Data Link Layer

- It is responsible for node-to-node delivery of data.
- It receives the data from network layer and creates FRAMES , add physical address to these frames & pas them to physical layer
- It consist of 2 layers:

Logical Link Layer (LLC) : Defines the methods and provides addressing information for communication between network devices.

Medium Access Control (MAC): establishes and maintains links between communicating devices.

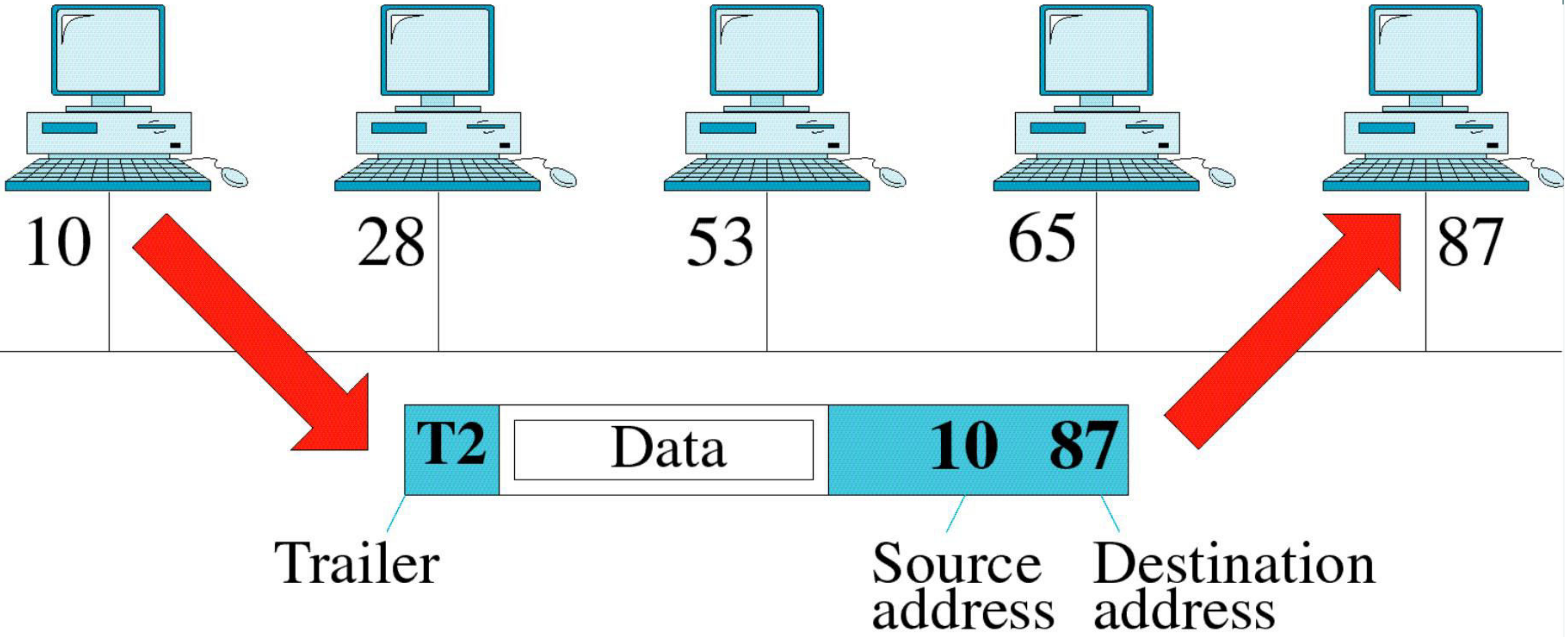
Functions of Data Link Layer

- **Framing :** DLL divides the bits received from N/W layer into frames. (Frame contains all the addressing information necessary to travel from S to D).

- **Physical addressing:** After creating frames, DLL adds physical address of sender/receiver (MAC address) in the header of each frame.

- **Flow Control:** DLL prevents the fast sender from drowning the slow receiver.

Data Link Layer Example



Functions of Data Link Layer

- **Error Control:** It provides the mechanism of error control in which it detects & retransmits damaged or lost frames.

- **Access Control:** When single comm. Channel is shared by multiple devices, MAC layer of DLL provides help to determine which device has control over the channel.

Contents

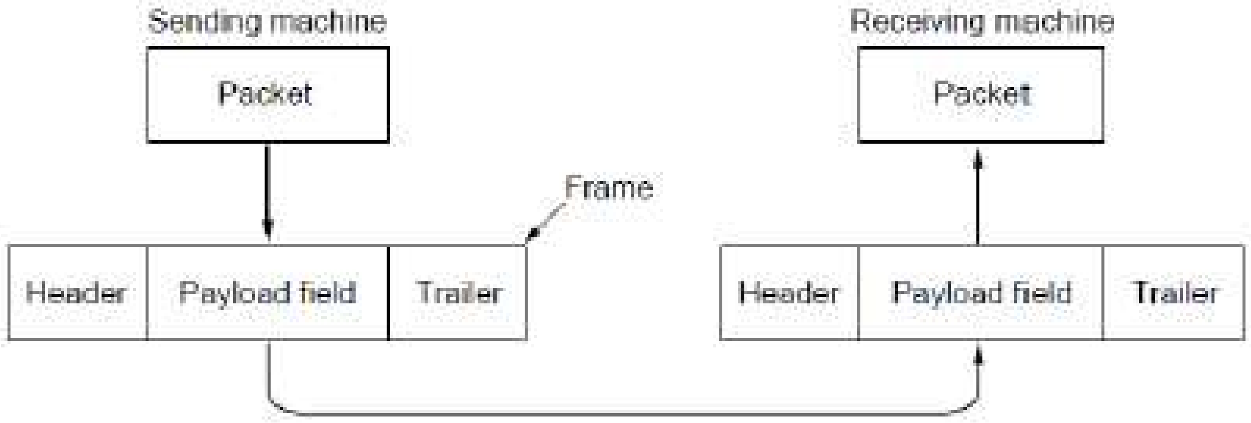
- Sub Layers
  1. LLC
  2. MAC
- MAC Address
- Framing
- Flow Control
  1. Stop and Wait ARQ
  2. Go Back N ARQ
  3. Selective Repeat ARQ
- Error Control Mechanisms
  1. Error Detection
  2. Error Correction
- Channel (Multiple) Access
  1. ALOHA
  2. CSMA
- IEEE 802 Standards
- Virtual Circuit Switching
  1. Frame Relay
  2. ATM
  3. X.25

1.Data Link Layer

- Functions of the data link layer include:
- Providing a well-defined service interface to the network layer (framing)
- Dealing with transmission errors (error control)
- Regulating the flow of data so that slow receivers are not swamped by fast senders (flow control)

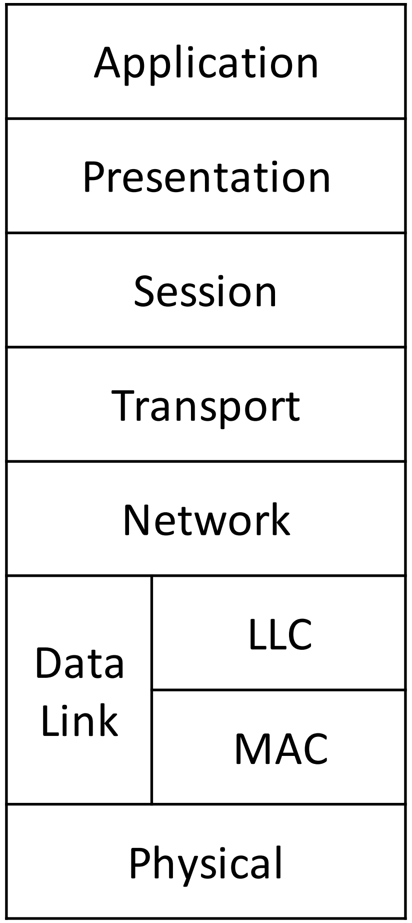
1. Data Link Layer

- To accomplish these goals, packets from the network layers are encapsulated into frames (See Figure Given Below):



1.1Data Link Sub Layers

- Data link layer is divided into 2 sublayers
  1. MAC (Media Access Control)
  2. LLC (Logical Link Control)



Application

Presentation

Session

Transport

Network

Data Link

LLC

MAC

Physical

1.1 Link Sub Layers

1. MAC

- MAC sub layer directly interact with lower layer i.e. Physical layer
- Framing is done in MAC sub layer
- Framing done with help of MAC address

2. LLC

- LLC sub layer directly interact with upper layer i.e. network layer
- Error Control and Flow control is done in LLC sublayer

2 MAC Address

- **M**edia **A**ccess **C**ontrol (**MAC address**), also called physical **address**, is a unique identifier assigned to network interfaces for communications on the physical network segment
- It is used in data link layer communication
- If devices are in same network (LAN) MAC address is used for communication
- MAC address is 48 bit in length i.e. 6 Bytes(Octets)
- It represented using Hexa-decimal Values (6 groups)
- Example : F1-23-45-67-89-AB

2 MAC Address

- First 3 bytes is assigned to specific Organization

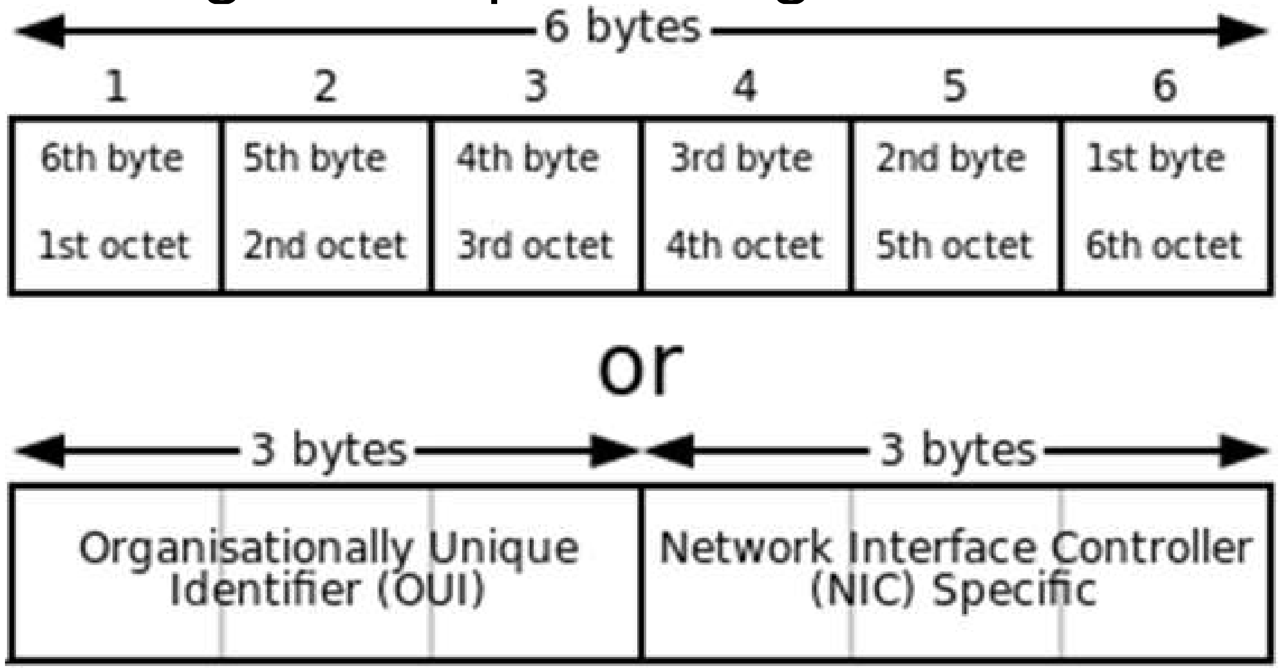


Figure : MAC Address Format

3 Framing

- Data transmission in the physical layer means moving bits in the form of a signal from the source to the destination
- The physical layer provides bit synchronization to ensure that the sender and receiver use the same bit durations and timing
- The data link layer, on the other hand, **needs to pack bits into frames**, so that **each frame is distinguishable from another**
- Framing in the data link layer **separates a message from one source to a destination, or from other messages to other destinations**, by adding a sender address and a destination address

3 Framing

- The **destination address** defines **where the packet is to go**; the **sender address** helps the **recipient acknowledge the receipt**
- **Frames are of 2 types**

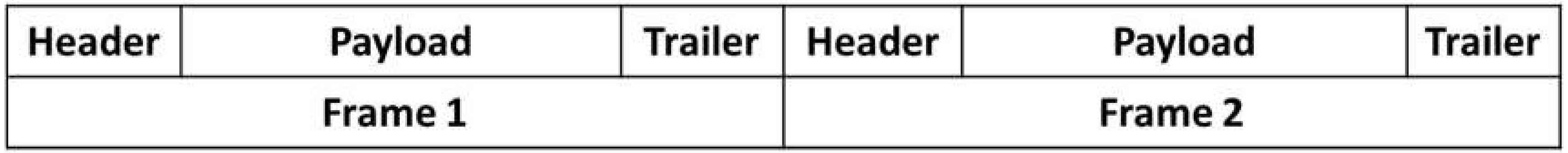
1. **Fixed Size**
2. **Variable**

3.1 Fixed size frame

- Fixed size frames all frames a have same size
- No need for defining frame boundary
- Size itself can be used as a delimiter
- Fixed type of framing is used ATM network
- ATM frame size is 53 bytes (48 for payload +5 for header)

3.2 Variable frame size

- Size of each frames will be different sizes
- In variable-size framing, **Start of frame and end of frame** (i.e. frame boundary) has to be defined
- Two approaches were used for this purpose defining frame boundary:
  1. Character(Byte) -oriented approach
  2. Bit-oriented approach

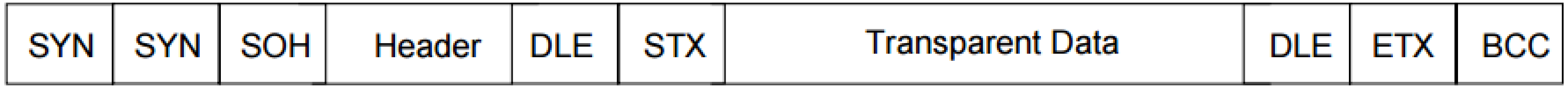


Header	Payload	Trailer	Header	Payload	Trailer
Frame 1			Frame 2		

Figure : Frame Format in Variable Size frame

3.2.1 Character (Byte)- Oriented

- Data to be carried are 8-bit characters from a coding system such as ASCII



SYN	SYN	SOH	Header	DLE	STX	Transparent Data	DLE	ETX	BCC

Figure : Frame in Character Oriented Protocol

- In character oriented protocols, a frame starts with synchronization characters (one or more)
- SYN- Synchronization Idle Character (Usually coded as 0x16)
- SOH- Start of Header(means Start of frame information)

3.2.1 Character (Byte)- Oriented

- STX - Start of Text(Means start of data)
- ETX - End of Text
- ETB - End of Transmission Block
  - If Multiple Frames are send ETB is used in intermediate frames and ETX at the last frame
- BCC – Binary Check Character (for error detection)
- DLE – Data Link Escape(Escape Character/ Flag/ Delimiter)
  - If the DLE character appears in the data field it must be replaced by the sequence DLE DLE **(** Known as **Character stuffing)**

NB :Character Stuffing Section is explained in next slide

3.2.1.1 Character(Byte) Stuffing

- If the DLE(Escape) character appears in the data field it must be replaced by the sequence DLE DLE This Known as **Character stuffing or Byte Stuffing**
- This creates another problem. What happens if the text contains one or more escape characters followed by a flag?
- The receiver removes the escape character, but keeps the flag, which is incorrectly interpreted as the end of the frame
- To solve this problem, the escape characters that are part of the text must also be marked by another escape character.
- In other words, if the escape character is part of the text, an extra Escape Character is added to show that the second one is part of the text

