 0
                REMAINDER   1 1 1 0
```

step3

- Codeword= 1101011011 +1110(reminder)
- The codeword at receiver side consist message & CRC
- The receiver treat it as 1 unit & divides it by the same (n+1).
- The reminder of the division is checked .if reminder is zero then received code word is error free.if reminder is non zero it indicates it has error.
-

Error correction code

- Classification of error code
- 1)Linear block code
- Hamming code
- 2)non-linear code

Hamming Code1001101 m=7

- $2^r \geq m + r + 1$
- $2^1 \geq 7 + 1 + 1 = 8$
- $2^2 \geq 7 + 2 + 1 = 10$
- $2^3 \geq 7 + 3 + 1 = 11$
- $2^4 \geq 7 + 4 + 1 = 12$

1001101 m=7

- Now r=4 bit =P1,P2,P3,P4
- $=2^{0}\quad 2^{1}\quad 2^{2}\quad 2^{3}$

11	10	9	8	7	6	5	4	3	2	1
1	0	0	P4	1	1	0	P3	1	P2	P1
1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001

- P1=3,5,7,9,11 =1,0,1,0,1 =1
- P2=3,6,7,10,11 =1,1,1,0,1 =0

  P3=5,6,7 =0,1,1 =0
- P4=9,10,11 =0,0,1 =1

11	10	9	8	7	6	5	4	3	2	1
1	0	0	1	1	1	0	0	1	0	1
