CHAPTER -3

Data Link Layer

By: Asst. Prof. Sanjivan Satyal

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

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

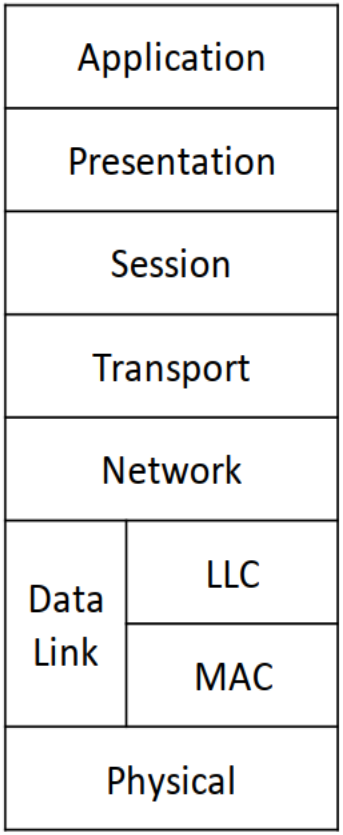
Access Control: When single comm. Channel is shared by multiple devices, MAC layer of DLL provides help to determine which device has control over the channel

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

Data Link SubLayers

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



SUB-LINK 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 sub-layer

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

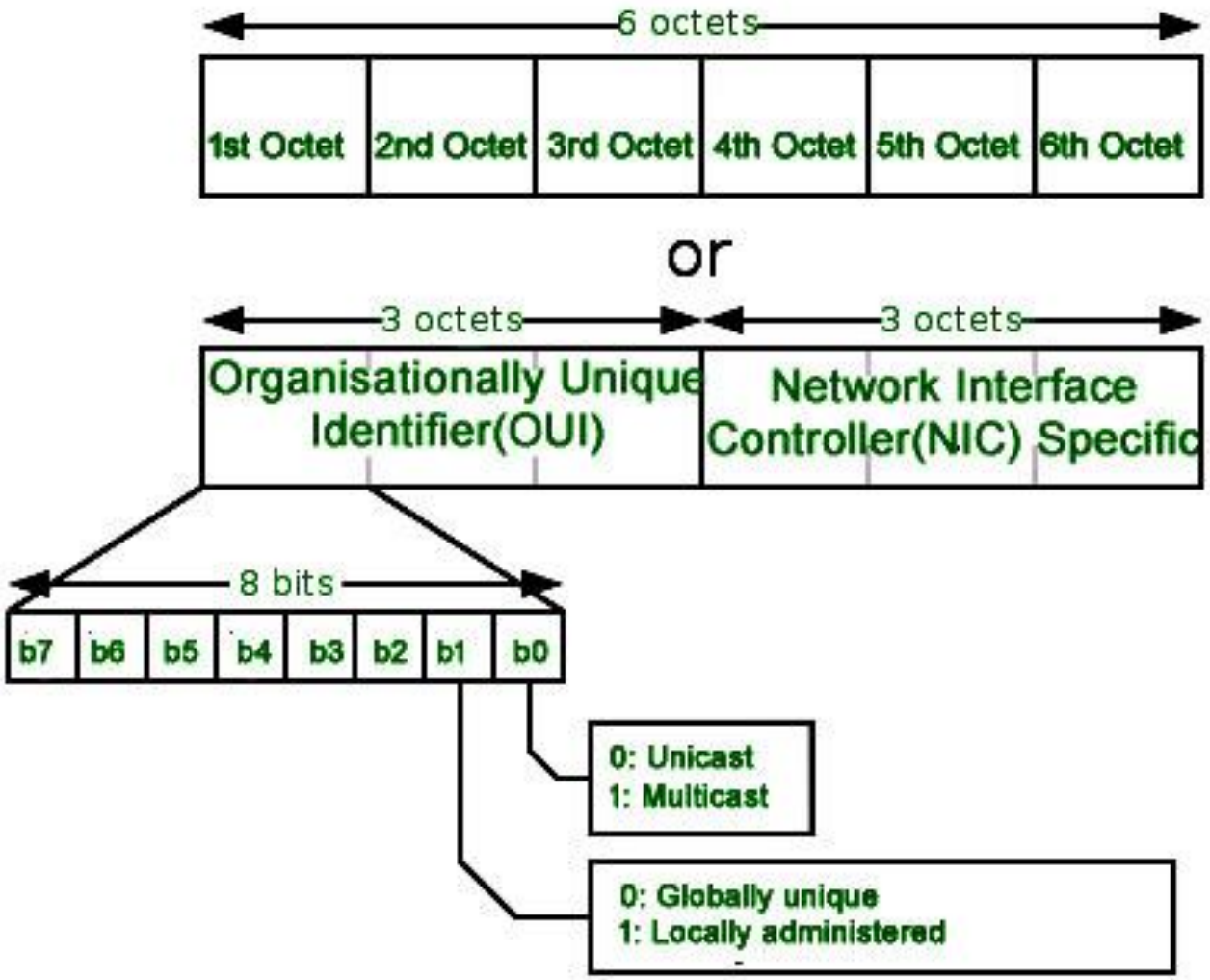
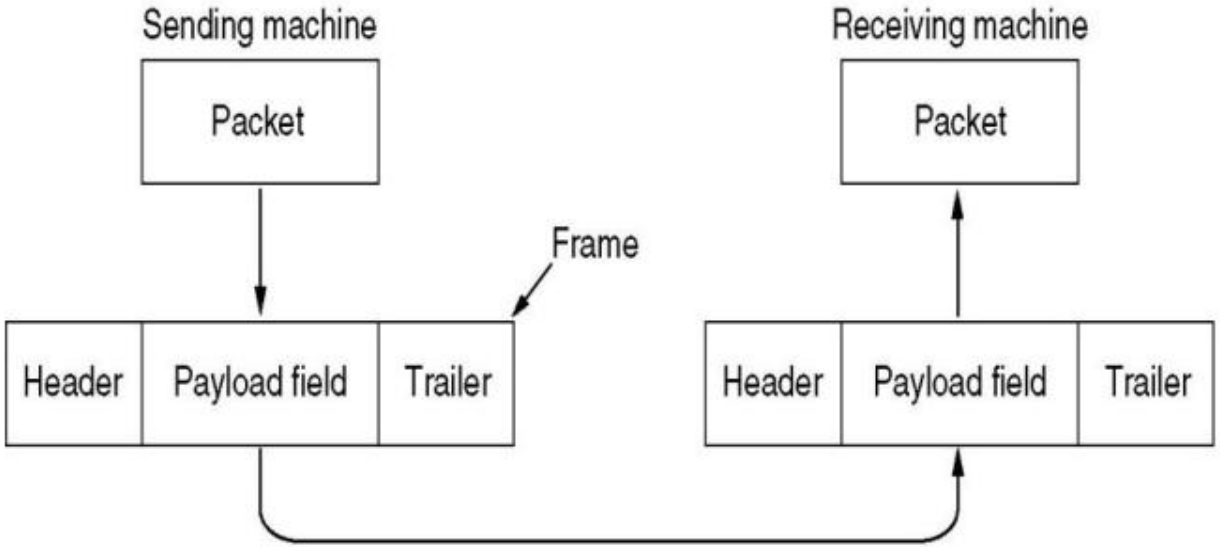