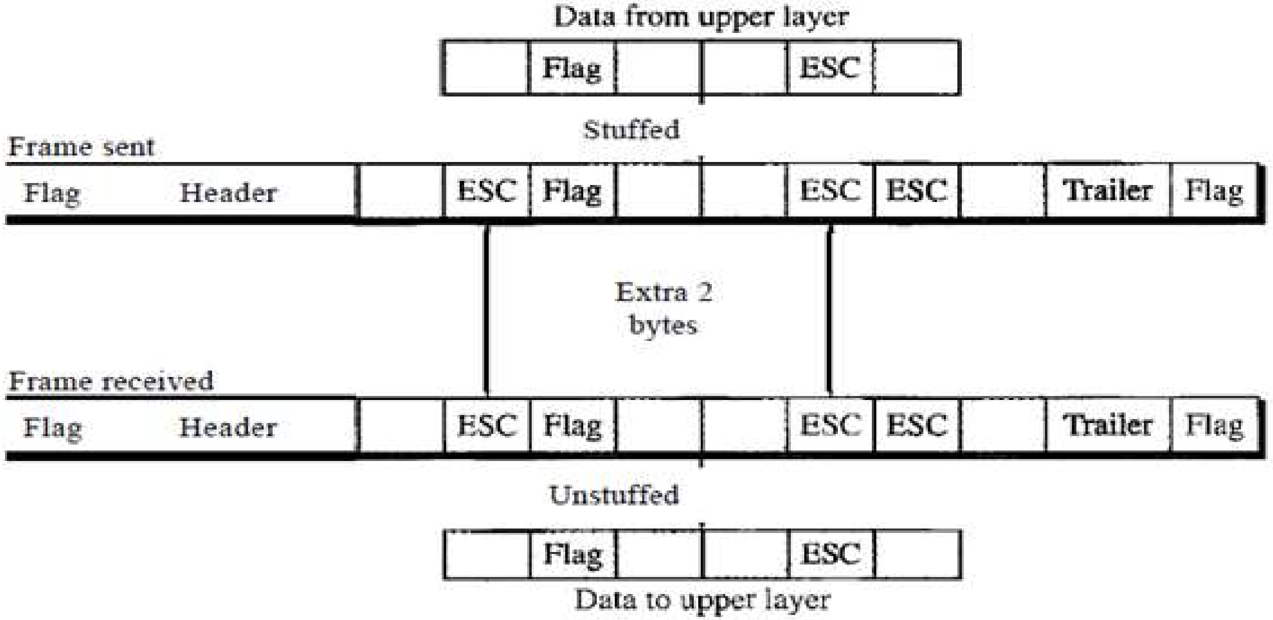
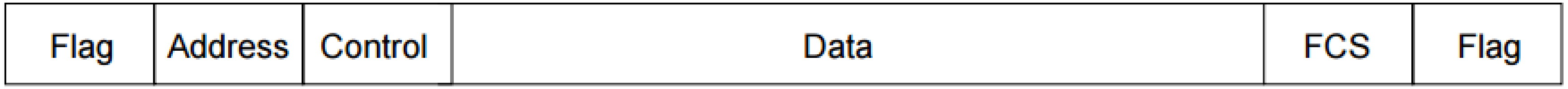


Figure : Byte Stuffing (Sender) and Un-stuffing (Receiver)

3.2.2 Bit Oriented Protocol

- Frame is a collection of bit
- In a bit-oriented protocol, the data section of a frame is a sequence of bits to be interpreted by upper layer data(Text, video, audio, etc..)
- Each frame have **Address, control, data, FCS and Delimiter(Flag)**



Flag	Address	Control	Data	FCS	Flag

Figure 3 : Frame in Bit Oriented Protocol

- Flag (Delimiter)
  - Used to mark start and end of frame(usually 8 bit pattern flag 01111110)
  - One flag is used to separate end and start of next frame if they are contiguous

3.2.2 Bit Oriented Protocol

- **Address** field indicates source and destination address
- **Control** field indicates type or length of frame
- **FCS (Frame Check Sequence)** : For error detection
- **Bit Stuffing:** is the process of adding one extra 0 whenever there is 5 consecutive 1's in frame , so that the receiver does not mistake the pattern 0111110 for a flag.

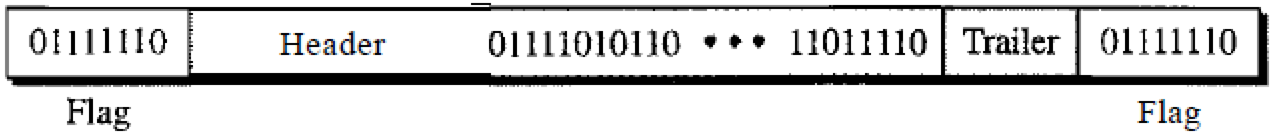


Figure 5 : Bit Oriented Frame with data field having 01111110 pattern flag

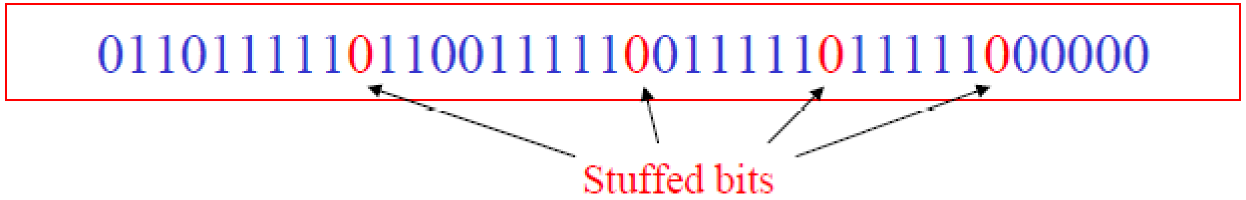
3.2.2.1 Bit stuffing

- Each frame begins and ends with a special bit pattern called a flag byte [01111110]
- Whenever sender data link layer encounters *five consecutive 1's* in the data stream, it automatically stuffs a 0 bit into the outgoing stream
- When the receiver sees *five consecutive incoming 1's followed by a 0 bit*, it automatically dyestuffs the 0 bit before sending the data to the network layer

3.2.2.1 Bit stuffing

0110111111100111110111111111100000

Figure : Original Data in sender



011011111011001111100111110111110000000

Figure: Data sent to receiver after stuffing

0110111111100111110111111111100000

Figure: Data un stuffed after receiving

3.3 Frame format

Preamble(7)	SFD(1)	Destination Address(6)	Source Address(6)	Type/ Length (2)	Data And Padding	CRC (4)

Figure : Frame format (Numbers in each filed indicates size in bytes)

- Ethernet does not provide any mechanism for acknowledging received frames, making it what is known as an unreliable medium
- Acknowledgments must be implemented at the higher layers
- The Ethernet frame contains seven fields as shown in figure

NB : Explanation of Each field is given in next page

3.3.1 Frame format- Preamble

- The first field of frame contains 7 bytes (56 bits)
- Alternating 0’s and 1’s that alerts the receiving system to the coming frame and enables it to synchronize its input timing

01010101010101010101010101010101010101010101010101010101

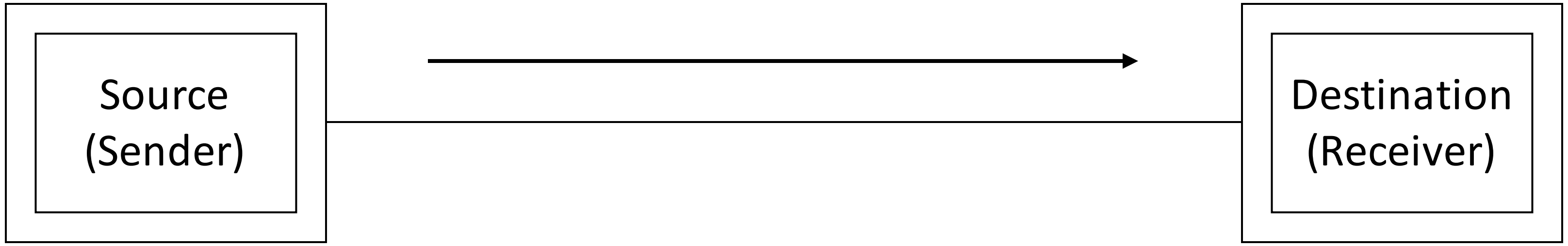
- The pattern provides only an alert and a timing pulse
- The 56-bit pattern allows the stations to miss some bits at the beginning of the frame
- The preamble is actually added at the physical layer and is not (formally) part of the frame.

3.3.2 Frame format- SFD

- The second field is Start Frame Delimiter(SFD) is of 1 byte
- 10101011 (8 bits)
- signals the beginning of the frame
- The SFD warns the station or stations that this is the last chance for synchronization
- The last 2 bits alerts the receiver that the next field is the destination address

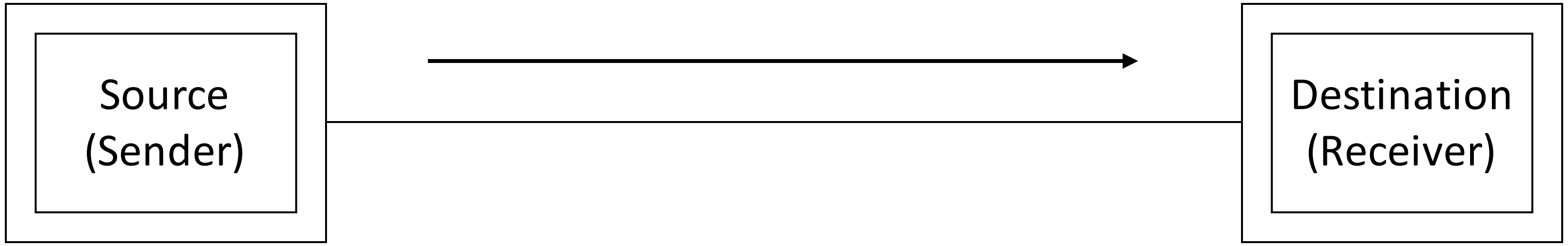
3.3.3 Frame format- Destination Address

- The DA field is 6 bytes
- It contains the physical address(MAC) of the destination station or stations to receive the packet



3.3.4 Frame format- Source Address

- The SA field is 6 bytes
- It contains the physical address(MAC) of the source station or stations to receive the packet

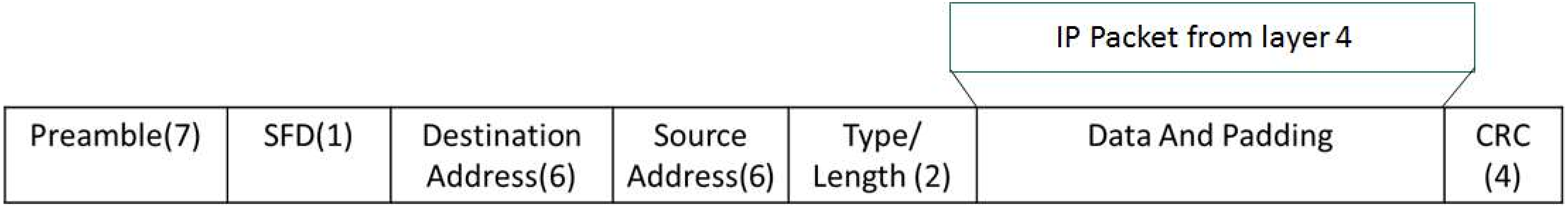


3.3.5 Frame format- Type/Length

- This field is defined as a type field or length field.
- The original Ethernet used this field as the **type field to define the upper-layer protocol using the MAC frame**.
- The IEEE standard used it as the **length field to define the number of bytes in the data field**

3.3.6 Frame format- Data and Padding

- This field carries data encapsulated from the upper-layer protocols.
- It is a minimum of 46 and a maximum of 1500 bytes, as we will see later
- If upper layer data is less than 46 byte add 0’s
- used to insure data is minimum 46 bytes.



IP Packet from layer 4

Preamble(7)	SFD(1)	Destination Address(6)	Source Address(6)	Type/ Length (2)	Data And Padding	CRC (4)

3.3.7 Frame format- CRC

- CRC- Cyclic Redundancy Checking
- For Error control
- It is 4 bytes

4 Error Control

- Error is corruption (change)in bit / bits due to noise, signal distortion or attenuation in the media
- If errors do occur, then some of the bits will either change from 0 to 1 or from 1 to 0
- There are 2 types of error
  1. **Bit error :** Only one bit is corrupted in data
  2. **Burst error:** More than one bits are corrupted in data
- Error Control allows the **receiver to inform the sender** of any frames lost or damaged in transmission and coordinates the **retransmission** of those frames by the sender

4 Error Control

2 ways of error control

1. FEC(Forward Error Correction)
2. ARQ(Automatic Repeat reQuest)

1. FEC

- FEC is accomplished by adding redundancy to the transmitted information using a predetermined algorithm
- Each redundant bit is invariably a complex function of many original information bits

4 Error Control

1. ARQ

- Receiver detects transmission errors in a message and automatically requests a retransmission from the transmitter
- When the transmitter receives the ARQ, the transmitter retransmits the message until it is either correctly received or the error persists beyond a predetermined number of retransmissions (*usually 15*)
- A few types of ARQ protocols are Stop-and-wait ARQ, Go-Back-N ARQ and Selective Repeat ARQ *(We will discuss in Flow control)*

4 Error Control

- Error control is divided into 2 categories
  1. **Error detection**
  2. **Error correction**
- **Error Detection:** It allows a receiver to check whether received data has been corrupted during transmission. If corrupted it can check for retransmission *(Example: Parity Checking and CRC)*
- **Error Correction:** It allows a receiver check for error and to reconstruct the original information when it has been corrupted during transmission *(Example : Hamming code)*

4.1 Error Detection

- It allows a receiver to check whether received data has been corrupted during transmission
- If corrupted it can check for retransmission
- Here ARQ is used for Error correction
- *Error Detection Mechanism*
    1. *Parity Checking (Bit error checking)*
    2. *CRC(Burst error checking)*
    3. *Checksum (Burst error checking)*

4.1.1 Parity Checking

- The simplest error-detection scheme is to append a parity bit to the end of a block of data
- Value of parity bit is selected so that the character has an even number of 1s (even parity) or an odd number of 1s (odd parity)

Data Bits (n)	Parity bit(1)
Message to be Sent = Data + Parity (total n+1 bits)	

Figure : Parity Message format

- Parity can detect odd numbers of errors only.

4.1.1 Parity Checking

- Consider parity encoding scheme with 4 bit data and bit parity
- Code word = data + parity (5 bit)

a3	a2	a1	a0	p

- Parity bit is calculate with the help modulo 2 arithmetic

p=a3 **XOR** a2 **XOR** a1 **XOR** a0 (modulo 2 method)

- Here even parity is used
- If data bits have odd number of 1’s then parity bit will be 1
- If data bits have even number of 1’sthen parity bit will be 0

4.1.1 Parity Checking

Example Scenario

- Consider message 1010 (4 bit)
- Parity bit p is calculating using **XOR**ing (modulo 2 method)
- Transmitter will append parity bit (0 because of even number of 0’s)
- Code word 10100 is send through medium to receiver(message + parity bit , total 5 bits)
- Receiver will check received data and do **XOR**ing (modulo 2 arithmetic) and compare the result with parity bit
- If parity bit in code word and **XOR**ing in receiver matches **NO ERROR**

4.1.2 CRC

- CRC (Cyclic Redundancy Checking)
- CRC is based on polynomial
- Sender and receiver have to choose a common polynomial for checking error
- Instead of one parity bit here **R parity bits** are used for error checking
- The **value of R** is determined by the degree of polynomial selected
- Data word(N bits) + parity word(R bits) = code word (K bits)