1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001

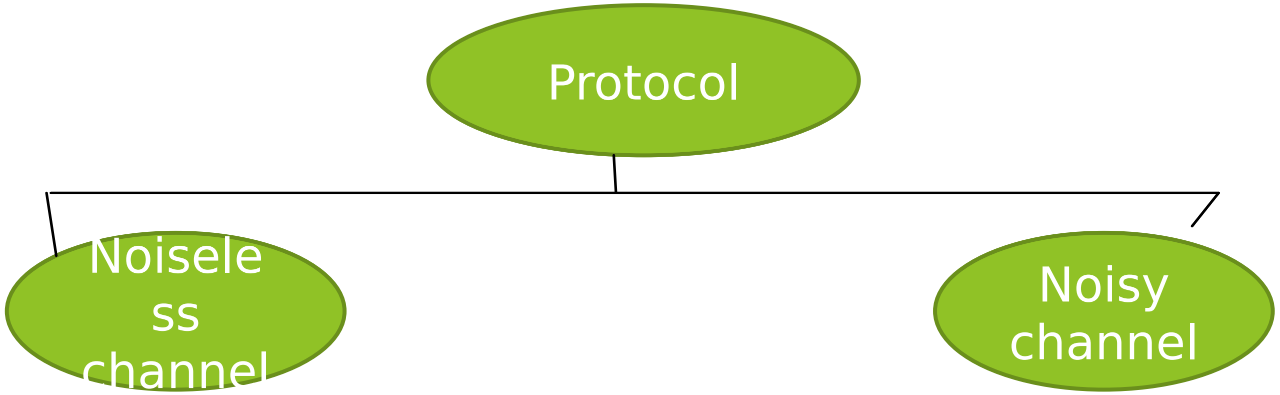
D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1
11	10	9	8	7	6	5	4	3	2	1
1	0	0	1	1	1	0	0	1	0	1
1	0	0	1	0	1	0	0	1	0	1
1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001

P1=1,3,5,7,9,11 =1,1,0,0,0,1 =1
P2=2,3,6,7,10,11 =0,1,1,0,0,1 =1
P3=4,5,6,7 =0,0,1,0 =1
P4=8,9,10,11 =1,0,0,1 =0

0111
D7 denotes error

Flow control

- This is another design issue related to DLL
- sender computer wants to send data at a faster rate than the capacity of receiver
- This happens when sender using faster computer than the reciver.
- The solution to this problem is flow control
- Flow control, control the rate of transmission of frame.
- Sender can send data store in a buffer before goining to reciver . When buffer is full it denotes don’t send data till buffer is empty



1)Simplest
ARQ
2)Stop-and-wait

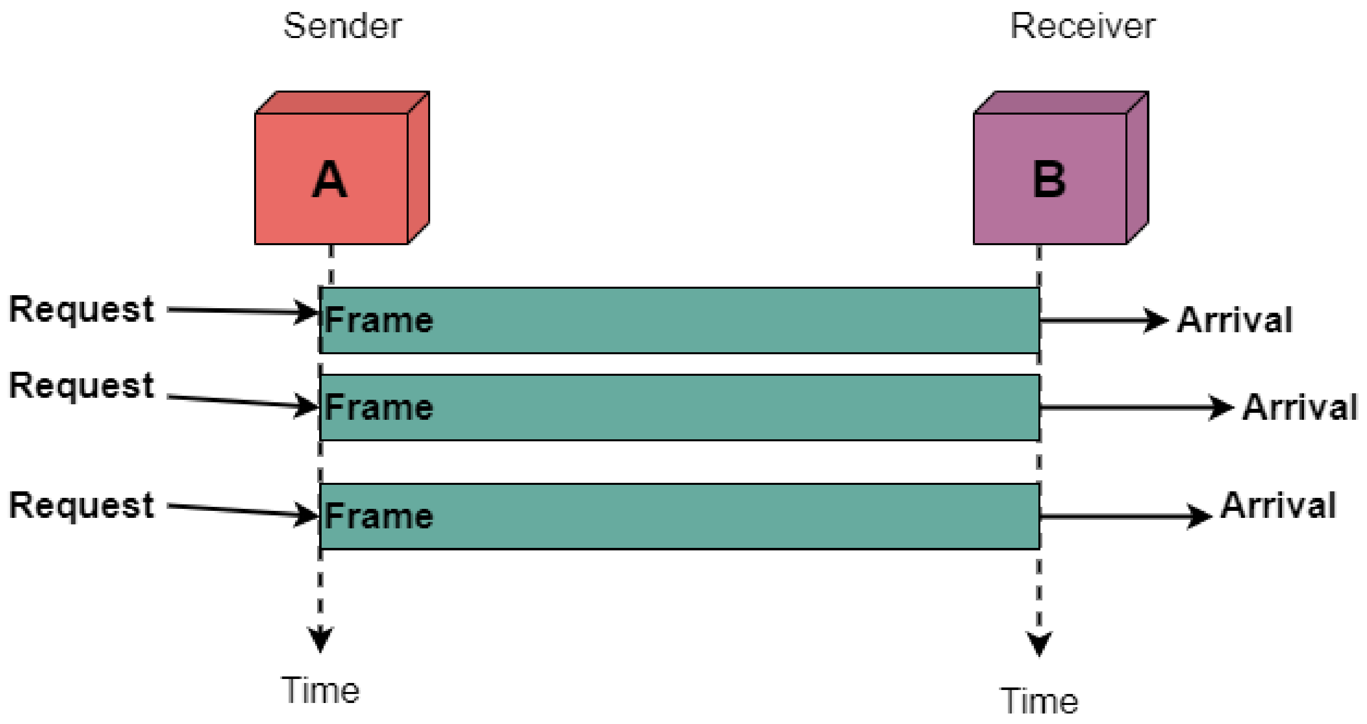
ARQ

1)stop-and-wait

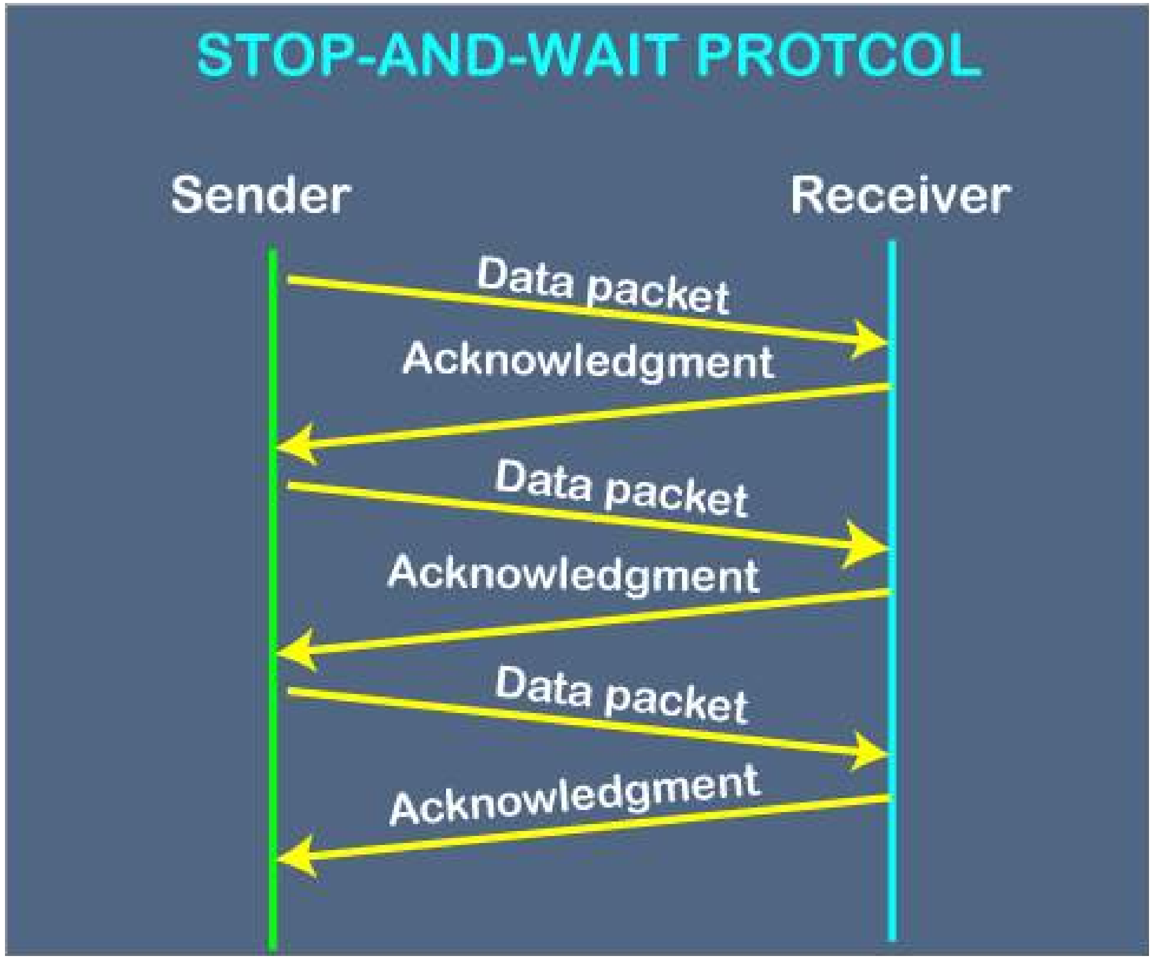