Figure: MAC Address Format

Framing

- The DLL is responsible for taking the packets of information that it receives from the network layer and putting them into frames for transmission.
- Each frame holds the payload plus a header and a trailer (overhead).
- It is the frames that are transmitted over the physical layer



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**
   - 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)

2. **Variable size format**
   - 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

Header	Payload	Trailer	Header	Payload	Trailer
Frame 1			Frame 2		

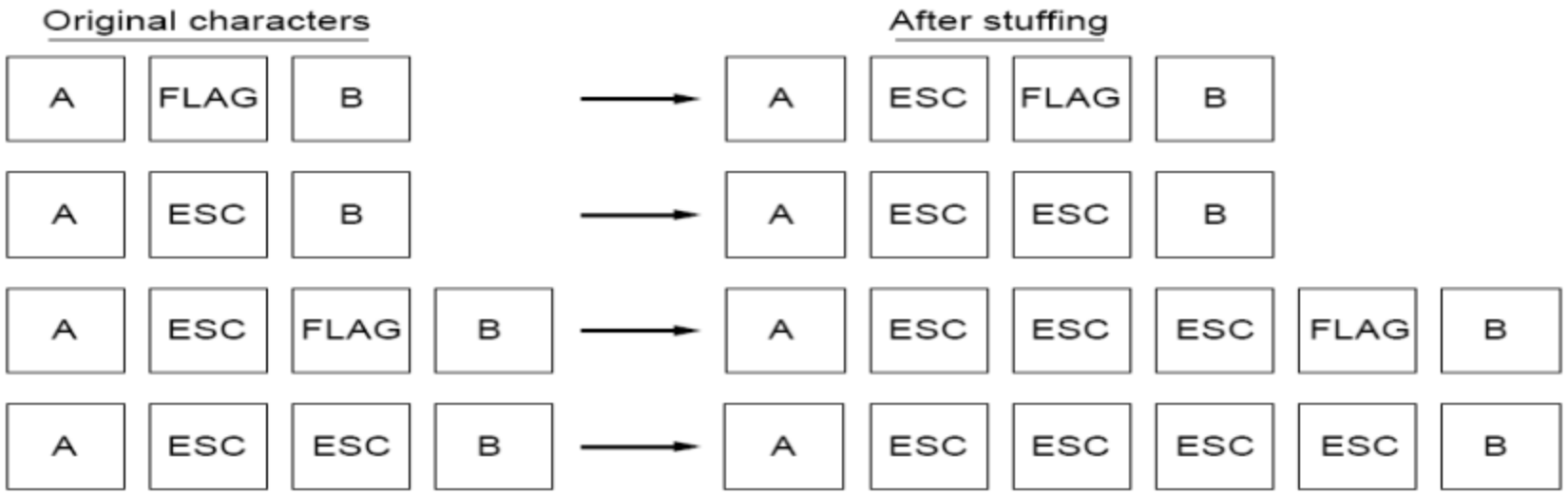
Figure : Frame Format in Variable Size frame

Two approaches were used for this purpose defining frame boundary:

1. Character(Byte) -oriented approach
2. Bit-oriented approach

1.Byte stuffing •The sender inserts a special byte (e.g., ESC) just before each "accidental" flag byte in the data (like in C language, " is replaced with \").