4.1.2 CRC

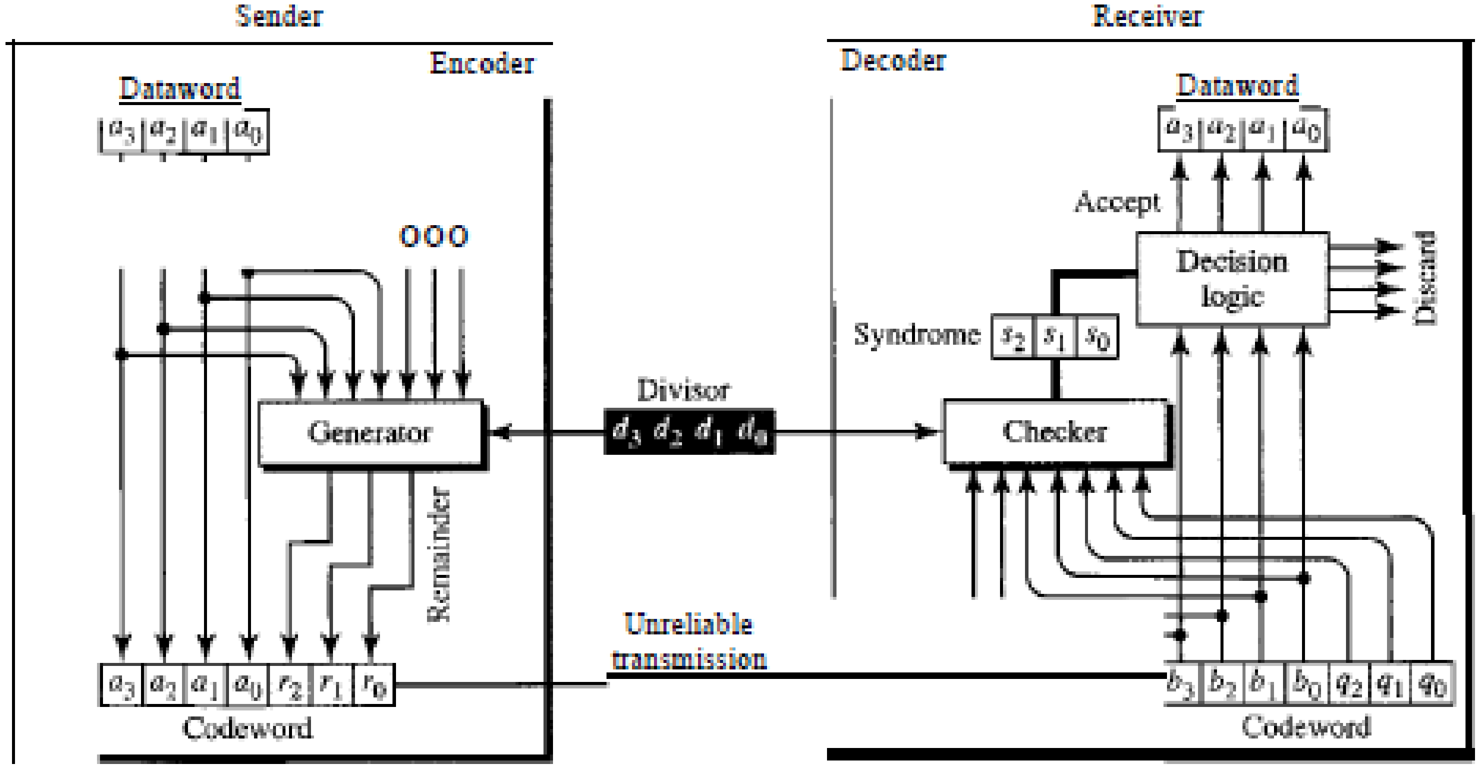


Figure : CRC Encoder/Decoder

4.1.2.1 CRC – Encoder (Sender)

- Consider
- Data word(message) =1001 (4 bit)
- Polynomial =x3+x+1(i.e. in binary representation 1011)
- The above polynomial is of degree 3 (so 3 parity bit is used)
- Code word (k)= 7 bit in total
- Code word is generated with help of data bit and polynomial

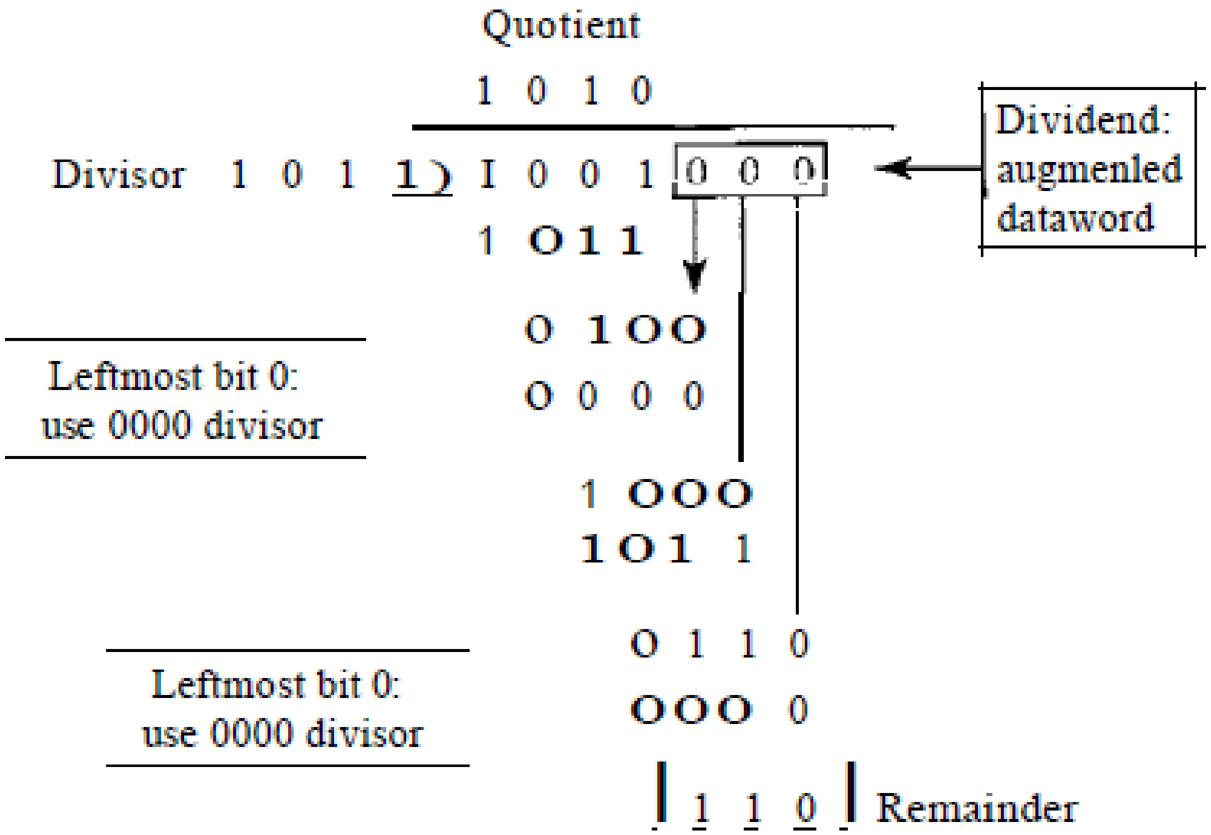
4.1.2.1 CRC – Encoder (Sender)

Steps in encoder

1. First add R bits of 0’s in the with the data word
2. 1001 is data word and 3 0’s augmented(added) so 1001000 is generated
3. 1001000 is called augmented data word
4. 1001000 is divided with the polynomial bits i.e. 1011(divisor)
5. 110 is obtained as reminder and quotient is discarded
6. 110 is augmented with data bits i.e. 1001 to create code word i.e. 1001110
7. Code word is sent through the transmission media

NB: Division is shown in next page

4.1.2.1 CRC – Encoder (Sender)



Quotient

1 0 1 0

Divisor 1 0 1 1) 1 0 0 1 0 0 0

Dividend: augmenled dataword

1 0 1 1

0 1 0 0

Leftmost bit 0: use 0000 divisor

0 0 0 0

1 0 0 0

1 0 1 1

0 1 1 0

Leftmost bit 0: use 0000 divisor

0 0 0 0

1 1 0 Remainder

4.1.2.1 CRC – Decoder (Receiver)

- Received Code word(message + parity) =1001110 (7 bit)
- Have same Polynomial =x3+x+1(i.e. in binary representation 1011)
- The above polynomial is of degree 3 (so 3 parity bit is used)
- Receiver will check for error by doing division with the code word received and polynomial as divisor
- If the remainder is 000 then there is no error in transmission else there is error in transmission.

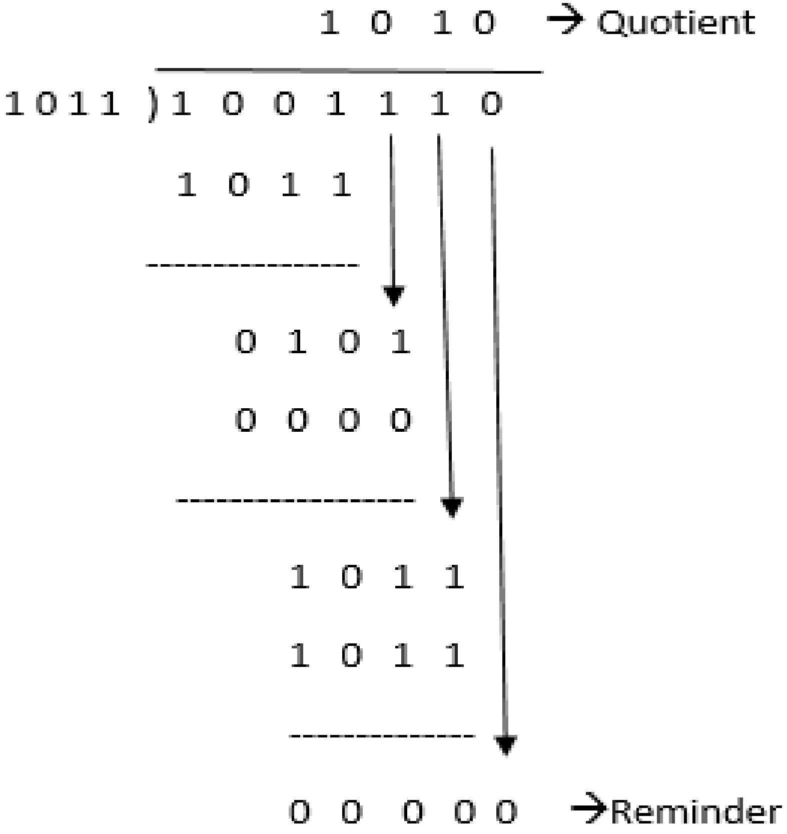
4.1.2.1 CRC – Decoder (Receiver)

Steps in Decoder.

1. Code word (1001110) is received through the transmission media
2. First divide (binary modulo 2 division) the code word the polynomial bits i.e. 1011(divisor)
3. After division discard the quotient and take reminder only
4. If reminder is 000 then there is no error in transmission

NB : Division is shown in next page

4.1.2.1 CRC – Decoder (Receiver)



```
              1 0  1 0   → Quotient
       ___________________
1011 ) 1 0 0 1 1 1 0
       1 0 1 1
      -----------
         0 1 0 1
         0 0 0 0
       -----------
           1 0 1 1
           1 0 1 1
           ---------
           0 0  0 0 0   →Reminder
```

4.2. Error Correction Codes

- It allows a receiver check for error and to reconstruct the original information when it has been corrupted during transmission
- It is also called FEC (Forward Error Correction)
- Hamming Code is Error correction code
- Error correction done with help of Hamming Distance

4.2.1. Hamming Distance

- Given any two code words that may be transmitted or received—say, 10001001 and 10110001 respectively
- To determine how many bits differ(error), just XOR the two code words and count the number of 1 bits in the result

1 0 0 0 1 0 0 1 XOR

1 0 1 1 0 0 0 1

----

0 0 1 1 1 0 0 0 → Here 3 bits One So 3 error bits(Hamming Distance = 3)

4.2.2. Minimum Hamming Distance

- The minimum Hamming distance is the smallest Hamming distance between all possible pairs in a set of words
- First find all distances and find the minimum distance
- For 2 bit word all possible combinations are (00,01,10,11)

d(00,01)=1 d(00,10)=1 d(00,11)=2

d(01,10)=2 d(01,11)=1 d(10,11)=1

Here minimum distance for 2 bit word is 1

4.2.3. Hamming Code

- Hamming code can be used to check and correct errors
- Hamming code works based on minimum hamming distance(d_{min})
- Consider Hamming code(n,k)
  - n- no of code bits
  - k- no of data bits
  - r- no of redundant bit(Bits added with data For Error detection and correction)
  - n =k+r
- *With the help of below equation we can calculate the number of redundant bit for given data bit*

$$2^r \geq k + r + 1$$

4.2.4. Hamming Code(7,4)

- Consider example hamming code(7,4)
- No of data bits (k)=4
- Calculating redundant bits with help of formula $2^r \geq k + r + 1$
    - Consider r=1 → $2 \geq 4 + 1 + 1\ (Not\ Feasible)$
    - Consider r=2 → $4 \geq 4 + 2 + 1\ (Not\ Feasible)$
    - Consider r=3 → $8 \geq 4 + 3 + 1\ (Feasible)$
- For data (k) bits =4 we need 3 redundant (r) bits
- So total number bits in code word (n)=4+3 (k+r)=7
- In this example Minimum hamming distance is $d_{min} = 3$

4.2.4.1. Hamming Code(7,4) Encoder

- Consider 4 data bits as (d_0, d_1, d_2, d_3) and 3 redundant bits(r_0, r_1, r_2)
- Calculation of redundant bits (Modulo 2 addition / XOR)
- r_0=$d_3 + d_2 + d_1$
- r_1=$d_3 + d_2 + d_0$
- r_2=$d_3 + d_1 + d_0$

4.2.4.1. Hamming Code(7,4) Encoder

Consider message 0011 sent using hamming bit(order is d3,d2,d1,d0)

Find the redundant bits with help of formula given in previous page

- r_0=$d_2 + d_1 + d_0 = 0 + 1 + 1 = 0$
- r_1=$d_3 + d_2 + d_1 = 0 + 0 + 1 = 1$
- r_2=$d_3 + d_1 + d_0 = 0 + 1 + 1 = 0$

d_0	d_1	d_2	d_3
1	1	0	0
Data bits			

r_0	r_1	r_2
0	1	0
Redundant Bits		

- Code word=0011010($d_3\ d_2\ d_1\ d_0\ r_2\ r_1\ r_0$)
- This code word is transmitted to receiver

4.2.4.2. Hamming Code Decoder

- Here 7 bit code word is received and receiver has to do error checking and correction
- Syndrome bit is used for error correction
- First step receiver will find the number of syndrome bits
- Here number of code word position in which error might occur is 7 and other condition is no error in code word (total 8 conditions)
- So we can consider 3 syndrome bits which will produce 8 combinations (000 to 111)

4.2.4.2. Hamming Code Decoder

- Calculation of syndrome bits
- (Modulo 2 addition / XOR)
- $s_0 = r_0 + d_3 + d_2 + d_1$
- $s_1 = r_1 + d_3 + d_2 + d_0$
- $s_2 = r_2 + d_3 + d_1 + d_0$

If there is error in transmission syndrome matrix will help to find which bit the error is located and corrected by receiver

s_2	s_1	s_0	Error Bit	Remark
0	0	0	No Error	
0	0	1	r_0	r_0 *only in* s_0
0	1	0	r_1	r_1 *only in* s_1
0	1	1	d_2	d_2 *in* s_0, s_1
1	0	0	r_2	r_2 *only in* s_2
1	0	1	d_1	d_1 *in* s_0, s_2
1	1	0	d_0	d_0 *in* s_2, s_1
1	1	1	d_3	d_3 *in* s_0, s_1, s_2

Syndrome matrix in receiver

4.2.4.2. Hamming Code Decoder (No Error Condition)

- Consider the message we encoded in the Encoder section i.e. 0011010($d_3\ d_2\ d_1\ d_0\ r_2\ r_1\ r_0$) received same message
- Syndrome bit 0 means no error in that bit, 1 means error present

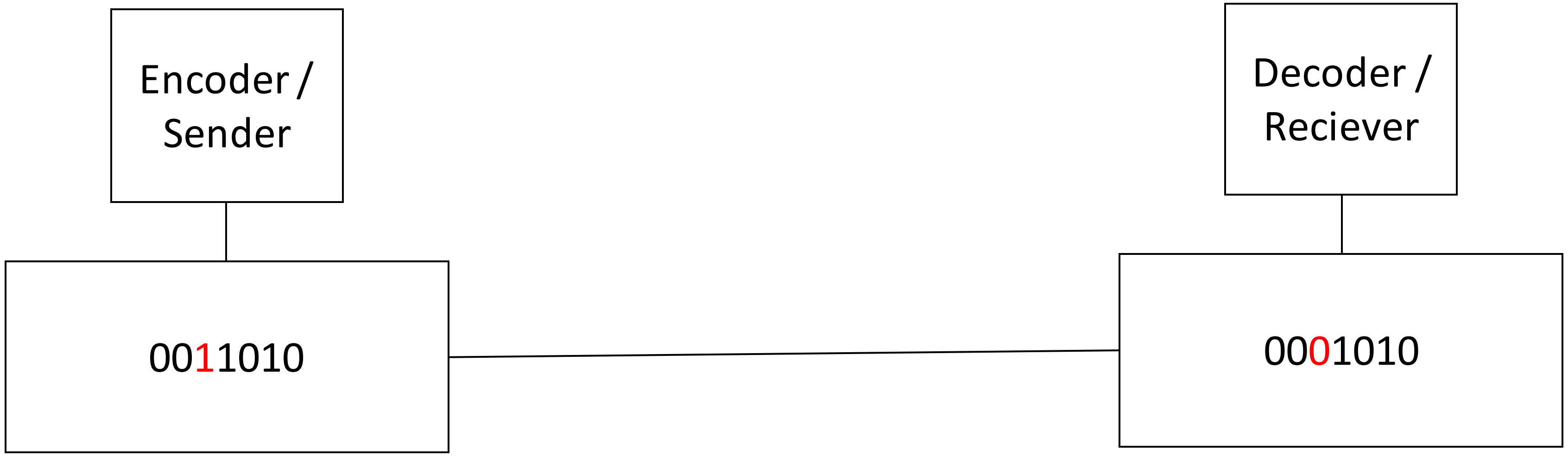
Encoder / Sender
0011010
Decoder / Reciever
0011010

4.2.4.2. Hamming Code Decoder (Error Condition)

- First we have to find the syndrome bits
  - $s_0 = r_0 + d_3 + d_2 + d_1 = 1 + 0 + 0 + 1 = 0$
  - $s_1 = r_1 + d_3 + d_2 + d_0 = 1 + 0 + 0 + 1 = 0$
  - $s_2 = r_2 + d_3 + d_1 + d_0 = 0 + 0 + 1 + 1 = 0$
- Syndrome bit 000(s2,s1,s0) means no error (From syndrome matrix) present in the received data

4.2.4.2. Hamming Code Decoder (Error Condition)

- Consider an error condition left most 3rd bit is changed in transmission
- 0011010($d_3\ d_2\ d_1\ d_0\ r_2\ r_1\ r_0$) send message
- 0001010($d_3\ d_2\ d_1\ d_0\ r_2\ r_1\ r_0$) received message



4.2.4.2. Hamming Code Decoder (Error Condition)

- First we have to find the syndrome bits
  - $s_0 = r_0 + d_3 + d_2 + d_1 = 1 + 0 + 0 + 0 = 1$
  - $s_1 = r_1 + d_3 + d_2 + d_0 = 1 + 0 + 0 + 1 = 0$
  - $s_2 = r_2 + d_3 + d_1 + d_0 = 0 + 0 + 0 + 1 = 1$
- Syndrome bits are 101(s2,s1,s0)
- From the syndrome matrix error presents in bit which is present in s2,s0 and not present on s1, d1 is commonly present in calculation of s2 and s0 not in s1
- Negate the d1 bit for correcting the error.