2)Go-back-N-ARQ
3)selective repeat

Elementary data link protocol

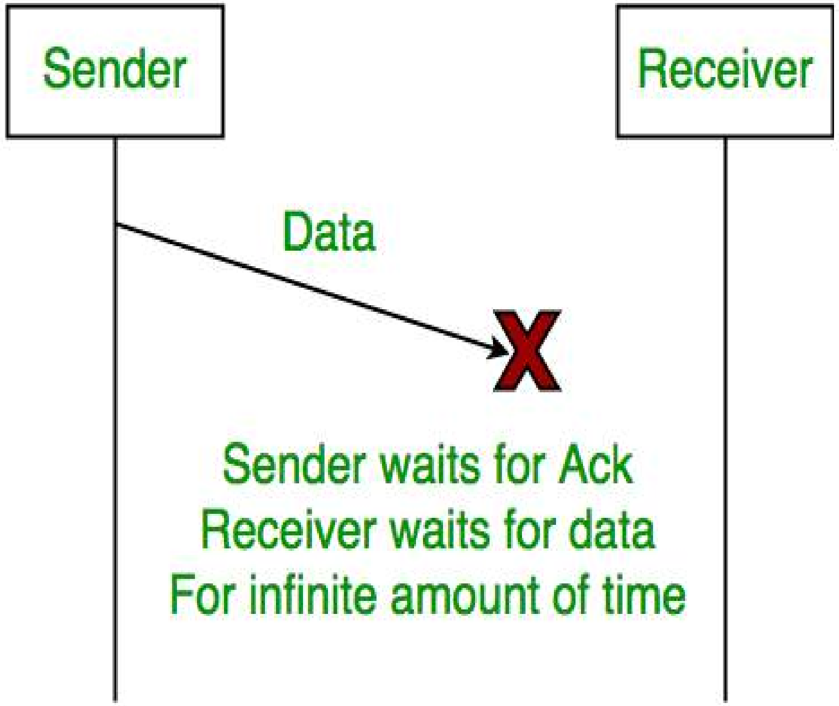
- 1)Unrestricted simplex protocol(noise less channel)
- The transmission of data take place in only only direction
- Also assumed that we can ignore processing time & buffer space available
- The communication channel is imagined to be noise free.



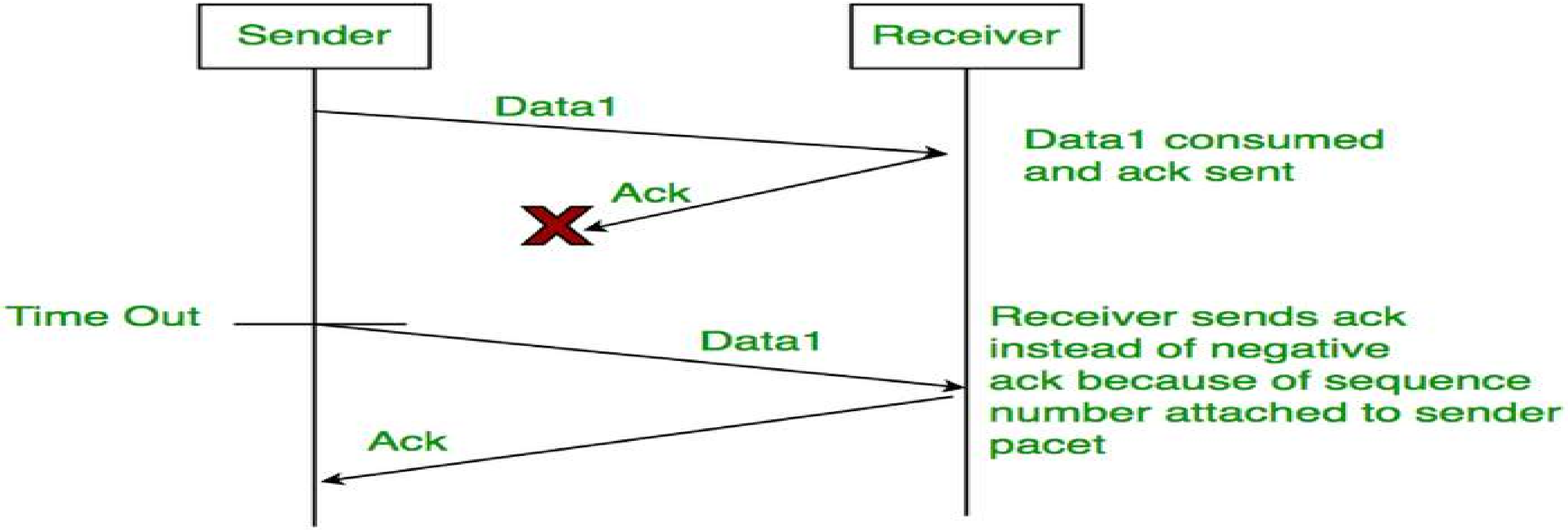
- STOP AND WAIT protocol(noise less channel)
- Stop and wait protocol is data link layer protocol for transmission of frames over noiseless channels
- It provide unidirectional data transmission with flow control facilities .but there is no error control.
- After transmitting one frame the sender waits for an acknowledgement before transmitting the next frame.
- Primitives
- Sender side
- Rule 1:send one data packet at a time
- Rule 2: send the next packet after only after reciving ack.
- Reciver side
- Rule 1:recive & consume data packet
- Rule 2: after consuming packet ack to be send.