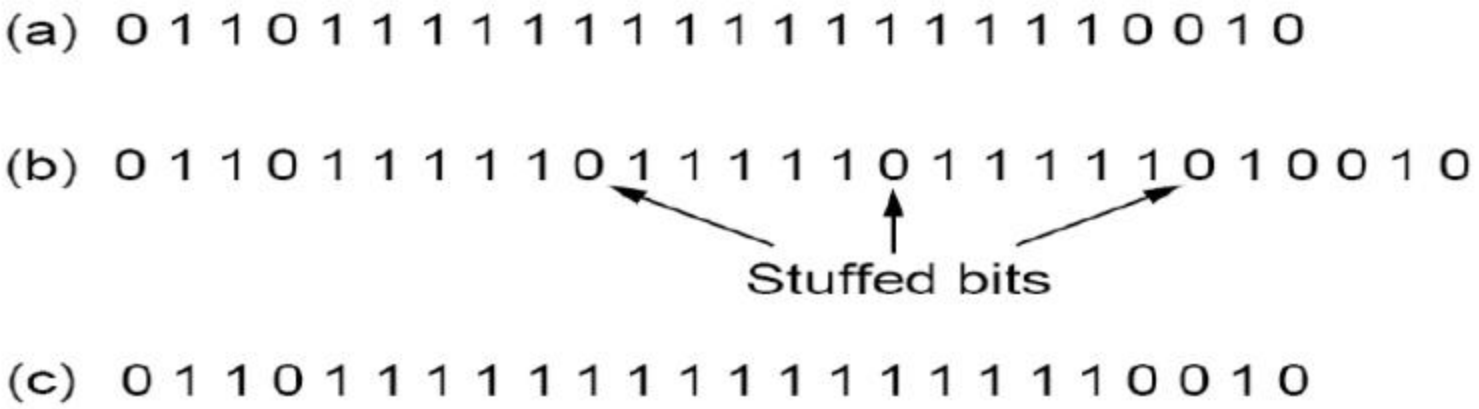
Rules: - Replace each FLAG in data with ESC FLAG - Replace each ESC in data with ESC ESC Four examples of byte sequences before and after byte stuffing



2. Bit stuffing: each frame starts with a flag byte “01111110”.

- Whenever the sender encounters five consecutive 1s in the data, it automatically stuffs a 0 bit into the outgoing bit stream.
- When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically deletes the 0 bit

Bit stuffing:



Bit stuffing:
(a)The original data.

(b)The data as they appear on the line.

(c)The data as they are stored in the receiver’s memory after destuffing.

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)

The Ethernet frame contains seven fields as shown in figure

1. 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

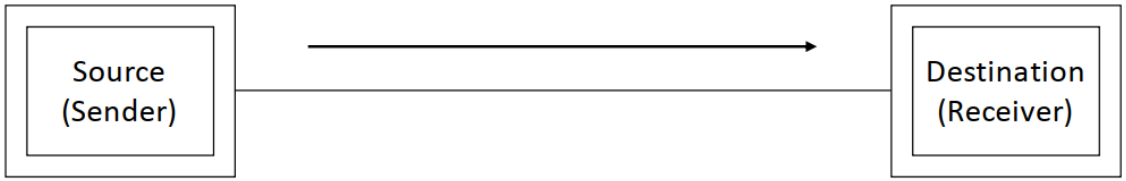
- The pattern provides only an alert and a timing pulse
- The 56-bit pattern allows the stations to miss some bits at the beginning of the frame
- The preamble is actually added at the physical layer and is not (formally) part of the frame.

2.SFD

- The second field is Start Frame Delimiter(SFD) is of 1byte
- 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. Destination Address

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



4. Source Address

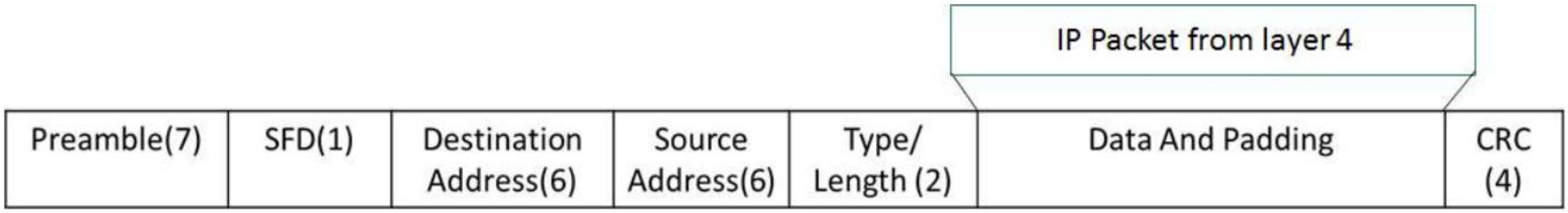
- The SA field is 6 bytes
- It contains the physical address(MAC) of the source station or stations to receive the packet

5. 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**

6. 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.



7. CRC

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

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

The possibility of two types of errors

- **Lost frame:** A frame fails to arrive at the other side. For example, a noise burst may damage a frame to the extent that the receiver is not aware that a frame has been transmitted.
- **Damaged frame:** A recognizable frame does arrive, but some of the bits are in error (have been altered during transmission).