5. Flow Control

- There are 2 techniques of Error correction
  1. FEC (Forward Error correction) – Using Hamming codes
  2. ARQ (Automatic Repeat reQuest)– Resending of data
- In noisy Channel error control is achieved with help of ARQ which is a flow control mechanism
- Flow control is a technique for assuring that a transmitting entity does not overwhelm a receiving entity with data
- Flow control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver

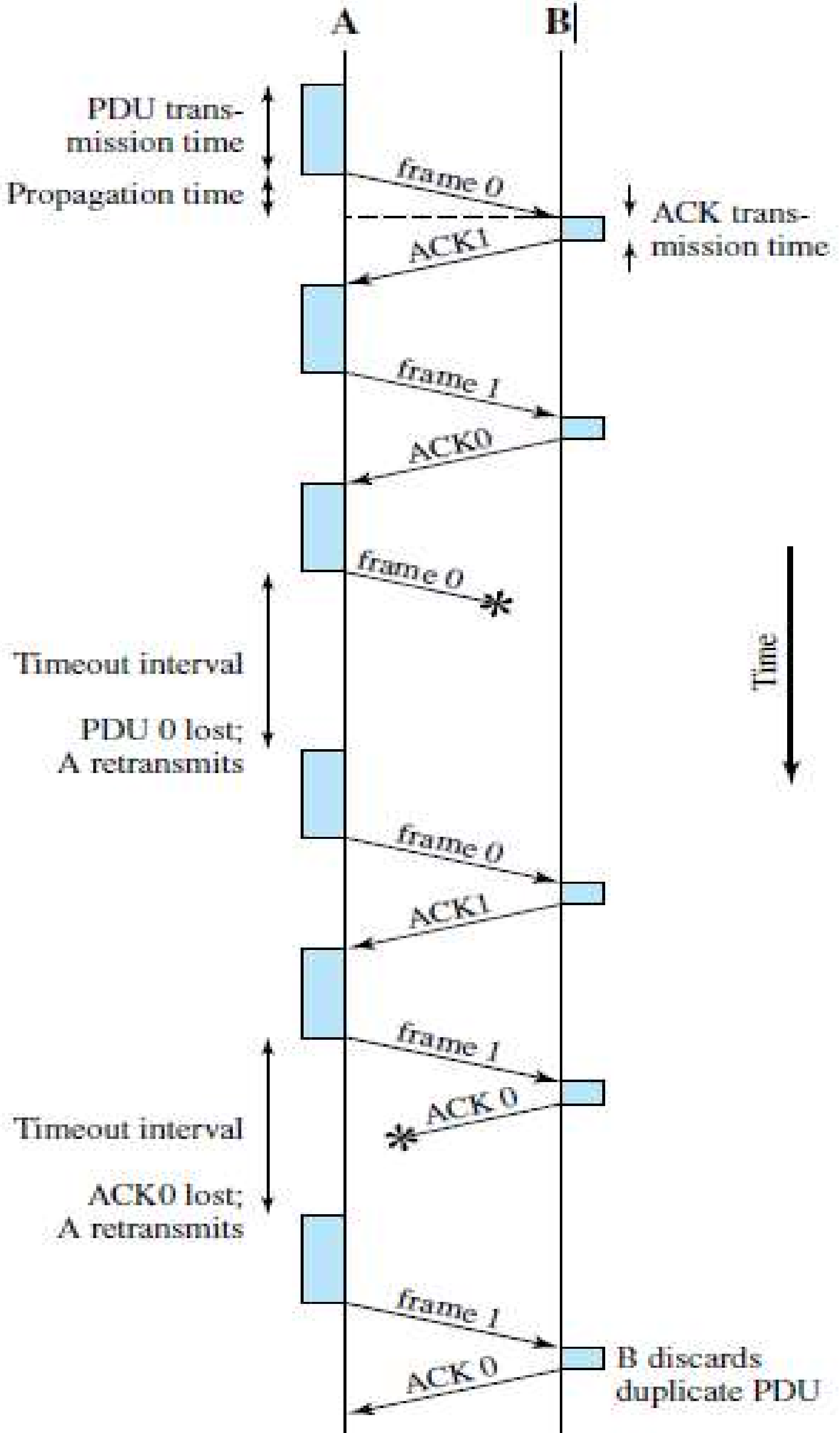
5.1 ARQ (Automatic Repeat reQuest)

- Any time an error is detected in an exchange, specified frames are retransmitted. This process is called ARQ
- ARQ - Which basically means the retransmission of data
- The ARQ techniques are:-
  1. Stop and Wait ARQ
  2. Go-Back-N ARQ
  3. Selective Repeat ARQ

Sliding Window Protocol

5.1.1 Stop and Wait ARQ

- Protocols in which the sender sends one frame and then waits for an acknowledgement before proceeding are called **stop-and-wait**
- No other data frames can be sent until the destination station's reply arrives at the source station
- Two sorts of errors could occur :
    1. Error in Data Frame
    2. Error in Acknowledgement



5.1.1 Stop and Wait ARQ

1. Error in Data Frame

- The frame that arrives at the destination could be damaged
- The receiver detects this by using the error-detection technique referred to earlier and simply discards the frame
- To account for this possibility, the source station is equipped with a timer
  - ❖After a frame is transmitted, the source station waits for an acknowledgment
  - ❖If no acknowledgment is received by the time that the timer expires, then the same frame is sent again

5.1.1 Stop and Wait ARQ

- To avoid this problem, frames are alternately labelled with 0 or 1 and positive acknowledgments are of the form ACK0 and ACK1
- ACK0 acknowledges receipt of a frame numbered 1 and indicates that the receiver is ready for a frame numbered 0
- All frame reaching for transmission is numbered 0 and 1 alternatively

2. Error in Acknowledgement
    1. Station A sends a frame.
    2. The frame is received correctly by station B, which responds with an acknowledgment (ACK).
    3. The ACK is damaged in transit and is not recognizable by A, which will therefore time out and resend the same frame
    4. This duplicate frame arrives and is accepted by B.
    5. B has therefore accepted two copies of the same frame as if they were separate.

5.1.2 Go-Back-N ARQ

- Uses Sliding window Technique
- Station may send a series of frames sequentially numbered modulo some maximum value (Maximum size of window)
- Acknowledgement is send for the group of frame instead of single frame.
- Have positive and negative acknowledgement

Sequence Numbers

- Frames from a sending station are numbered sequentially However, because we need to include the sequence number of each frame in the header, we need to set a limit
- Range of sequence number varies from 0 to 2^m-1

5.1.2 Go-Back-N ARQ

- Consider m=3 so value is 0 to 7
- However, we can repeat the sequence. So the sequence numbers are

0,1,2,3,4,5,6, 7,0,1,2,3,4,5,6,7,0,1,…

Stop and wait single frame have single ack but in

- There is 2 types of Acknowledgement frames in this scenario
  1. RR(Receive Ready)
  2. REJ(Reject)

5.1.2 Go-Back-N ARQ

1. Receive Ready(RR)

It is like positive acknowledgement.

When all frames upto i^{th} frame is received receiver will send RR i+1 and after that sender will receive RR i+1 message then the sender will send i+2 th frame

1. REJECT(REJ)

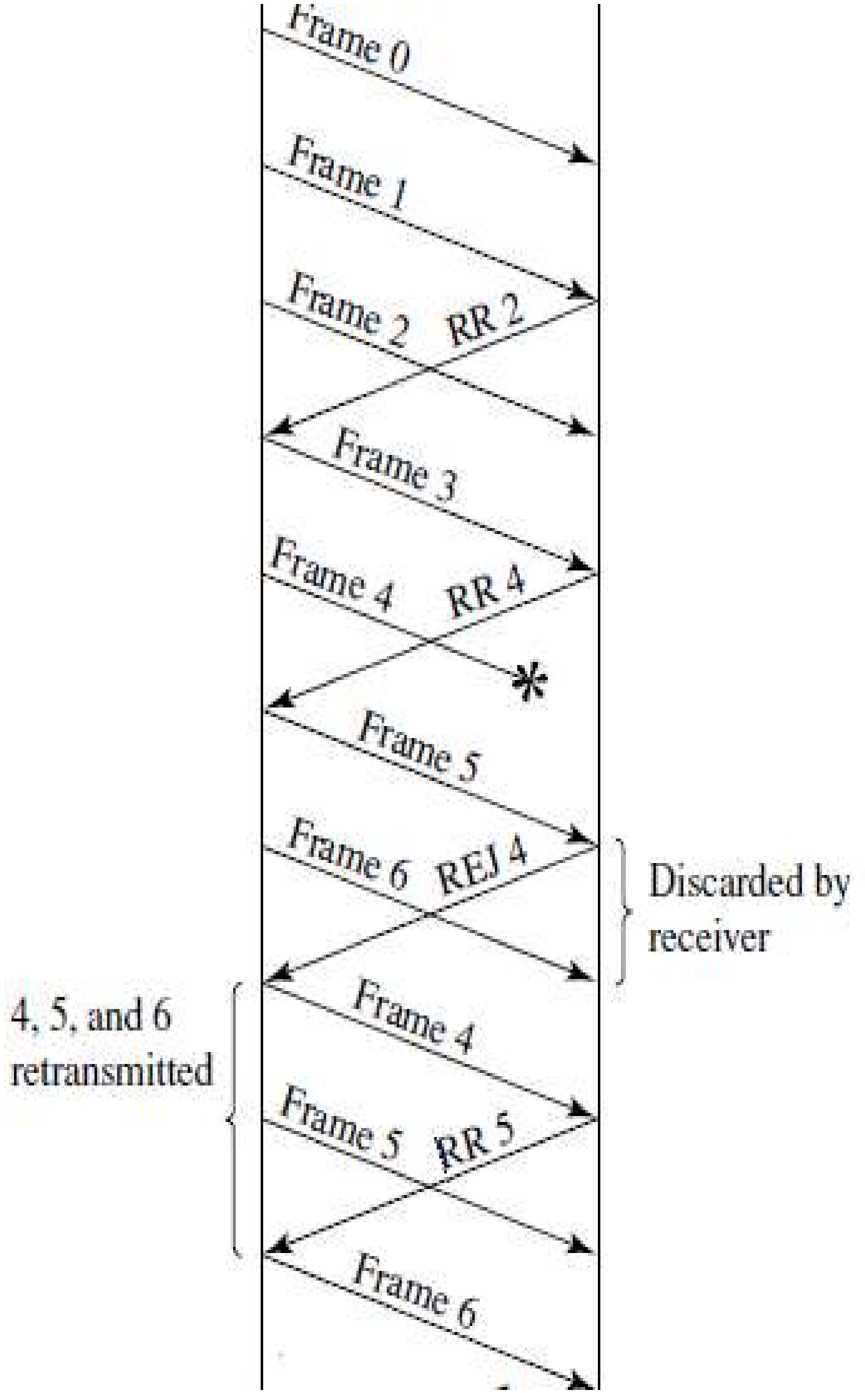
It is like negative acknowledgement.

When frame number i is not received

REJ i is send and sender will resend the frames starting from i

5.1.2 Go-Back-N ARQ

- In the figure First frame 0,1,2 is sent after Receiving Frame 1, RR2 is send by the receiver then frame 3 , 4 , 5 is sent and 4 is lost in middle of transmission and REJ 4 is sent after that frame 4,5 is sent again RR5 is sent after receiving frame 6,7 is sent

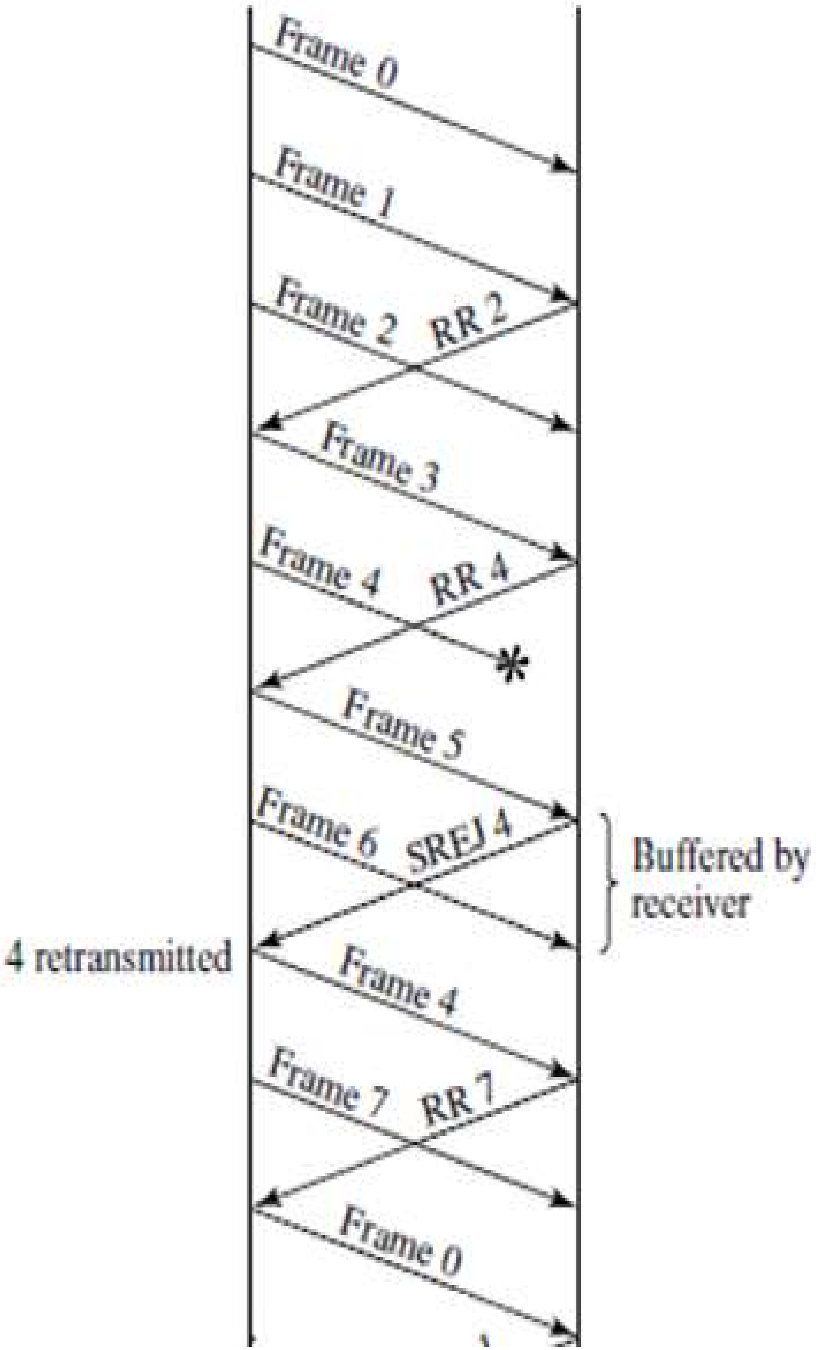


5.1.3 Selective Repeat ARQ

- RR is used similarly as in GO back N
- SREJ used as negative ACK
- With selective-reject ARQ, the only frames retransmitted are those that receive a negative acknowledgment, in this case called **SREJ**, or those that time out
- Selective reject would appear to be more efficient than go-back-N, because it minimizes the amount of retransmission
- On the other hand, the receiver must maintain a buffer large enough to save post-SREJ frames until the frame in error is retransmitted and must contain logic for reinserting that frame in the proper sequence