STOP-AND-WAIT PROTCOL
Sender
Receiver
Data packet
Acknowledgment
Data packet
Acknowledgment
Data packet
Acknowledgment

- Problems of stop & wait protocol
- 1)problem due to lost data
- Sender wait for ack for infinite amount of time
- Receiver wait for data for infinite amount of time

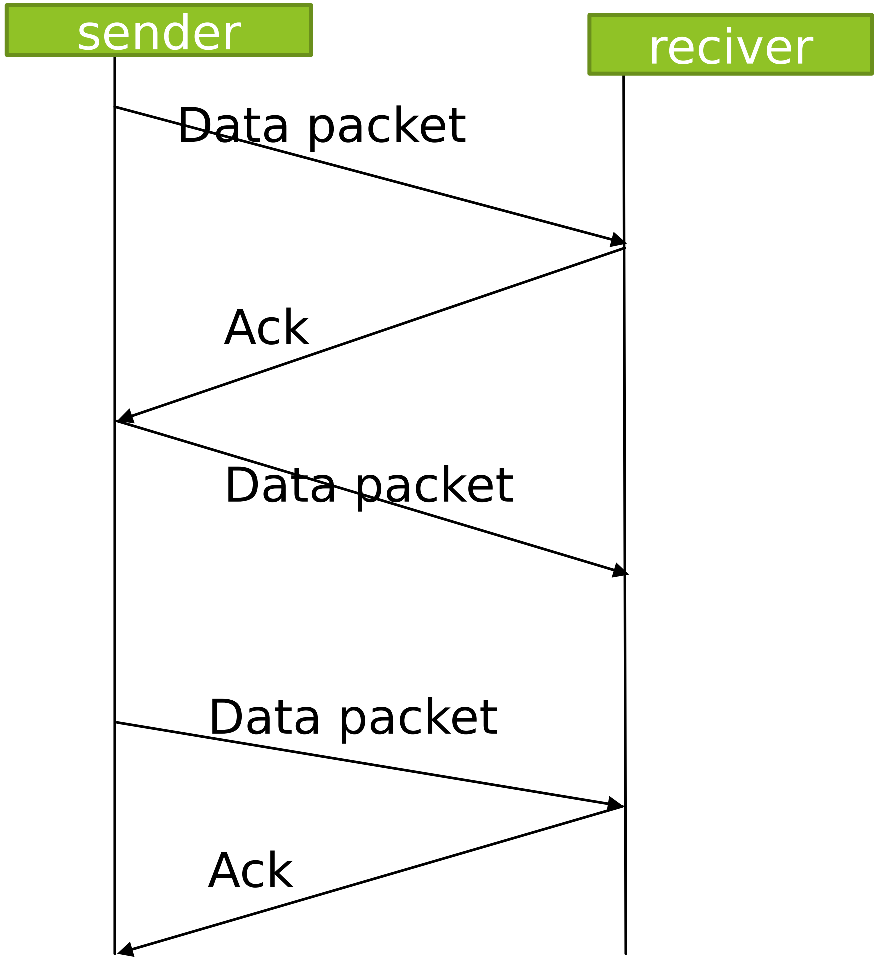


- 2)problem due to lost ack.
- Sender wait for an infinite amount of time for ack.
- 3)Problem due to delay ack
- After time out on sender side a delay ack might be wrongly considered as ack of another packet.


Sender
Receiver
Data1
Data1 consumed and ack sent
Ack
Time Out
Data1
Receiver sends ack instead of negative ack because of sequence number attached to sender pacet
Ack

Piggybacking

Stop & wait ARQ

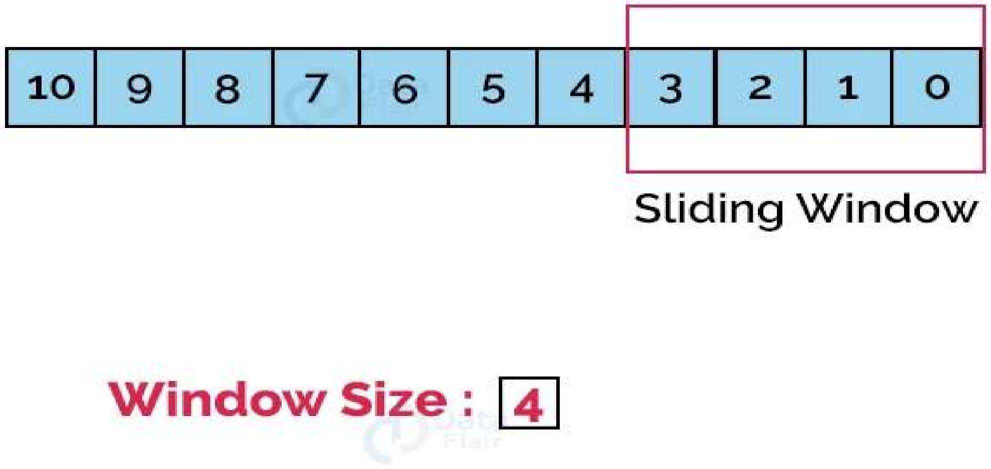