Requirements for error control mechanism:

- **Error detection** - The sender and receiver, either both or any, must ascertain that there is some error in the transit.
- **Positive ACK** - When the receiver receives a correct frame, it should acknowledge it.
- **Negative ACK** - When the receiver receives a damaged frame or a duplicate frame, it sends a NACK back to the sender and the sender must retransmit the correct frame.
- **Retransmission:** The sender maintains a clock and sets a timeout period. If an acknowledgement of a data-frame previously transmitted does not arrive before the timeout the sender retransmits the frame, thinking that the frame or it's acknowledgement is lost in transit.

Automatic Repeat ReQuest

- Automatic Repeat ReQuest (ARQ) is a group of error – control protocols for transmission of data over noisy or unreliable communication network.
- These protocols reside in the Data Link Layer and in the Transport Layer of the OSI (Open Systems Interconnection) reference model. They are named so because they provide for automatic retransmission of frames that are corrupted or lost during transmission.
- ARQ is also called Positive Acknowledgement with Retransmission (PAR).
- ARQs are often used in Global System for Mobile (GSM) communication.

Working Principle

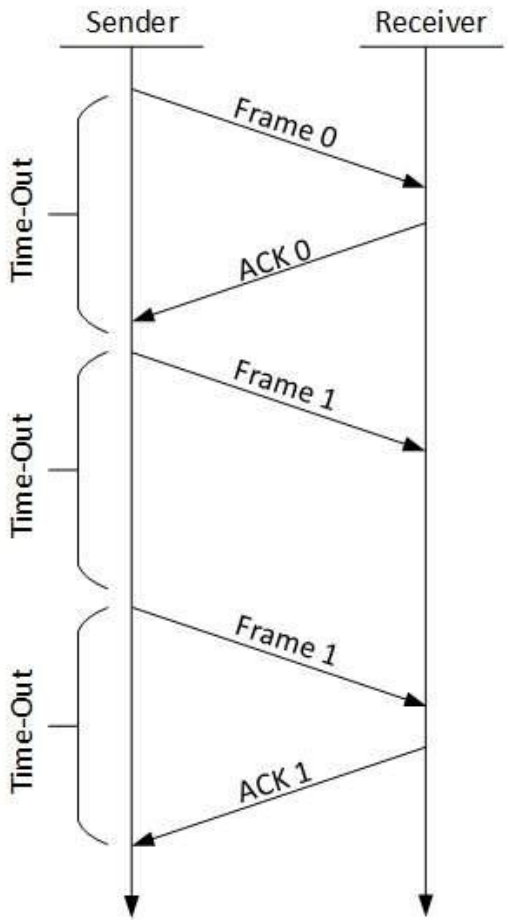
- In these protocols, the receiver sends an acknowledgement message back to the sender if it receives a frame correctly.
- If the sender does not receive the acknowledgement of a transmitted frame before a specified period of time, i.e. a timeout occurs, the sender understands that the frame has been corrupted or lost during transit.
- So, the sender retransmits the frame. This process is repeated until the correct frame is transmitted.

Types of ARQ:
1. **Stop and Wait ARQ**
2. **Sliding Window Protocol**
   a. Go-Back-N
   b. Selective Reject

Stop-and-wait ARQ

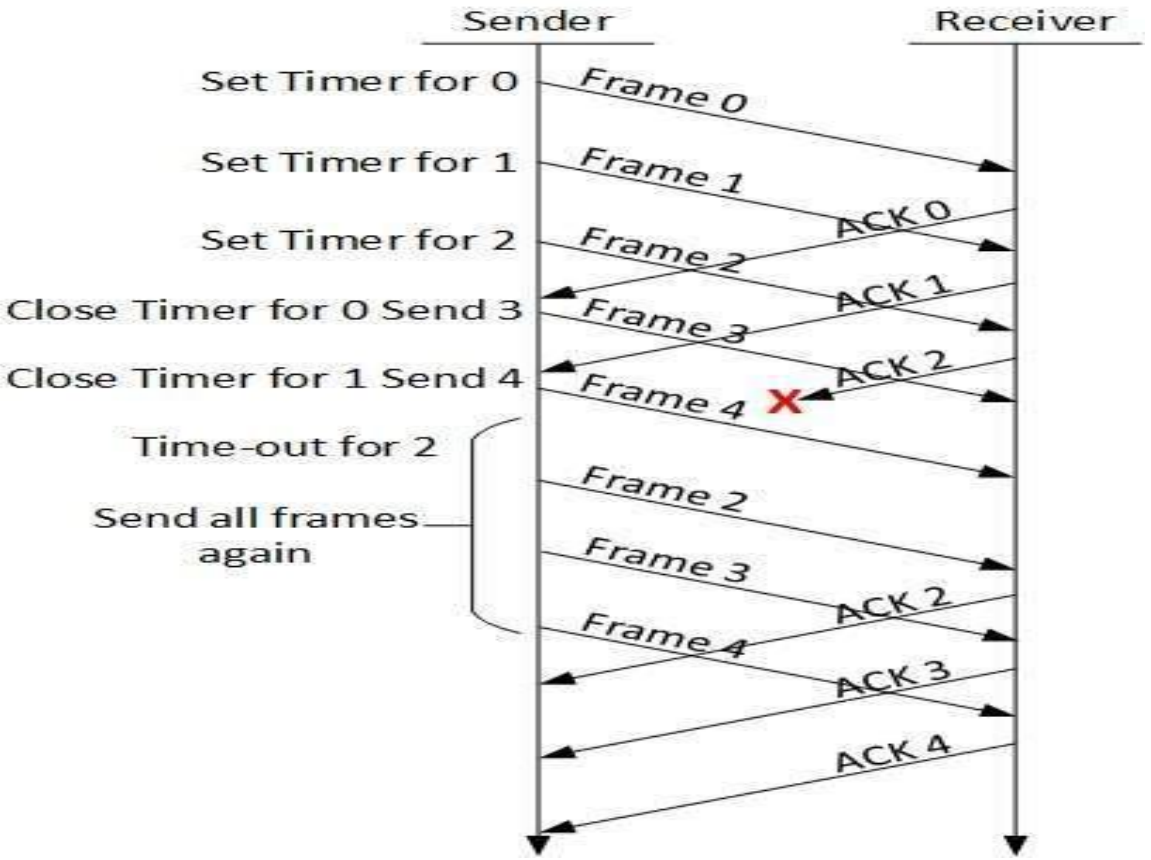
The following transition may occur in Stop-and-Wait ARQ:

- The sender maintains a timeout counter.
- When a frame is sent, the sender starts the timeout counter.
- If acknowledgement of frame comes in time, the sender transmits the next frame in queue.
- If acknowledgement does not come in time, the sender assumes that either the frame or its acknowledgement is lost in transit. Sender retransmits the frame and starts the timeout counter.
- If a negative acknowledgement is received, the sender retransmits the frame.


Sender
Receiver
Frame 0
Time-Out
ACK 0
Frame 1
Time-Out
Frame 1
Time-Out
ACK 1

Go-Back-N ARQ

- Stop and wait ARQ mechanism does not utilize the resources at their best. When the acknowledgement is received, the sender sits idle and does nothing.
- In Go-Back-N ARQ method, both sender and receiver maintain a window.



- The sending-window size enables the sender to send multiple frames without receiving the acknowledgement of the previous ones.

- The receiving-window enables the receiver to receive multiple frames and acknowledge them.

- The receiver keeps track of incoming frame's sequence number.

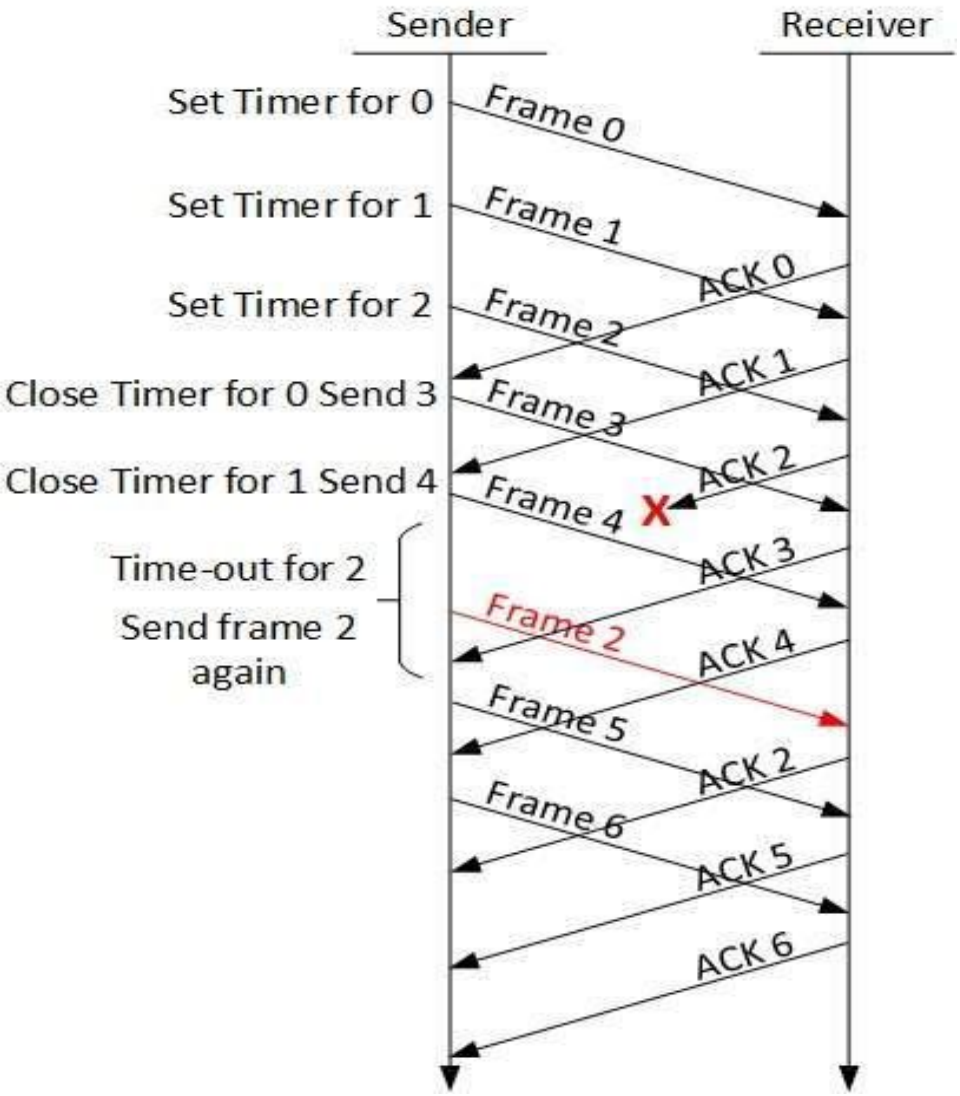
- When the sender sends all the frames in window, it checks up to what sequence number it has received positive acknowledgement.