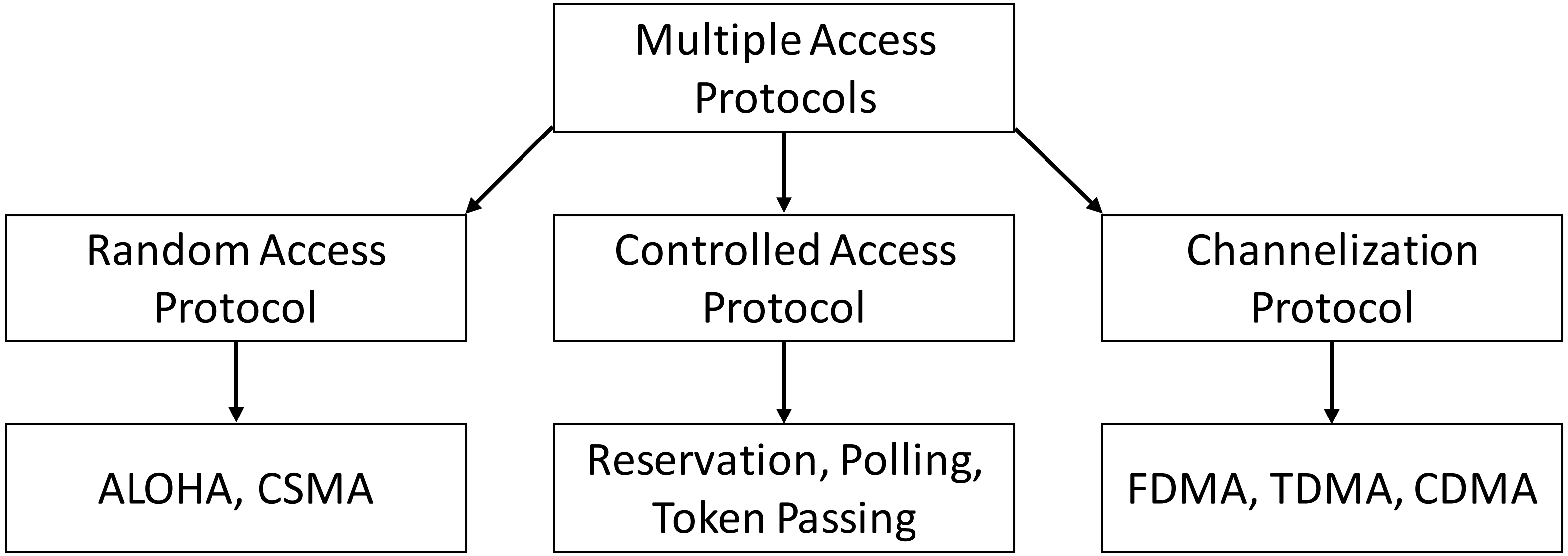
5.1.3 Selective Repeat ARQ

- The transmitter, too, requires more complex logic to be able to send a frame out of sequence
- Because of such complications, select-reject ARQ is much less widely used than go-back-N ARQ
- Selective reject is a useful choice for a satellite link because of the long propagation delay involved



6. Media Access(Multiple Access)

- In random access or contention methods, no station is superior to another station and none is assigned the control over another
- No station permits, or does not permit, another station to send(Randomly send if medium is free)


Multiple Access Protocols
Random Access Protocol
Controlled Access Protocol
Channelization Protocol
ALOHA, CSMA
Reservation, Polling, Token Passing
FDMA, TDMA, CDMA

6. Media Access(Multiple Access)

- Two features give random access method

1. There is no scheduled time for a station to transmit. Transmission is random among the stations
2. No rules specify which station should send next. Stations compete with one another to access the medium

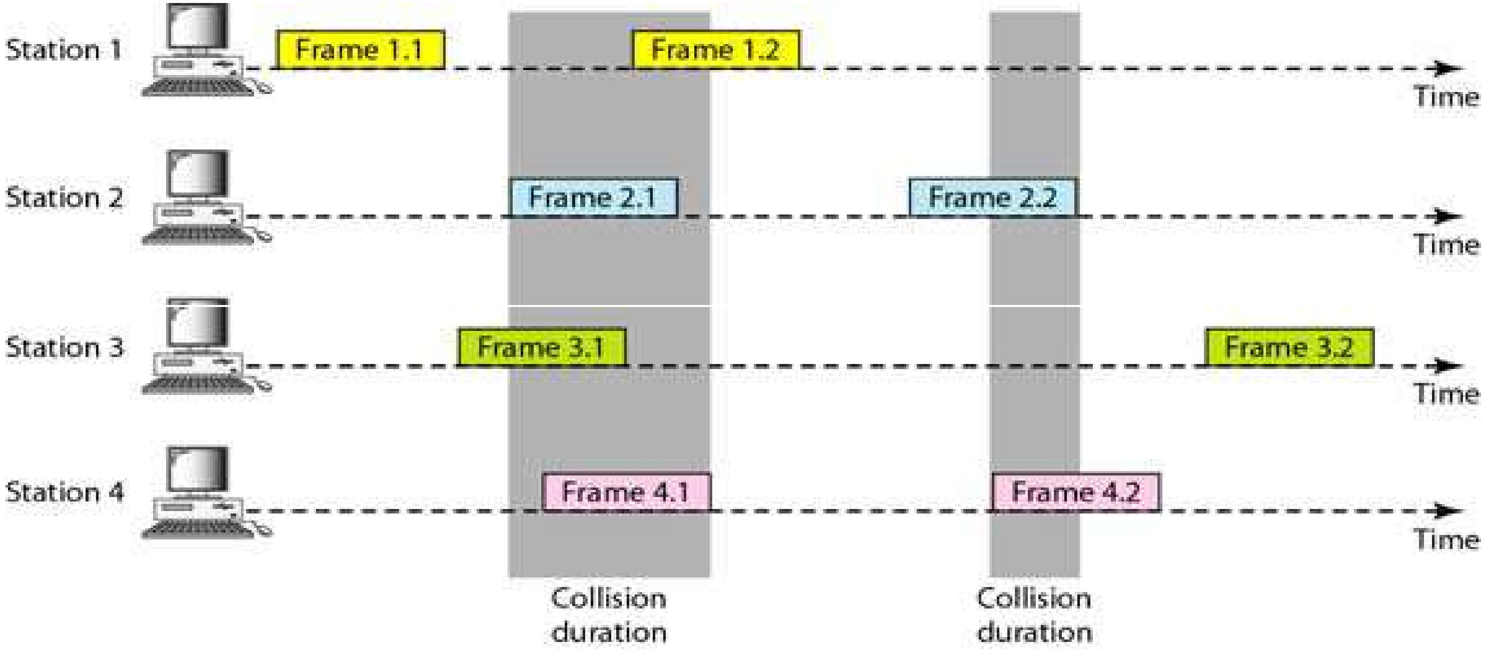
- There are 2 methods
    1. ALOHA
    2. CSMA

6.1 ALOHA

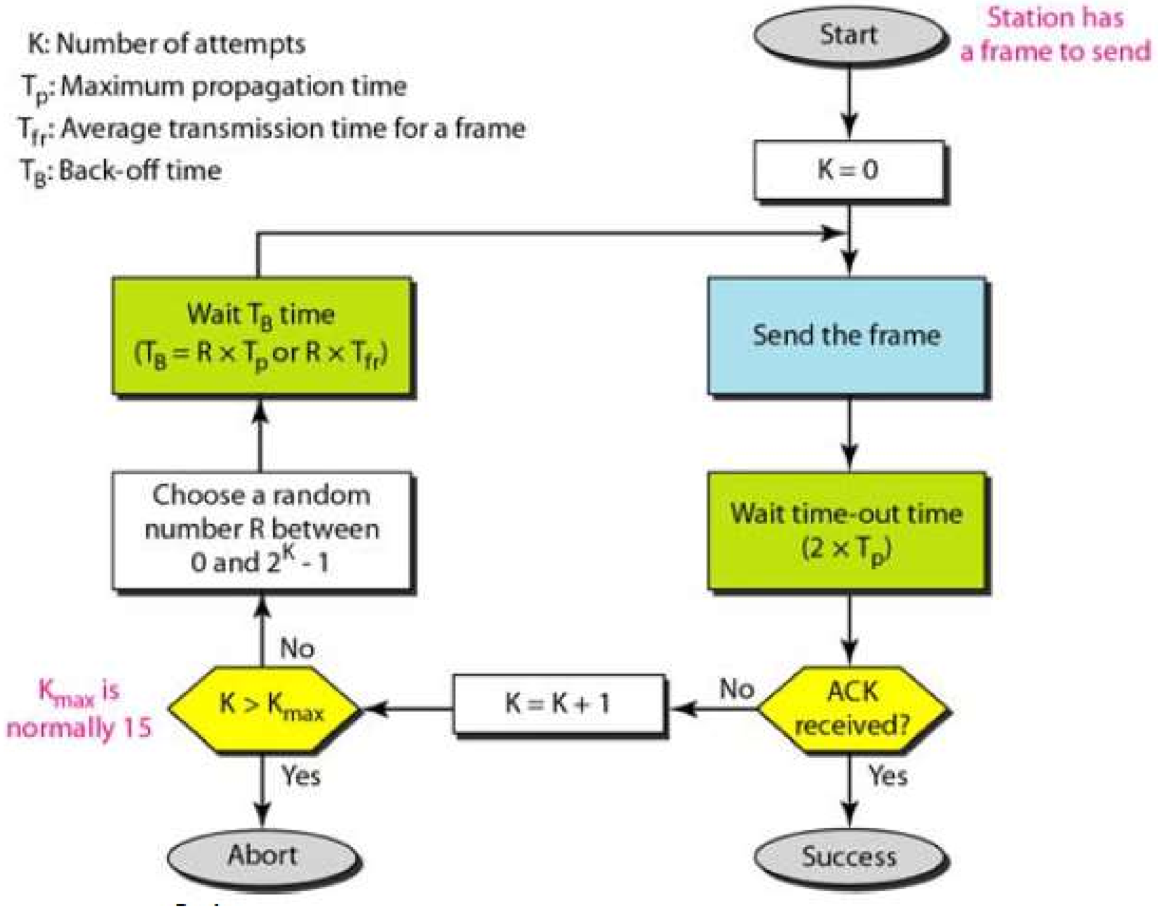
- Developed at the Univ. of Hawaii
- Random access method used for any type of shared medium(wireless and wired)
- ALOHA have 2 types
  1. Pure ALOHA
  2. Slotted ALOHA

6.1.1. pure ALOHA

- The node immediately transmits its frame completely If the frame is collided it retransmits the frame again (after completely transmitting its collided frame) with the probability



6.1.1. pure ALOHA *(Frame sending Procedure)*

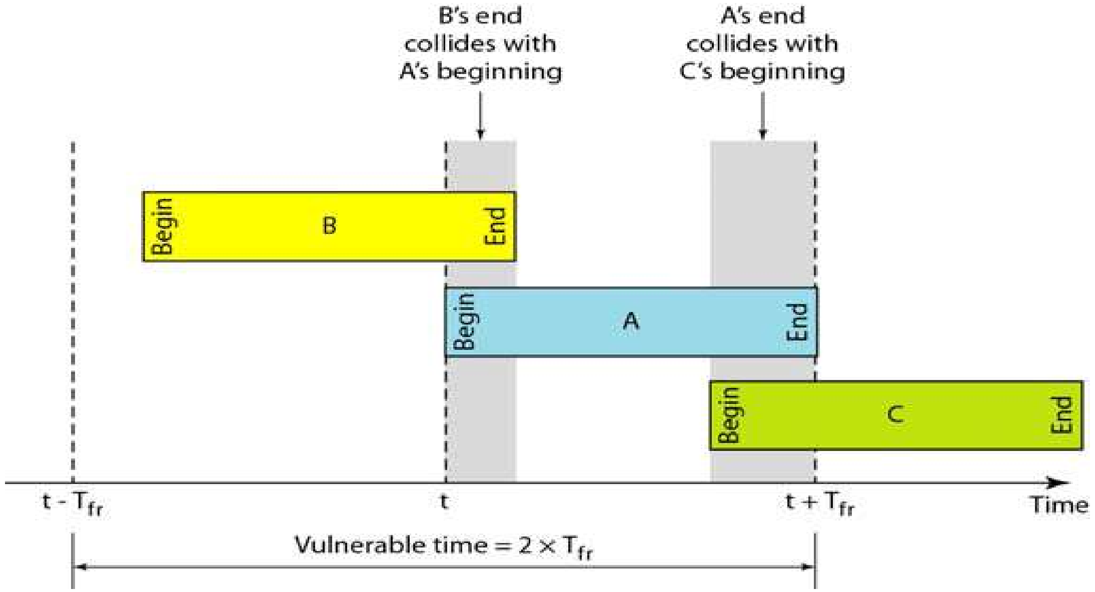

K: Number of attempts
T_p: Maximum propagation time
T_{fr}: Average transmission time for a frame
T_B: Back-off time
Start
Station has a frame to send
K = 0
Send the frame
Wait time-out time ($2 \times T_p$)
ACK received?
No
Yes
Success
K = K + 1
$K > K_{max}$
K_{max} is normally 15
Yes
Abort
No
Choose a random number R between 0 and $2^K - 1$
Wait T_B time ($T_B = R \times T_p$ or $R \times T_{fr}$)

6.1.1. pure ALOHA *(Frame sending Procedure)*

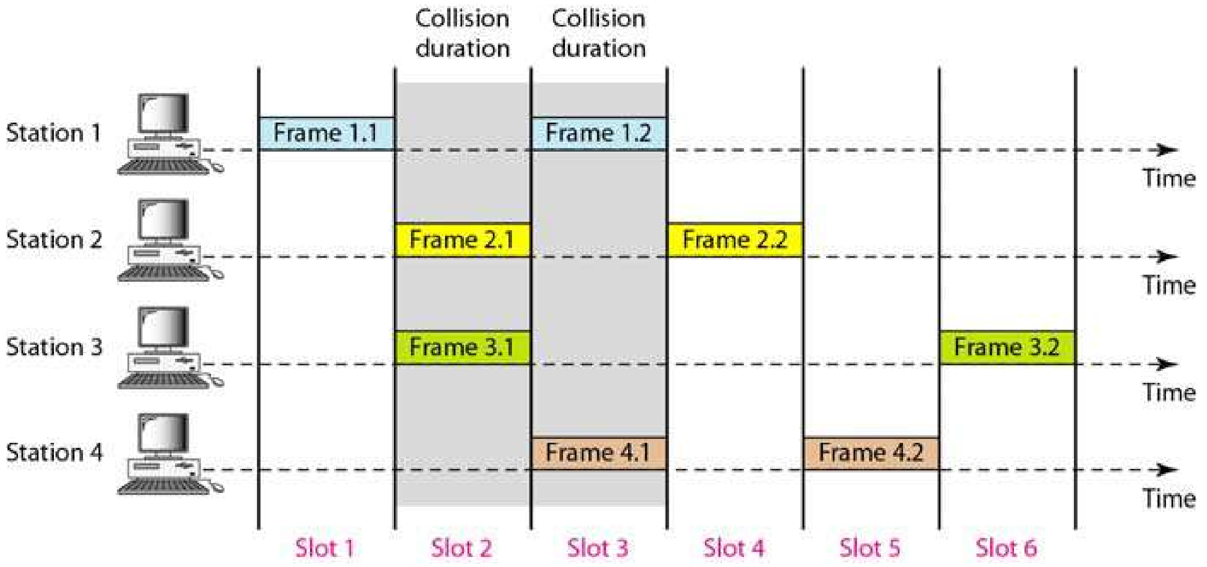
- Procedure in pure ALOHA:
  1. Send frame
  2. Wait for a time out(time interval) and check if acknowledgement is received or not if received transmission is **SUCCESS** else go to step 3
  3. Increment the number of attempts (k)and check if it reaches maximum(15) if it reaches maximum attempt **ABORT** transmission else go to step 4
  4. Choose a random time interval (R) between 0 to $2^k - 1$ and calculate back off time(T_B) using random time interval time and propagation time/frame transmission time(T_p/T_{fr}) → T_B=R*T_p or T_B=R*T_{fr}
  5. Repeat from step 1 until success/ abort

6.1.1. pure ALOHA *(Vulnerable Time)*

- Station A sends a frame at time *t*
- Now imagine station B has already sent a frame between *t* - *T*fr and *t*
- This leads to a collision between the frames from station A and station B
- The end of B's frame collides with the beginning of A's frame
- On the other hand, suppose that station C sends a frame between *t* and *t* + *Tfr*
- Here, there is a collision between frames from station A and station C. The beginning of C's frame collides with the end of A's frame
- Vulnerable time = 2*T_{fr}