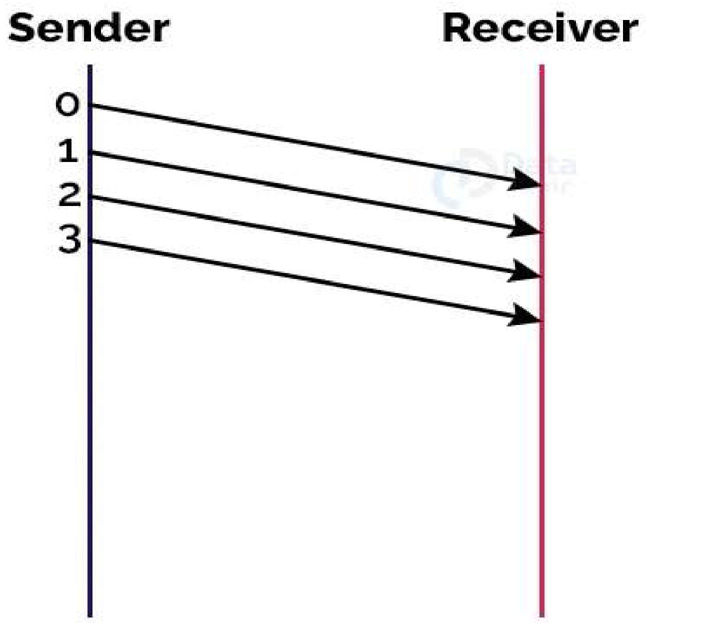
Stop & wait ARQ drawback

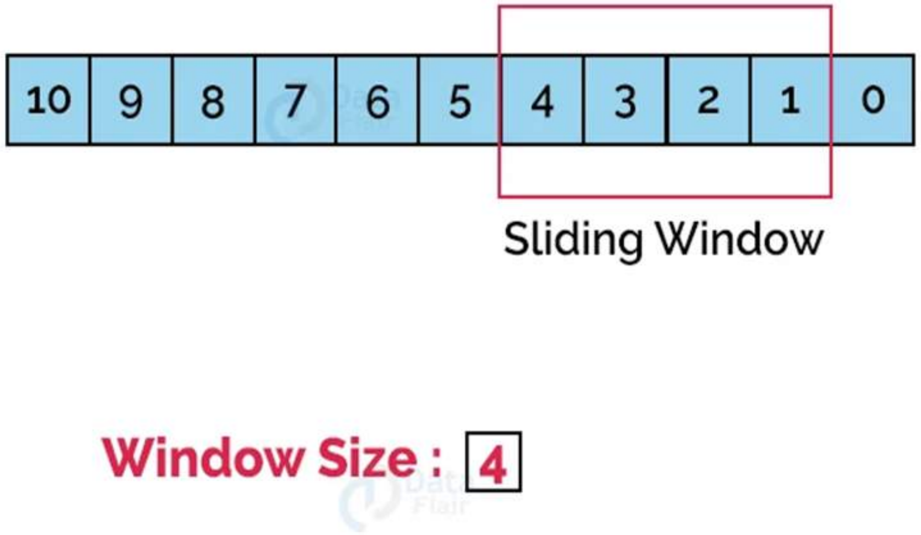
- Sender send the data packet & receiver receive it .& send ack. If receiver not send the ack . The timer on sender side times out.
- Now sender send the the pervious data packet of the frame.wait for ack. If ack is send then no problem.
- But if ack. Is not receive it continue the above procedure again & again.
- One frame at a time
- Poor utilization of bandwidth
- Poor performance

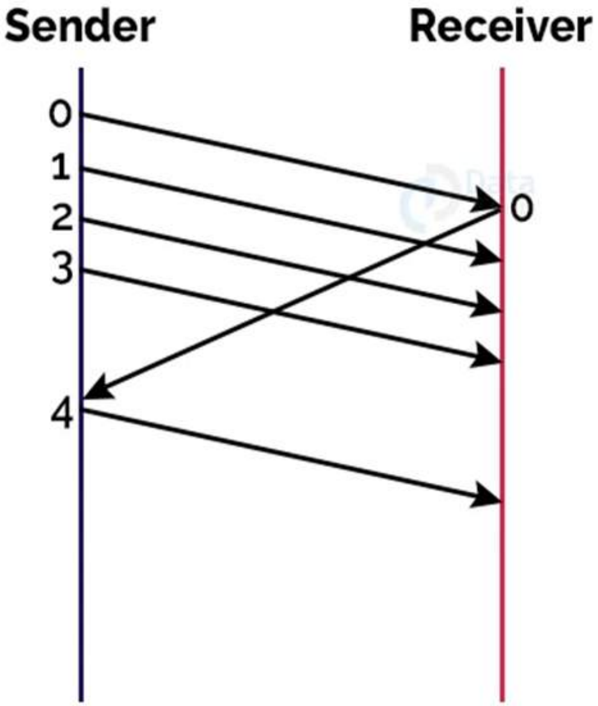
Sliding window protocol

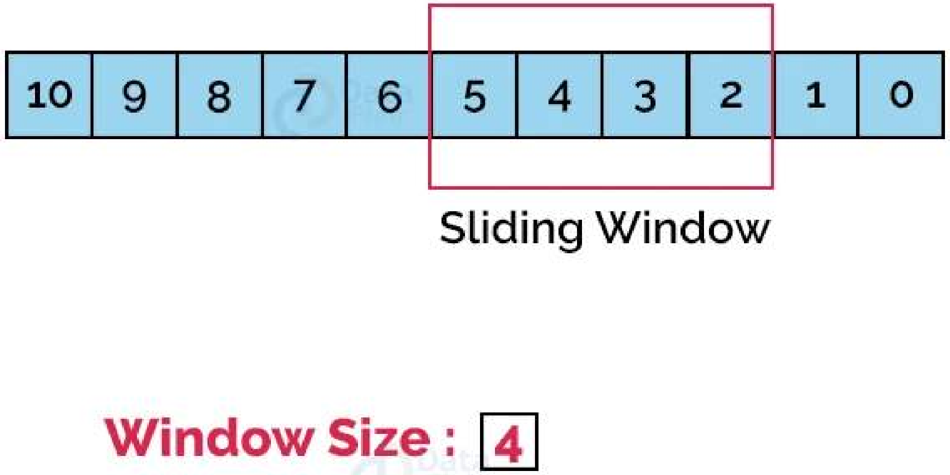
- Send multiple frame at a time
- Number of frames to be sent is based on window size.
- Each frame is numbered which we called as sequence number.