- If all frames are positively acknowledged, the sender sends next set of frames.

- If sender finds that it has received NACK or has not received any ACK for a particular frame, it retransmits all the frames after which it does not receive any positive ACK.

Selective Repeat ARQ

- In Go-back-N ARQ, it is assumed that the receiver does not have any buffer space for its window size and has to process each frame as it comes. This enforces the sender to retransmit all the frames which are not acknowledged.
- In Selective-Repeat ARQ, the receiver while keeping track of sequence numbers, buffers the frames in memory and sends NACK for only frame which is missing or damaged.
- The sender in this case, sends only packet for which NACK is received.


Sender
Receiver
Set Timer for 0
Frame 0
Set Timer for 1
Frame 1
ACK 0
Set Timer for 2
Frame 2
ACK 1
Close Timer for 0 Send 3
Frame 3
ACK 2
Close Timer for 1 Send 4
Frame 4
X
ACK 3
Time-out for 2
Send frame 2 again
Frame 2
ACK 4
Frame 5
ACK 2
Frame 6
ACK 5
ACK 6

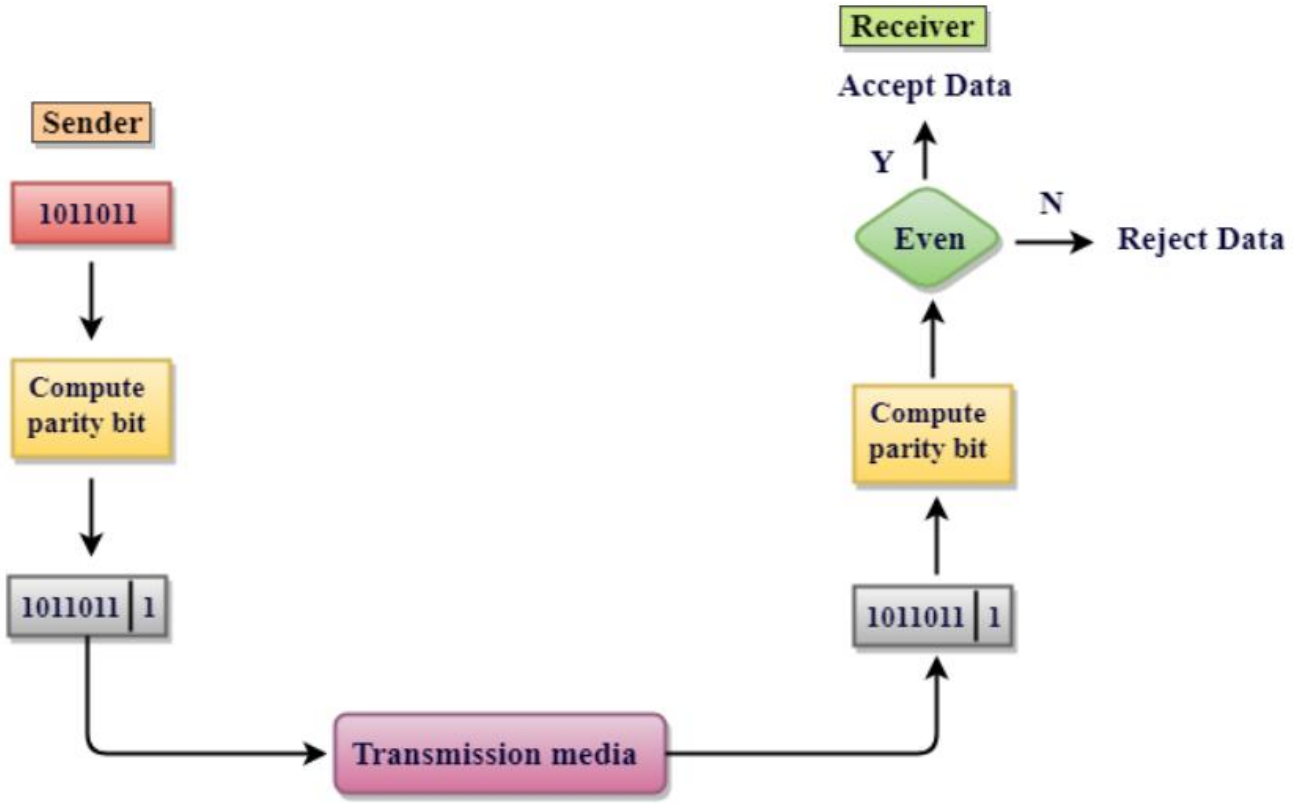
Error Control

- When data-frame is transmitted, there is a probability that data-frame may be lost in the transit or it is received corrupted.
- In both cases, the receiver does not receive the correct data-frame and sender does not know anything about any loss.
- In such case, both sender and receiver are equipped with some protocols which help them to detect transit errors such as loss of data-frame.
- Hence, either the sender retransmits the data-frame or the receiver may request to resend the previous data-frame.