B's end collides with A's beginning
A's end collides with C's beginning
Begin
B
End
Begin
A
End
Begin
C
End
t - T_{fr}
t
t + T_{fr}
Time
Vulnerable time = 2 × T_{fr}

6.1.2. slotted ALOHA



- Frames are of the same size time is divided into equal size slots, time to transmit 1 frame nodes start to transmit frames only at beginning of slots nodes are synchronized
- If a frame is ready for transmission after starting time of slot 1 it will be transmitted in slot 2
- If 2 or more nodes transmit in slot, all nodes detect collision

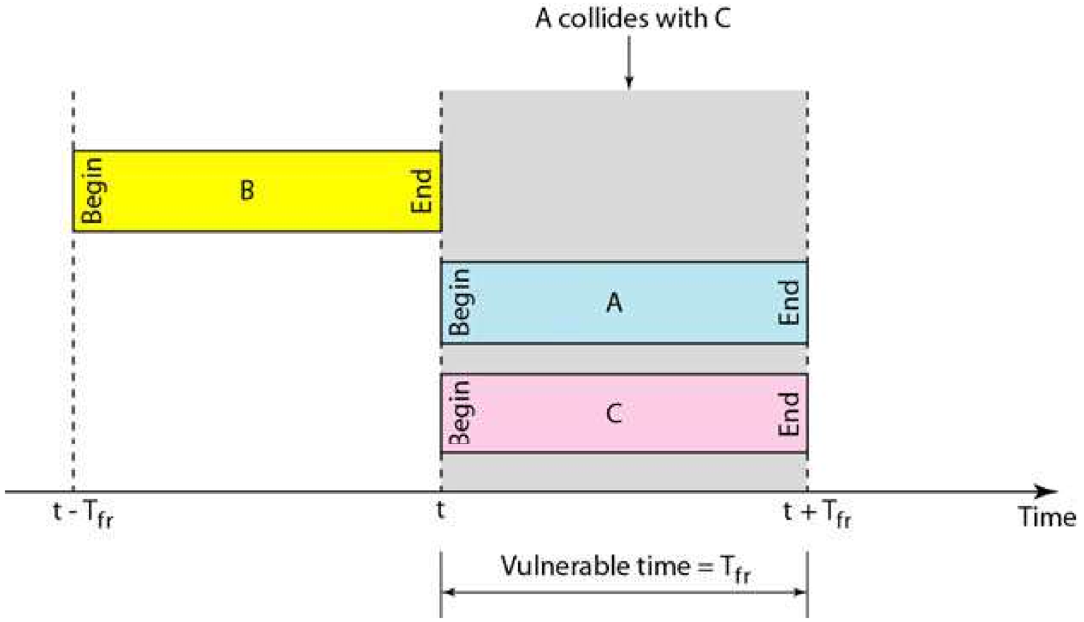
6.1.2. slotted ALOHA

- when node obtains fresh frame it transmits in next slot
- No collision, node can send new frame in next slot
- If collision, node retransmits frame in each subsequent slot with probability (p) until success
- Here Procedure of frame sending is similar to pure ALOHA but of frame will be sent only on starting of time slot

6.1.2. slotted ALOHA *(Vulnerable Time)*

- Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot
- This means that the station which started at the beginning of this slot has already finished sending its frame

- Vulnerable time = T_{fr}



6.2. CSMA (Carrier Sense Multiple Access)

- Invented to minimize collisions and increase the performance
- A station now “follows” the activity of other stations
- Simple rules for a polite human conversation
  1. Listen before talking
  2. If someone else begins talking at the same time as you, stop talking
- A node should not send if another node is already sending(Carrier Sensing)
- Vulnerable time is the propagation time which is the time needed for a signal to propagate from one end of the medium to the other

6.2.1 CSMA (Persistence Methods)

- Persistence methods :- Methods for Sensing the channel (busy/ idle)
- 3 Persistence methods are available:
  1. l-persistence
  2. Non-persistence
  3. P-persistence

6.2.1.1. I-Persistence Method

- **In** this method, after the station finds the line idle, it sends its frame immediately (with probability I)
- This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately

Figure : Behaviour I persistence

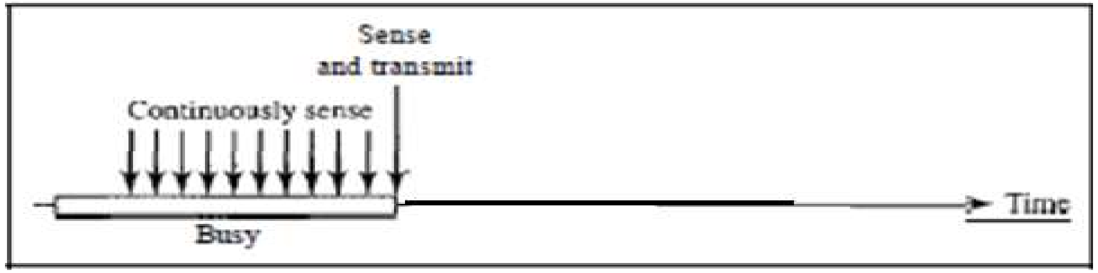
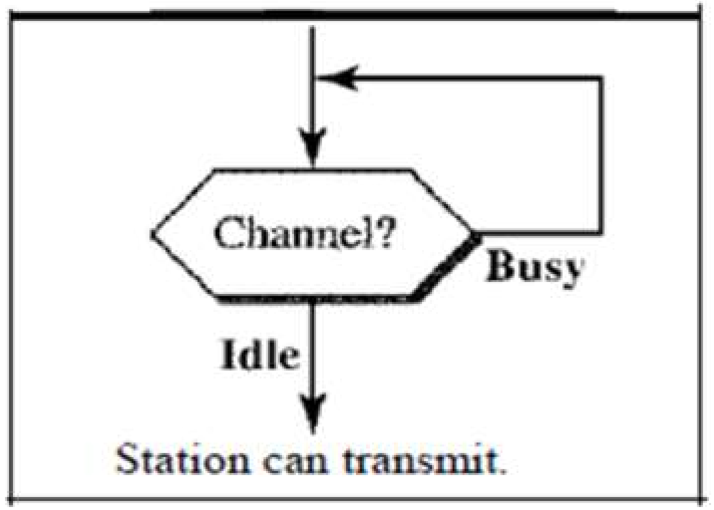


Figure : Flow diagram of I persistence



6.2.1.2. Non-Persistence Method

- In the Non-persistent method, a station that has a frame to send senses the line.
- If the line is idle, it sends immediately.
- If the line is not idle, it waits a random amount of time and then senses the line again.
- The Non-persistent approach reduces the chance of collision because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously
- This method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.

Figure : Behaviour of Non persistence

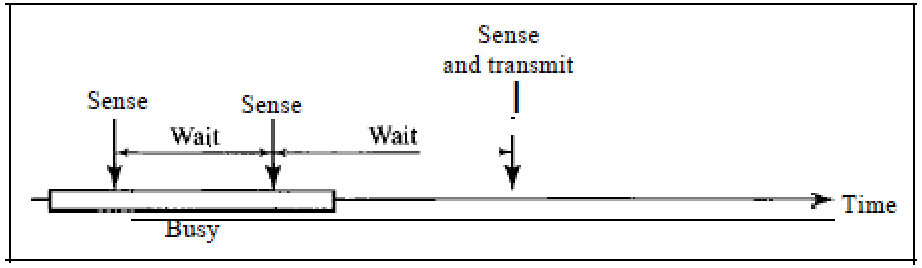
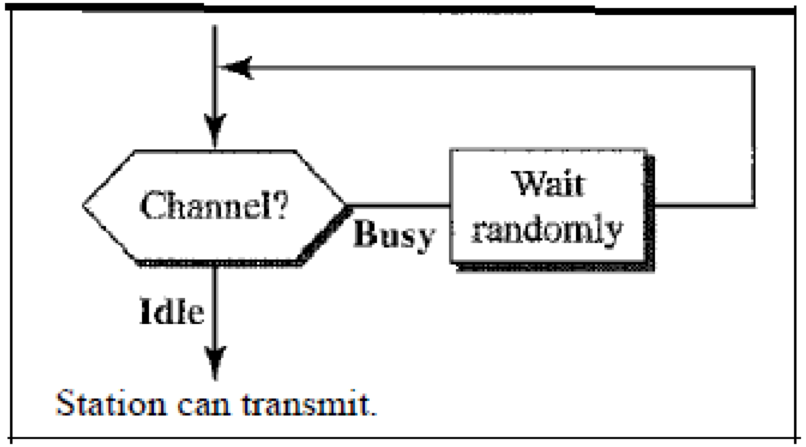


Figure : Flow diagram of Non persistence



6.2.1.3. P-Persistence Method

- **The p-persistent method** is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time
- The p-persistent approach combines the advantages of the other two strategies
- It reduces the chance of collision and improves efficiency.

Figure : Behaviour P persistence

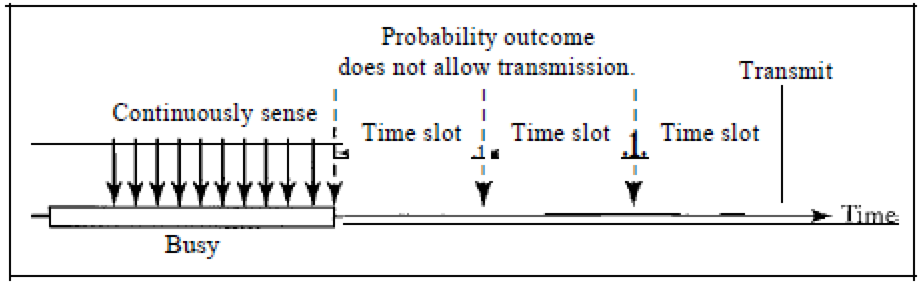
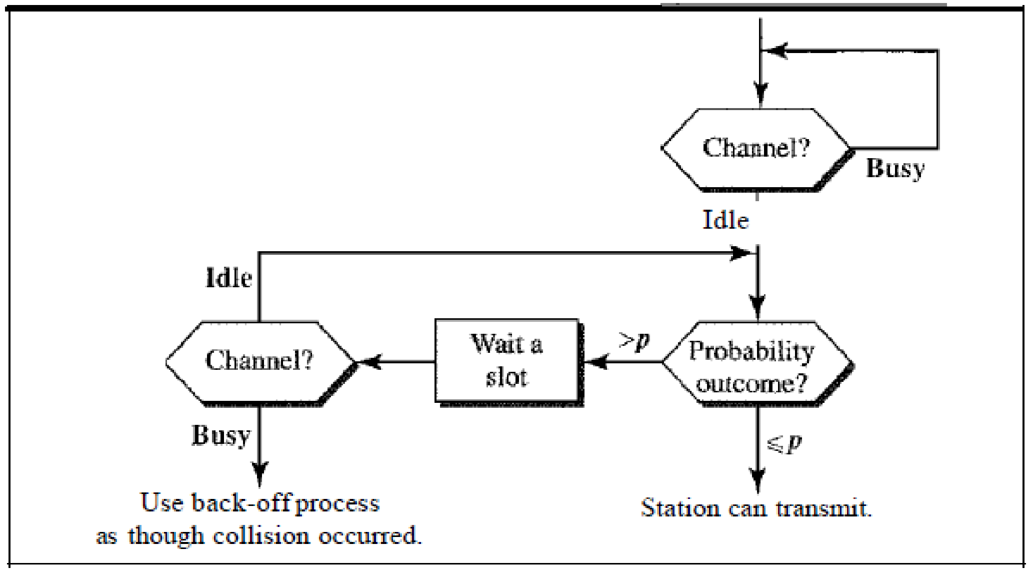


Figure : Flow diagram of P persistence

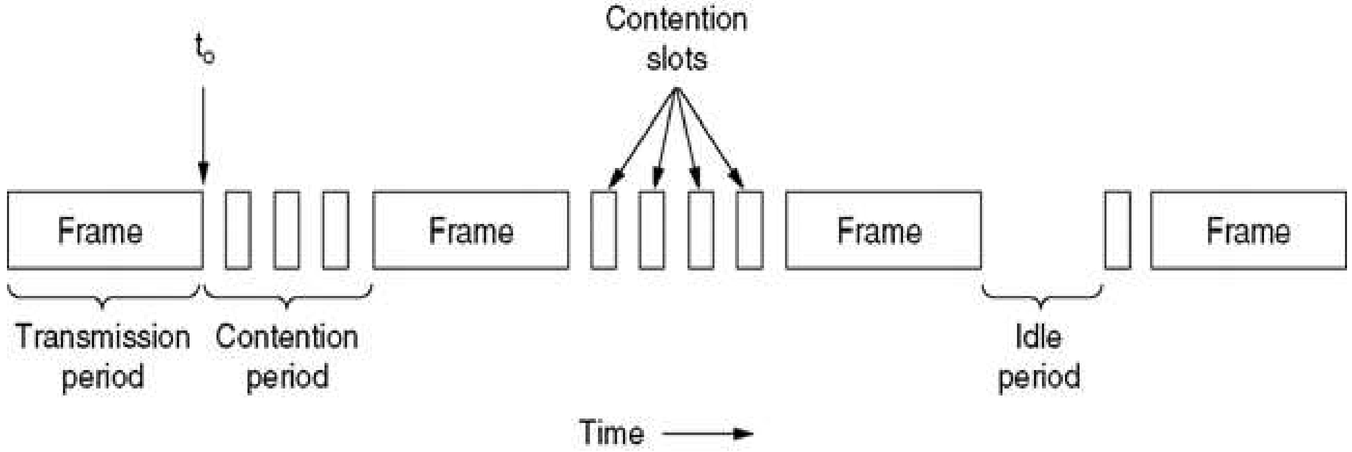


6.2.1.3. P-Persistence Method

- In this method, after the station finds the line idle it follows these steps:
  1. With probability p, the station sends its frame.
  2. With probability q = 1 - p, the station waits for the beginning of the next time slot and checks the line again.
     I. If the line is idle, it goes to step 1.
     II. If the line is busy, it acts as though a collision has occurred and uses the back off procedure.

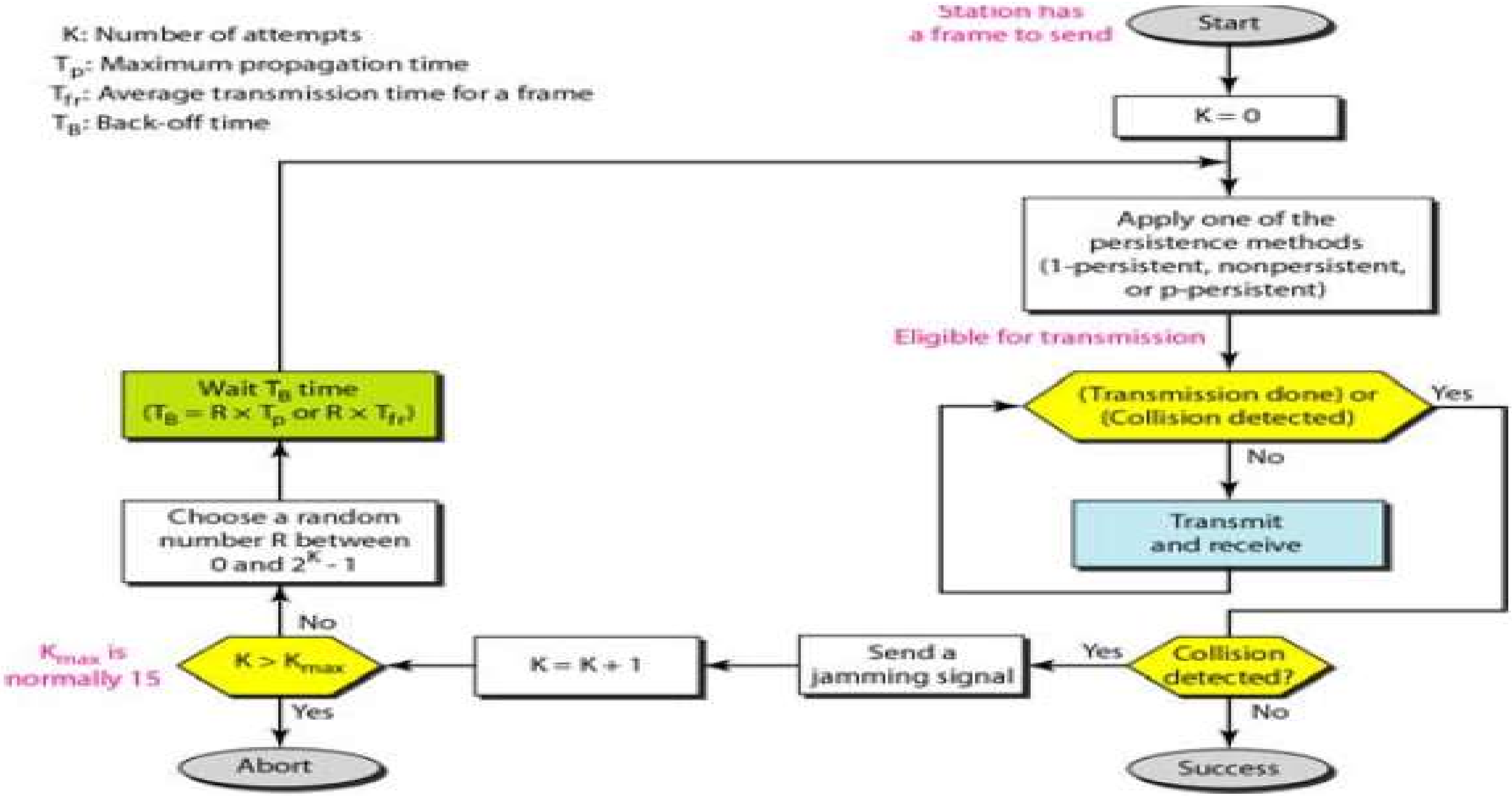
6.2.2 CSMA/CD (Collision Detection)

- In this method, a station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station is finished. If, however, there is a collision, the frame is sent again



- In CSMA/CD Channel can be in one of the three states: contention, transmission, and idle.