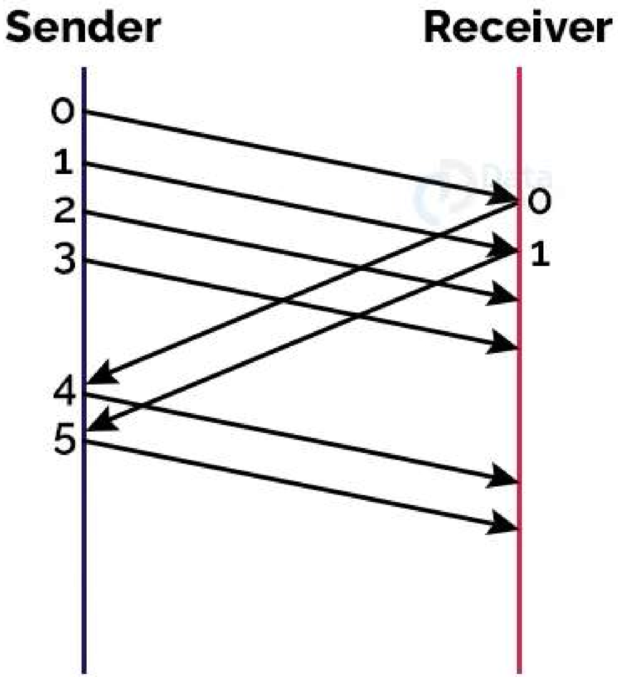





10
9
8
7
6
5
4
3
2
1
0
Sliding Window
Window Size : 4


Sender
Receiver
0
1
2
3
4
0

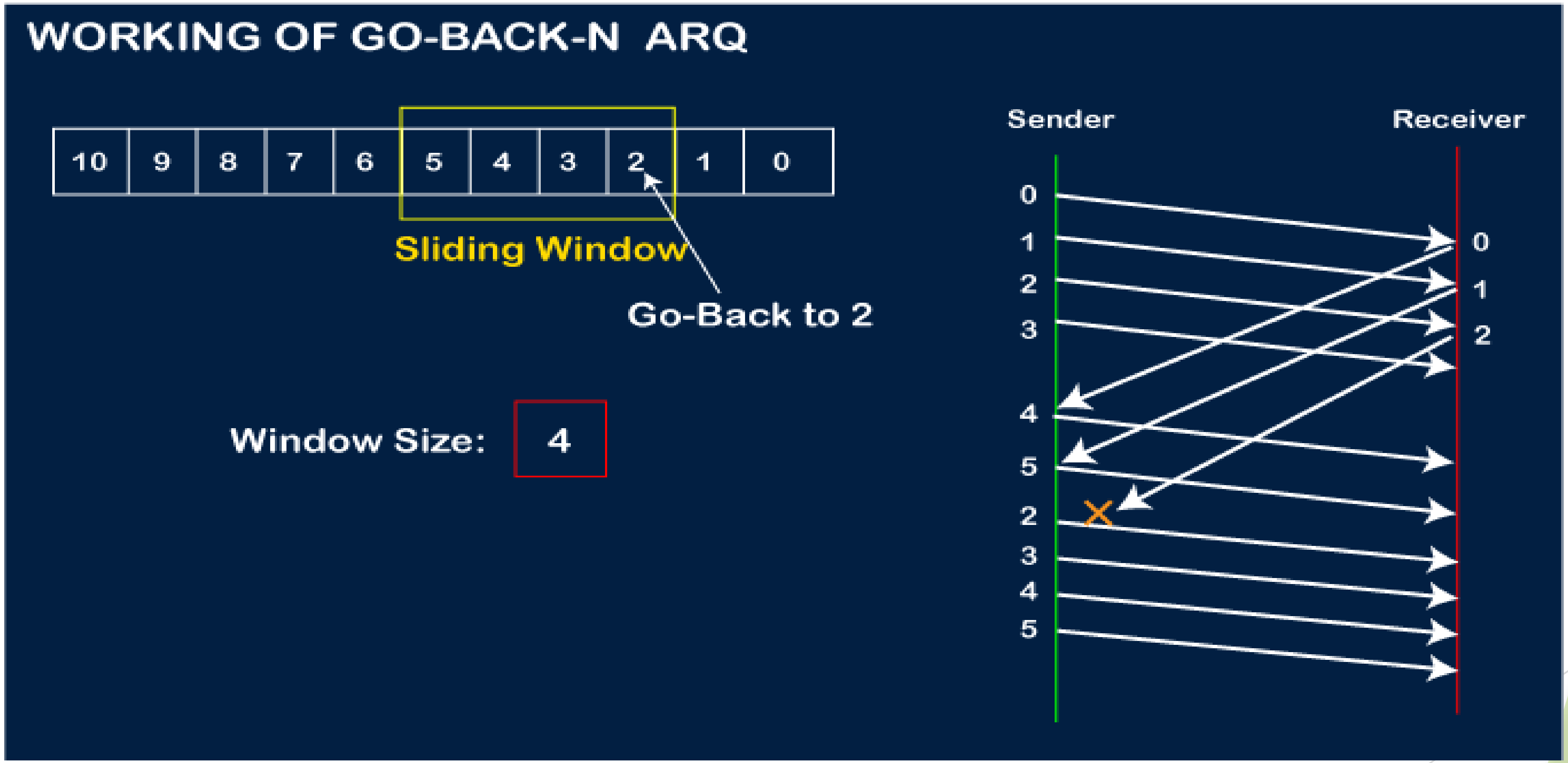

10
9
8
7
6
5
4
3
2
1
0
Sliding Window
Window Size : 4


Sender
Receiver
0
1
2
3
4
5
0
1

Go-Back N ARQ

- Go-Back-N ARQ uses the concept of protocol pipelining i.e the sender can send multiple frames before receiving the ack for the first frame.