Error Detecting Techniques:

1. Simple Parity Check

- Single Parity checking is the simple mechanism and inexpensive to detect the errors.
- In this technique, a redundant bit is also known as a parity bit which is appended at the end of the data unit so that the number of 1s becomes even. Therefore, the total number of transmitted bits would be 9 bits.
- If the number of 1s bits is odd, then parity bit 1 is appended and if the number of 1s bits is even, then parity bit 0 is appended at the end of the data unit.
- At the receiving end, the parity bit is calculated from the received data bits and compared with the received parity bit.
- This technique generates the total number of 1s even, so it is known as even-parity checking.

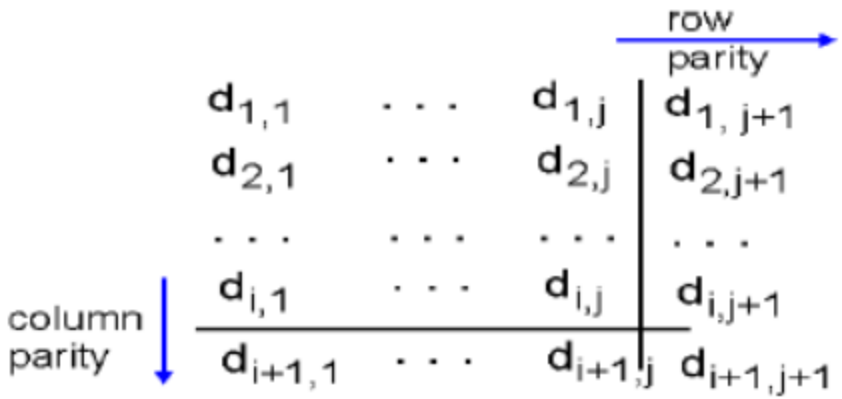


Drawbacks Of Single Parity Checking

- It can only detect single-bit errors which are very rare.
- If two bits are interchanged, then it cannot detect the errors.

Two Dimensional Bit Parity:

Detect *and correct* single bit errors




```
10101|1
11110|0
01110|1
-----+-
00101|0
```

no errors

```
10101|1
10110|0
01110|1
-----+-
00101|0
```

parity error

parity error

correctable single bit error

2. Checksum

- In checksum error detection scheme, the data is divided into k segments each of m bits.

In the sender's end

- the segments are added using 1's complement arithmetic to get the sum. The sum is complemented to get the checksum.
- The checksum segment is sent along with the data segments.

At the receiver's end,

- all received segments are added using 1's complement arithmetic to get the sum. The sum is complemented.
- If the result is zero, the received data is accepted; otherwise discarded.

E.g. : 10110011 10101011 to be transmitted . Using checksum method, find whether there is error or not

Tx:

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

1 0 1 0 1 1 1 1 0
1

0 1 0 1 1 1 1 1

1's complement .....

1 0 1 0 0 0 0 0
check sum

Rx:

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

1 0 1 0 1 1 1 1 0
1

0 1 0 1 1 1 1 1
1 0 1 0 0 0 0 0 check sum

1 1 1 1 1 1 1 1

1's complement .....
(0 0 0 0 0 0 0 0)
all zero i.e. there is no error

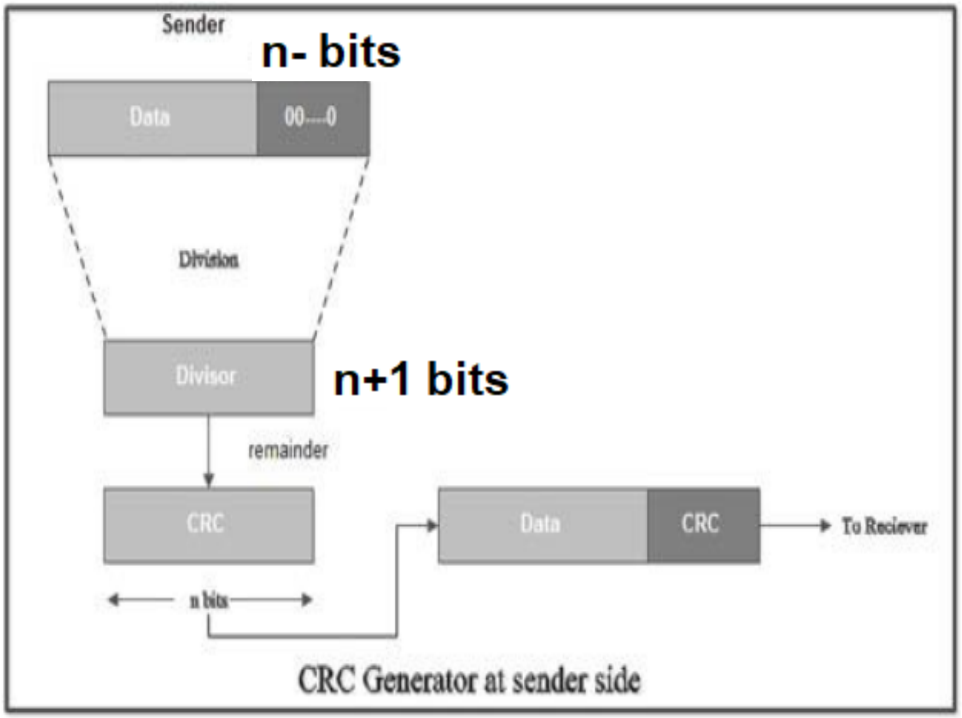
3.Cyclic Redundancy Code

An error detection mechanism in which a special number is appended to a block of data in order to detect any changes introduced during storage (or transmission).