6.2.2 CSMA/CD Procedure



K: Number of attempts

T_p: Maximum propagation time

T_{fr}: Average transmission time for a frame

T_B: Back-off time

Station has a frame to send

Start

K = 0

Apply one of the persistence methods (1-persistent, nonpersistent, or p-persistent)

Eligible for transmission

(Transmission done) or (Collision detected)

Yes

No

Transmit and receive

Collision detected?

Yes

No

Success

Send a jamming signal

K = K + 1

$K > K_{max}$

K_{max} is normally 15

Yes

Abort

No

Choose a random number R between 0 and $2^K - 1$

Wait T_B time ($T_B = R \times T_p$ or $R \times T_{fr}$)

6.2.2 CSMA/CD Procedure

- Procedure is similar to ALOHA but with certain differences
- The main differences are:-
  - Addition of the persistence process before transmission
  - Transmission is continuous process
  - Jamming signal is used in it

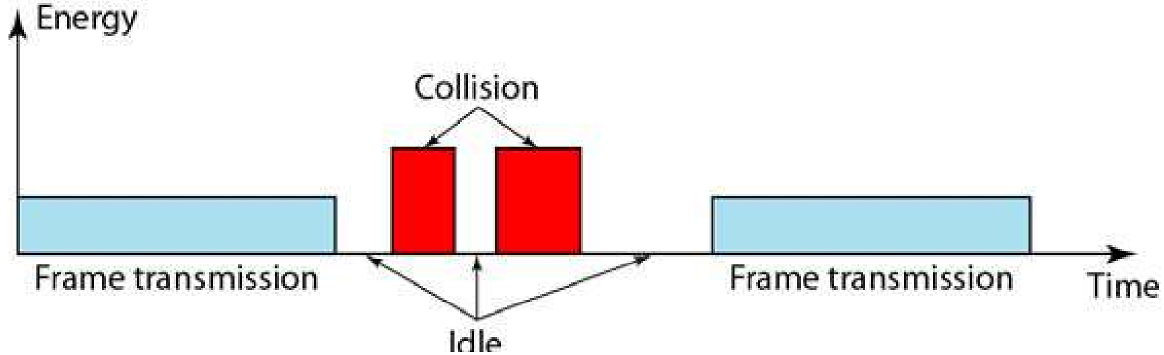


Figure : Energy Level During Transmission

6.2.2 CSMA/CD *Throughput*

- The throughput of *CSMAlCD* is greater than that of pure or slotted ALOHA
- The maximum throughput occurs at a different value of G and is based on the persistence method and the value of *p* in the p-persistent approach.
- For I-persistent method the maximum throughput is around 50 percent when G =1
- For non-persistent method, the maximum throughput can go up to 90 percent when G is between 3 and 8

6.2.3 CSMA/CA (Collision Avoidance)

- Collisions are avoided through the use of CSMAICA's three strategies:
  1. Inter Frame Space
  2. Contention window
  3. Acknowledgments

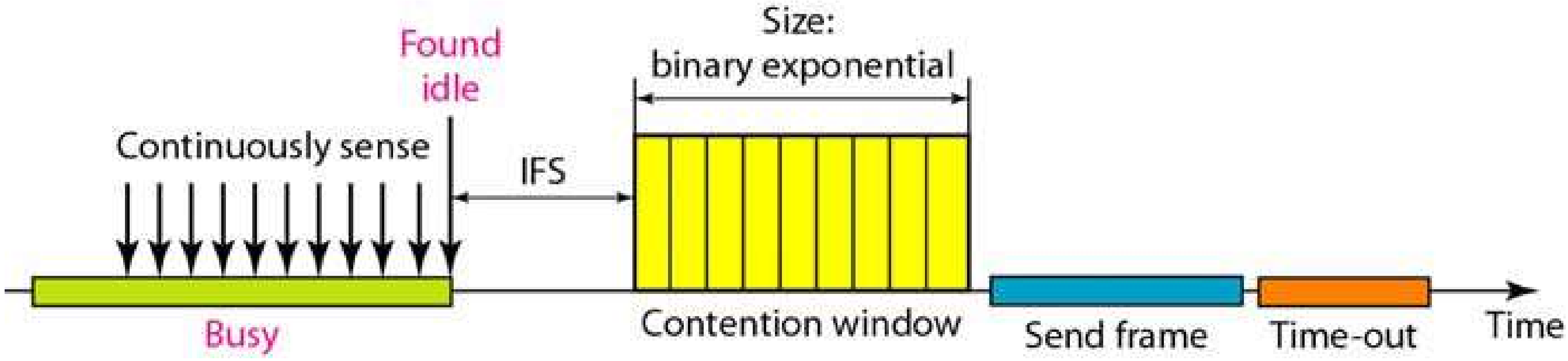


Figure : Timing in CSMA/CA

6.2.3 CSMA/CA Inter Frame Space

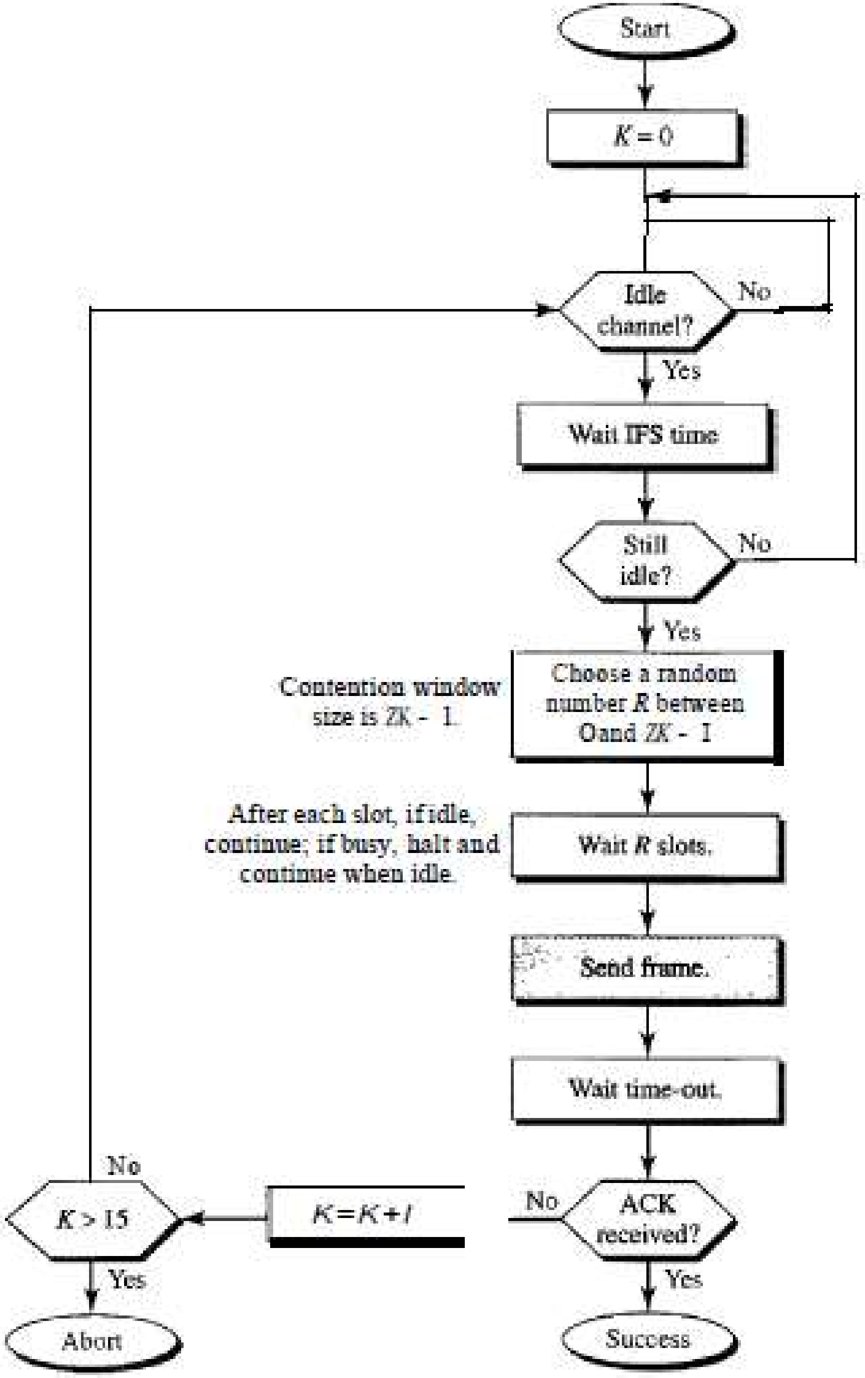
- When an idle channel is found, the station does not send immediately
- Station waits for a period of time called the inter frame space or IFS
- In CSMA/CA, the IFS can also be used to define the priority of a station or a frame
  - For EX: a station that is assigned a shorter IFS has a higher priority while sending

6.2.3 CSMA/CA Contention Window

- Contention window (random wait time) is an amount of time divided into slots
- A station that is ready to send chooses a random number of slots as its wait time
- The number of slots in the window changes according to the binary exponential back-off strategy
- Binary exponential back-off strategy means that it is set to one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time
- Restarts content window timer when the channel becomes idle

6.2.3 CSMA/CA Procedure

- Channel needs to be sensed before and after the IFS and sensed during the contention time
- For each time slot of the contention window, the channel is sensed
- If it is found idle, the timer continues else if the channel is found busy, the timer is stopped and continues after the timer becomes idle again.


Start
K = 0
Idle channel?
No
Yes
Wait IFS time
Still idle?
No
Yes
Contention window size is 2K - 1.
Choose a random number R between 0and 2K - 1
After each slot, if idle, continue; if busy, halt and continue when idle.
Wait R slots.
Send frame.
Wait time-out.
ACK received?
No
Yes
Success
K=K+l
K > 15
No
Yes
Abort

7. IEEE Data Link Layer Protocol/Standards

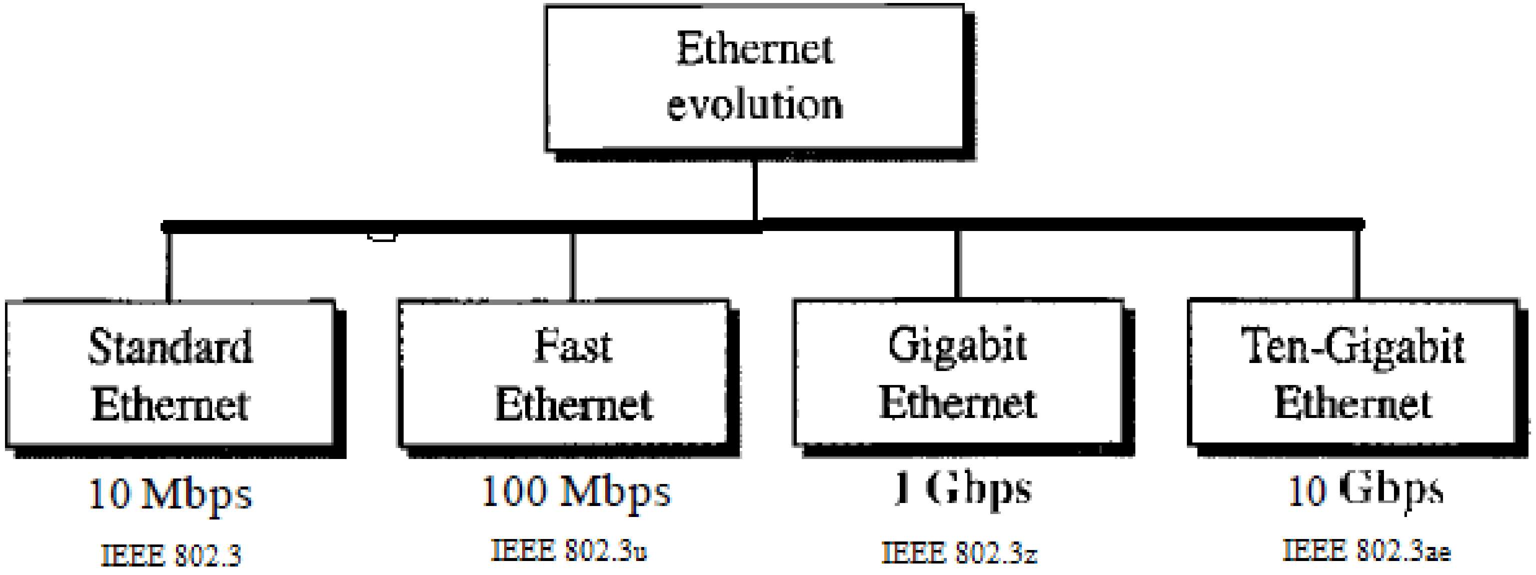
- Here we are discussing about the protocols and standard used in data link layer.
- Here 802 commonly refer to data link layer specifically(MAC Sub layer)

Standards

- **IEEE 802.3**
- **IEEE 802.4**
- **IEEE 802.5**

7.1. IEEE 802.3 Ethernet

- IEEE 802.3 frame format (refer section 3.3)
- It uses mainly **CSMA** as channel access mechanism



Ethernet evolution

Standard Ethernet
10 Mbps
IEEE 802.3

Fast Ethernet
100 Mbps
IEEE 802.3u

Gigabit Ethernet
1 Gbps
IEEE 802.3z

Ten-Gigabit Ethernet
10 Gbps
IEEE 802.3ae

Figure : Different IEEE 802.3 standards

7.1.1. IEEE 802.3 Standard Ethernet

- Maximum data rate is up to **10Mbps**
- Standard Ethernet uses **I-persistent CSMA/CD**
- Uses 48 bit addressing
- It uses following Technologies

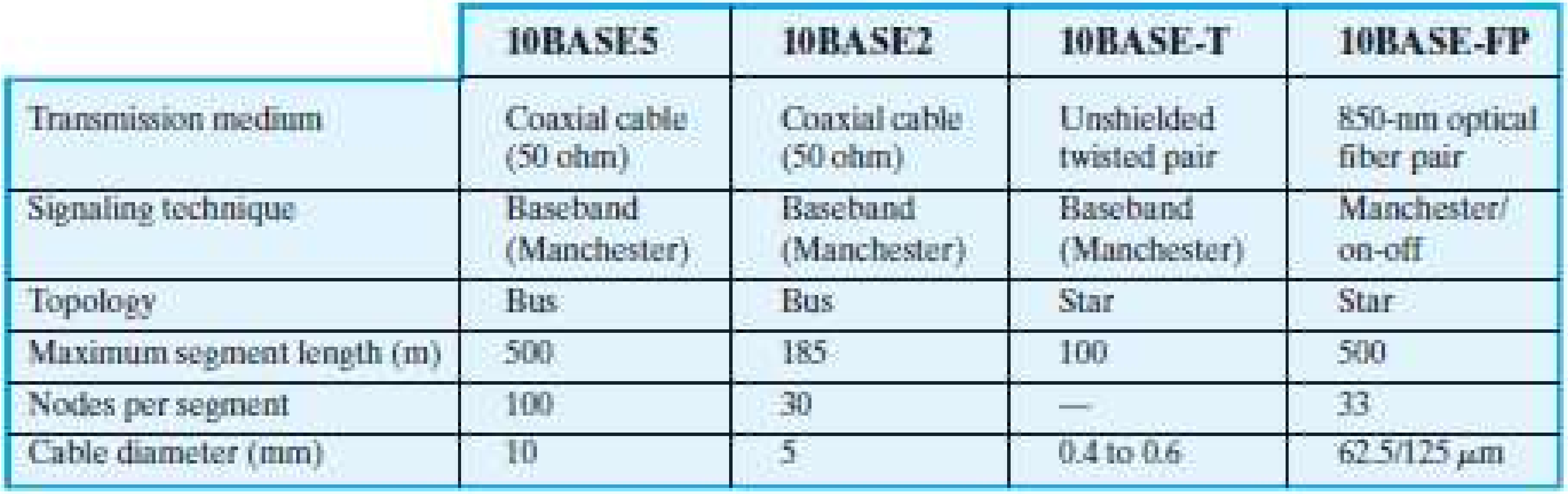
10Base2 - Thin Co-axial cable (Also called Thinnet / Thin Ethernet)