- There are finite number of frames and frames are numbered in a sequential manner.
- The number of frames that can be sent depends on the windows size of sender.
- If the ack. Of a frame is not received within an agreed upon time period,all frames in the current windows are transmitted.

- N-sender window size
- For example if the sending size is 4(2^2) then the sequence number will be
- 0,1,2,3,0,1,2,3,0,1 and so on
- The number of bits in the sequence number is 2 to generate the binary sequence 00,01,10,11.

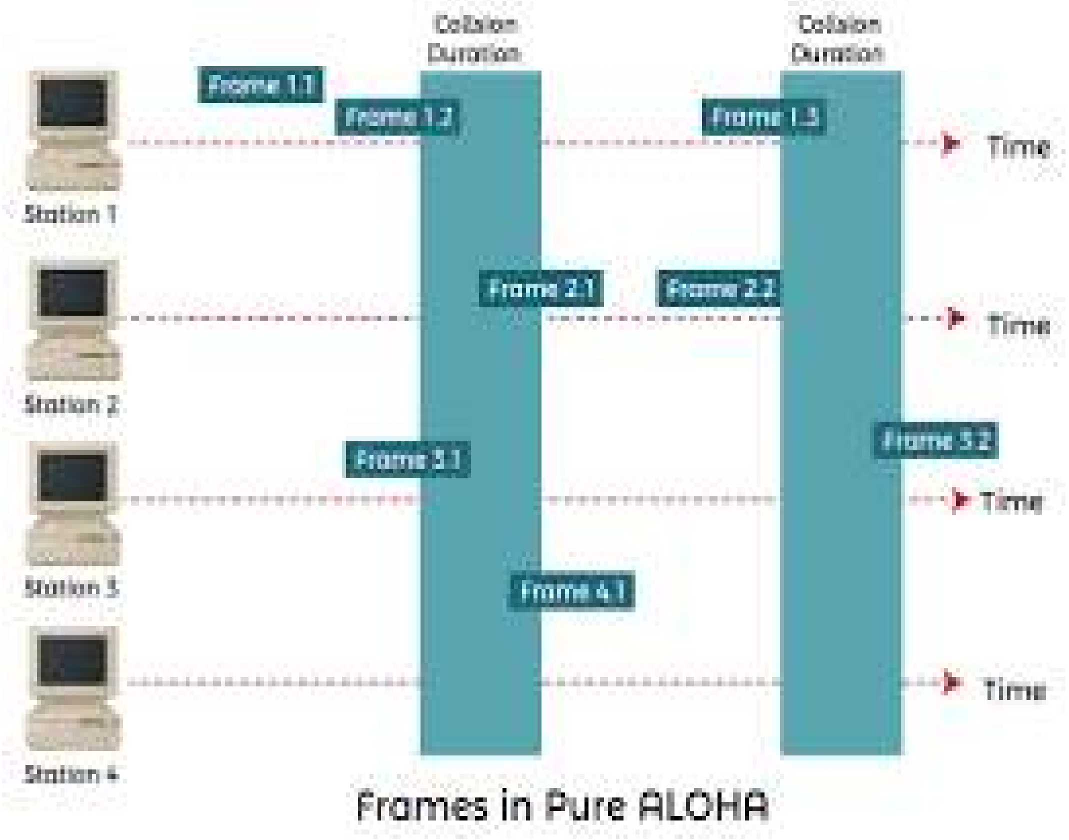


WORKING OF GO-BACK-N ARQ

MAC protocol

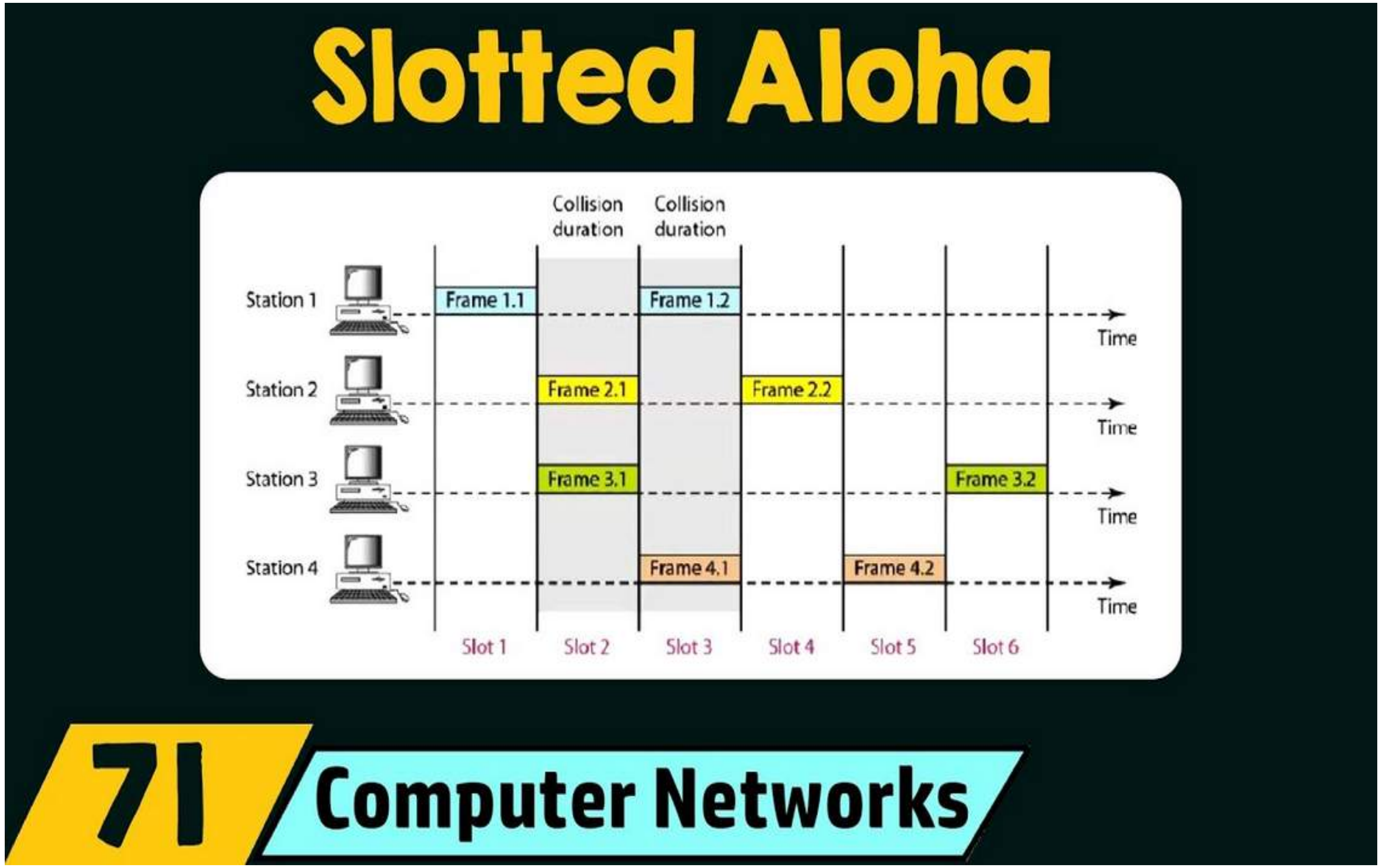
- ALOHA
- 1)PURE ALOHA
- 2)SLOTTED ALOHA

1)Pure ALOHA



Collision Duration
Collision Duration
Frame 1.1
Frame 1.2
Frame 1.3
Time
Station 1
Frame 2.1
Frame 2.2
Time
Station 2
Frame 3.1
Frame 3.2
Time
Station 3
Frame 4.1
Time
Station 4
Frames in Pure ALOHA

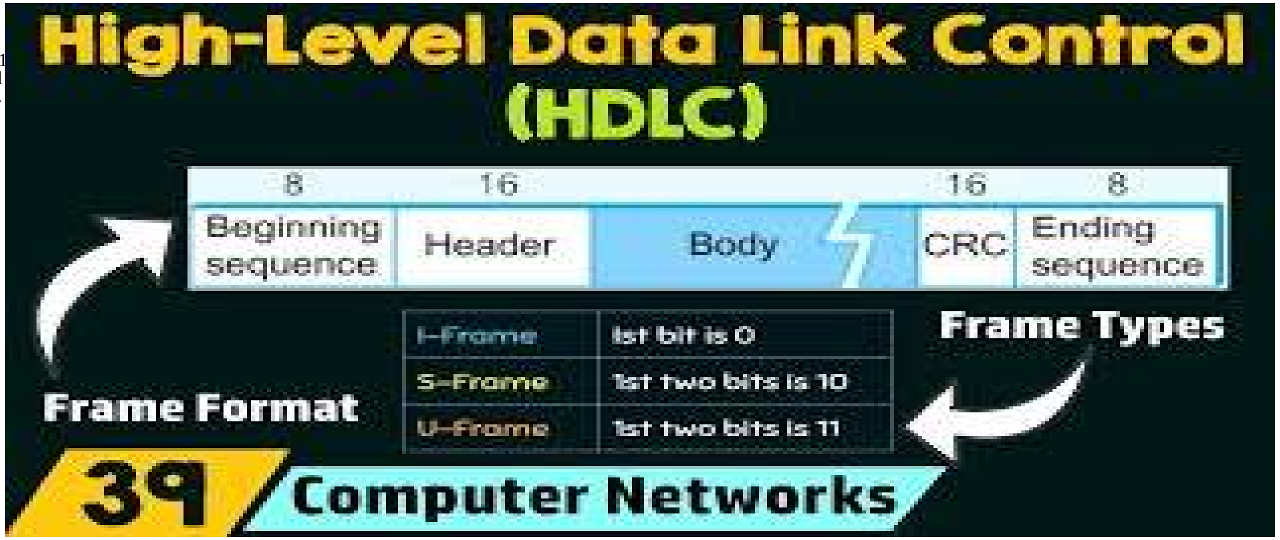
2)SLOTED ALOHA



HDLC

- The synchronous data link control(SDLC)protocol developed by IBM is an example of bit oriented protocol
- SDLCwas later standardized by the ISO as the high level data link control (HDLC)protocol.
- Bit oriented protocol

beginning & ending sequence:01
this sequence is also transmitted
Header:Address & control field.
Body :Payload(variable size)
CRC:error detection



8	16		16	8
Beginning sequence	Header	Body	CRC	Ending sequence

I-Frame	Ist bit is 0
S-Frame	1st two bits is 10
U-Frame	1st two bits is 11

Types of HDLC frames

- The types of frames is determined by the control field
- I-frame:Informative frame
- S-frame:Supervisory frame
- U-frame:Un-numbered frame

I-frame	1st bit is zero	Carry the information
S-frame	1st two bit is 10	Error control& flow control
U-frame	1st two bit is 11	Carry miscellaneous activity(link control)