10Base5 - Thick Co-axial cable (Also called Thicknet / Thick Ethernet)

10BaseT – Unsheilded Twisted Pair Cable (Also called Twisted pair Ethernet)

10BaseF – Optical Fiber Cable

7.1.1. IEEE 802.3 Standard Ethernet



	10BASE5	10BASE2	10BASE-T	10BASE-FP
Transmission medium	Coaxial cable (50 ohm)	Coaxial cable (50 ohm)	Unshielded twisted pair	850-nm optical fiber pair
Signaling technique	Baseband (Manchester)	Baseband (Manchester)	Baseband (Manchester)	Manchester/ on-off
Topology	Bus	Bus	Star	Star
Maximum segment length (m)	500	185	100	500
Nodes per segment	100	30	—	33
Cable diameter (mm)	10	5	0.4 to 0.6	62.5/125 μm

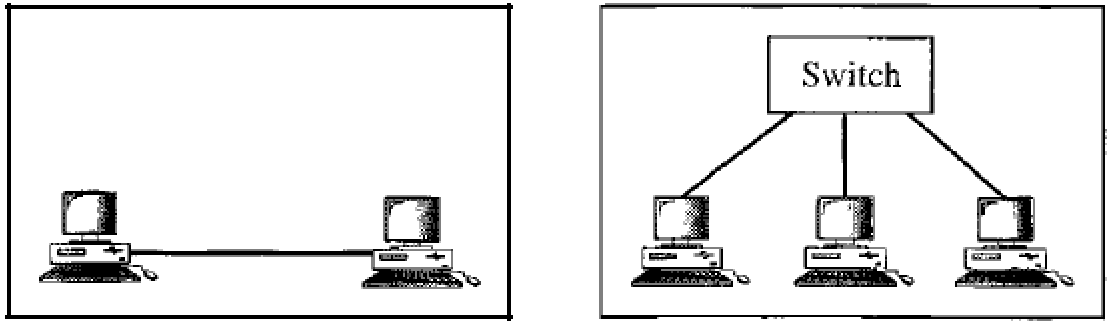
7.1.2. IEEE 802.3u Fast Ethernet

- Fast Ethernet is backward-compatible with Standard Ethernet, but it can transmit data 10 times faster at a rate of 100 Mbps

Features

1. Upgrade the data rate to 100 Mbps
2. Make it compatible with Standard Ethernet
3. Keep the same 48-bit address
4. Keep the same frame format
5. Keep the same minimum and maximum frame lengths
6. Can connect Point to point / Star

7.1.2. IEEE 802.3u Fast Ethernet



a. Point-to-point

b. Star

Figure : Fast Ethernet Connection Topologies

- 100BaseTX UTP Cat5 Two wire Implementation
- 100BaseFX Fiber Optic Cable
- 100BaseT4 UTP Cat3 Four Wire Implementation

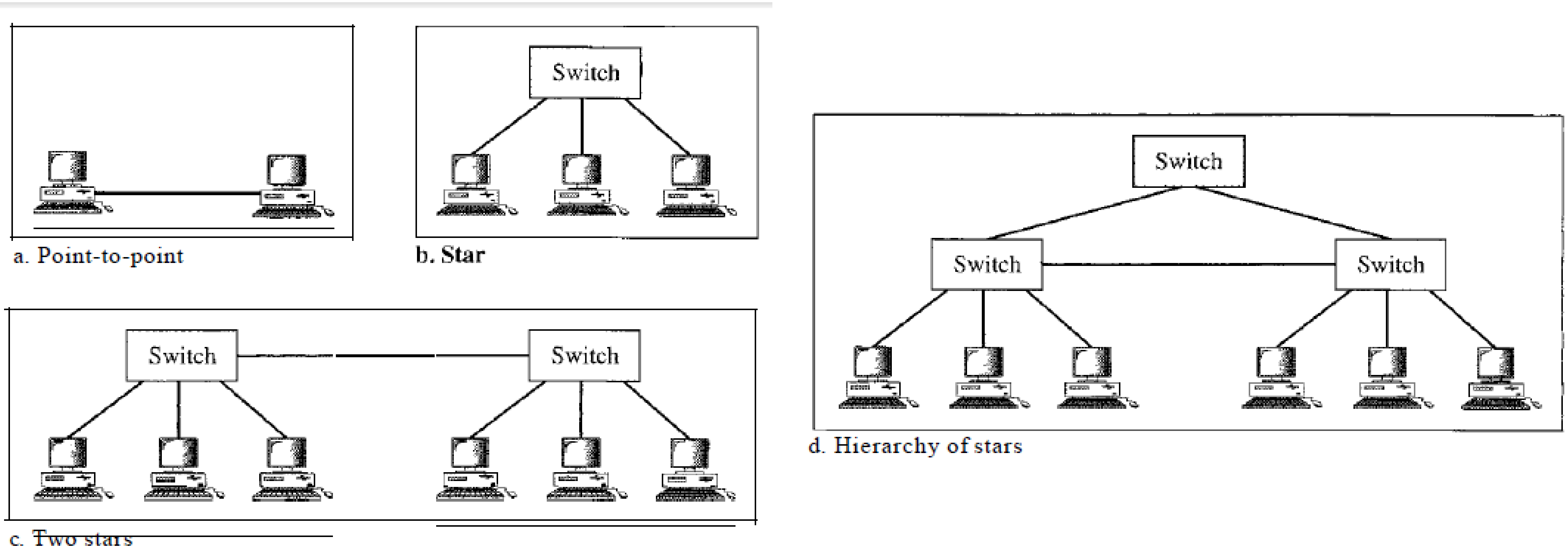
7.1.3. IEEE 802.3z GigaBit Ethernet

- Higher data rate than fast ethernet (1000 Mbps)

Features

1. Upgrade the data rate to 1 Gbps.
2. Make it compatible with Standard or Fast Ethernet.
3. Use the same 48-bit address.
4. Use the same frame format.
5. Keep the same minimum and maximum frame lengths.
6. To support auto negotiation as defined in Fast Ethernet

7.1.3. IEEE 802.3z Gigabit Ethernet



a. Point-to-point

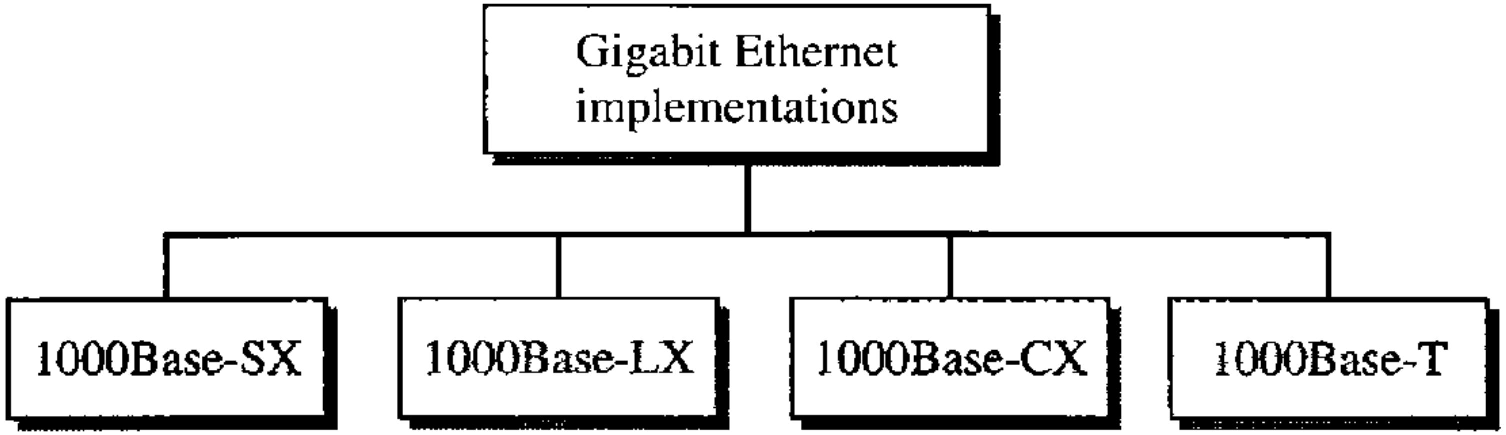
b. Star

c. Two stars

d. Hierarchy of stars

Figure : Topologies of Gigabit Ethernet

7.1.3. IEEE 802.3z Gigabit Ethernet



- 1000BaseSX – Fiber Optic Short wave (2 wire)
- 1000BaseLX – Fiber Optic Short wave (2 wire)
- 1000BaseCX – Copper STP Cable (2 wire)
- 1000BaseSX – Fiber Optic Short wave (4 wire)

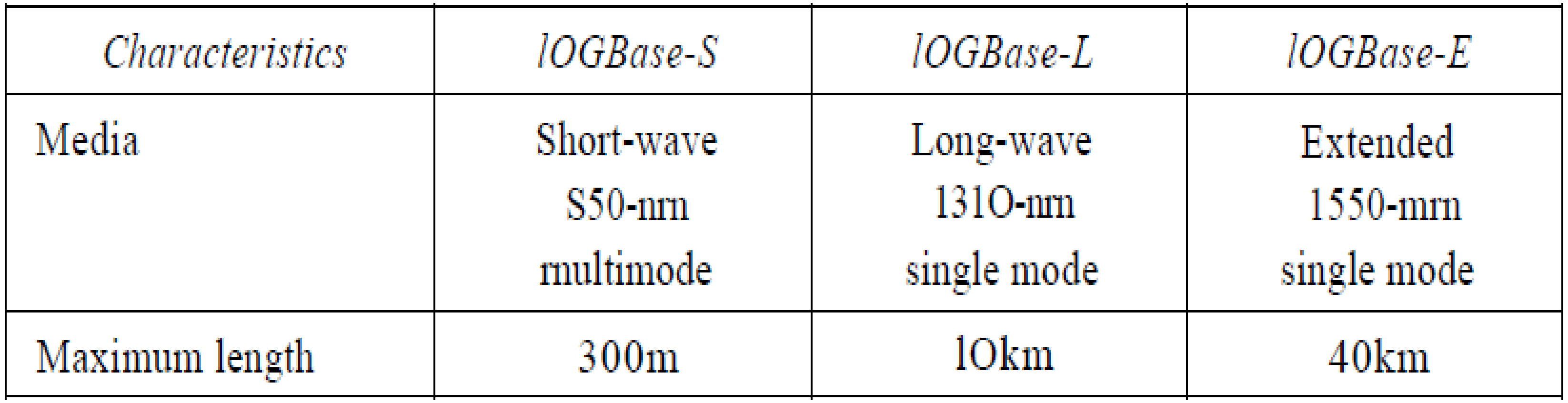
7.1.4. IEEE 802.3ae 10 GigaBit Ethernet

Features

1. Upgrade the data rate to 10 Gbps.
2. Make it compatible with Standard, Fast, and Gigabit Ethernet.
3. Use the same 48-bit address.
4. Use the same frame format.
5. Keep the same minimum and maximum frame lengths.
6. Allow the interconnection of existing LANs into a metropolitan area network (MAN) or a wide area network (WAN).
7. Make Ethernet compatible with technologies such as Frame Relay and ATM

7.1.4. IEEE 802.3ae 10 GigaBit Ethernet

- Ten-Gigabit Ethernet is designed for using fiber-optic cable over long distances



Characteristics	*lOGBase-S*	*lOGBase-L*	*lOGBase-E*
Media	Short-wave S50-nrn rnultimode	Long-wave 131O-nrn single mode	Extended 1550-mrn single mode
Maximum length	300m	lOkm	40km

Figure : 10 Gigabit Ethernet Implementation

7.2. IEEE802.4 Token Bus

- The 802.4 IEEE standard defines the Token Bus protocol for a token-passing access method on a bus topology.
- In a token-passing access method, a special packet called a token is passed from station to station and only the token holder is permitted to transmit packets onto the LAN.
- No collisions can occur with this protocol(Only One Station can transfer)
- When a station is done transmitting its packets, it passes the token to the "next" station.
- The next station does not need to be physically closest to this one on the bus, just the next logical station.

7.2. 802.4 Token Bus

- A station can hold the token for only a certain amount of time before it must pass it on -even if it has not completed transmitting all of its data.
- This assures access to all stations on the bus within a specified period of time.

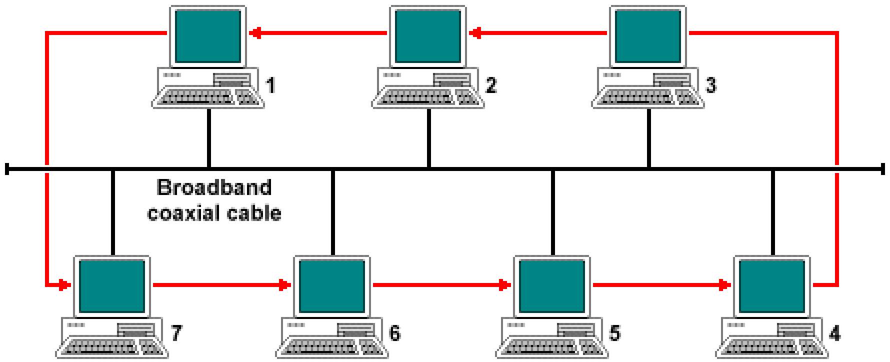


Figure : Token Bus Network (Red Arrow Indicates Token Passing Sequence)

7.3. 802.5 Token Ring

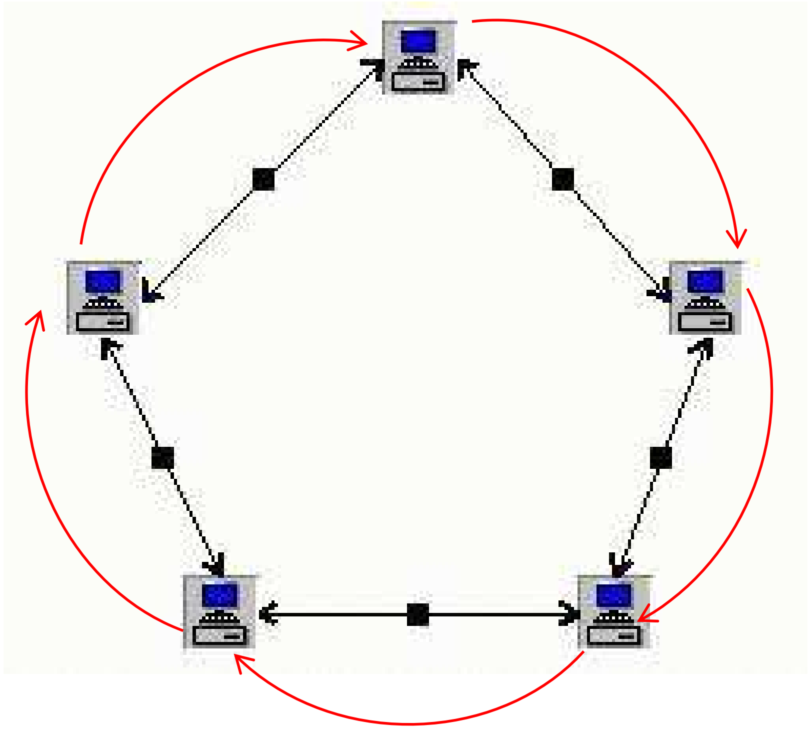


Figure : Token Bus Network (Red Arrow Indicates Token Passing Sequence)

7.3. 802.5 Token Ring

- The 802.5 IEEE standard defines the Token Ring protocol which, like Token Bus, is another token-passing access method, but for a ring topology
- A ring topology consists of a series of individual point-to-point links that form a circle
- A token is passed from station to station in one direction around the ring, and only the station holding the token can transmit packets onto the ring

7.3. 802.5 Token Ring

- Data packets travel in only one direction around the ring
- When a station receives a packet addressed to it, it copies the packet and puts it back on the ring
- When the originating station receives the packet, it removes the packet.