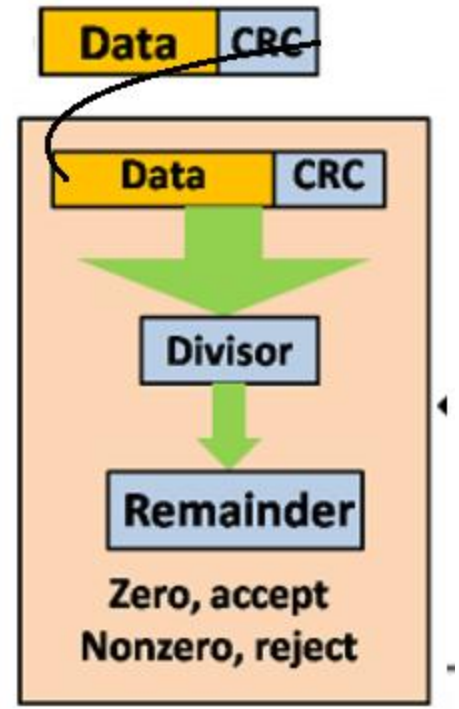
A CRC is derived using a more complex algorithm than the simple CHECKSUM, involving MODULO ARITHMETIC (hence the 'cyclic' name) and treating each input word as a set of coefficients for a polynomial.

Steps:

- At the sender side, the data unit to be transmitted IS divided by a predetermined divisor (binary number) in order to obtain the remainder. This remainder is called CRC.
- The CRC has one bit less than the divisor. It means that if CRC is of n bits, divisor is of n+ 1 bit.
- The sender appends this CRC to the end of data unit such that the resulting data unit becomes exactly divisible by predetermined divisor *i.e.* remainder becomes zero.
- At the destination, the incoming data unit *i.e.* data + CRC is divided by the same number (predetermined binary divisor).
- If the remainder after division is zero then there is no error in the data unit & receiver accepts it.
- If remainder after division is not zero, it indicates that the data unit has been damaged in transit and therefore it is rejected



Sender

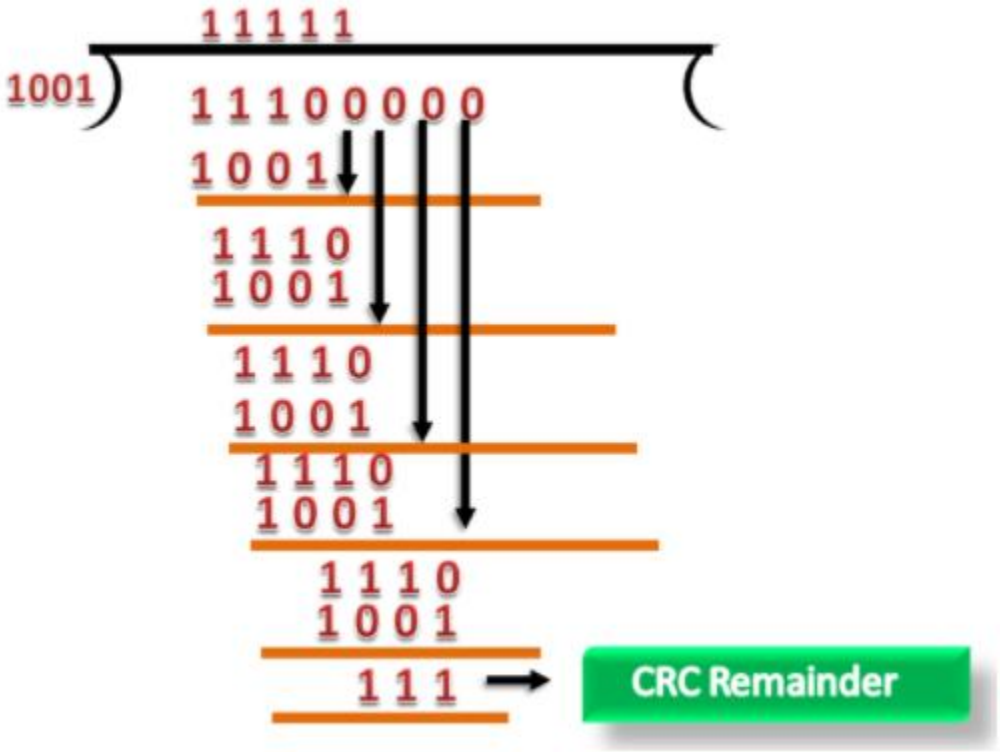


Receiver

Suppose the original data is 11100 and divisor is 1001.

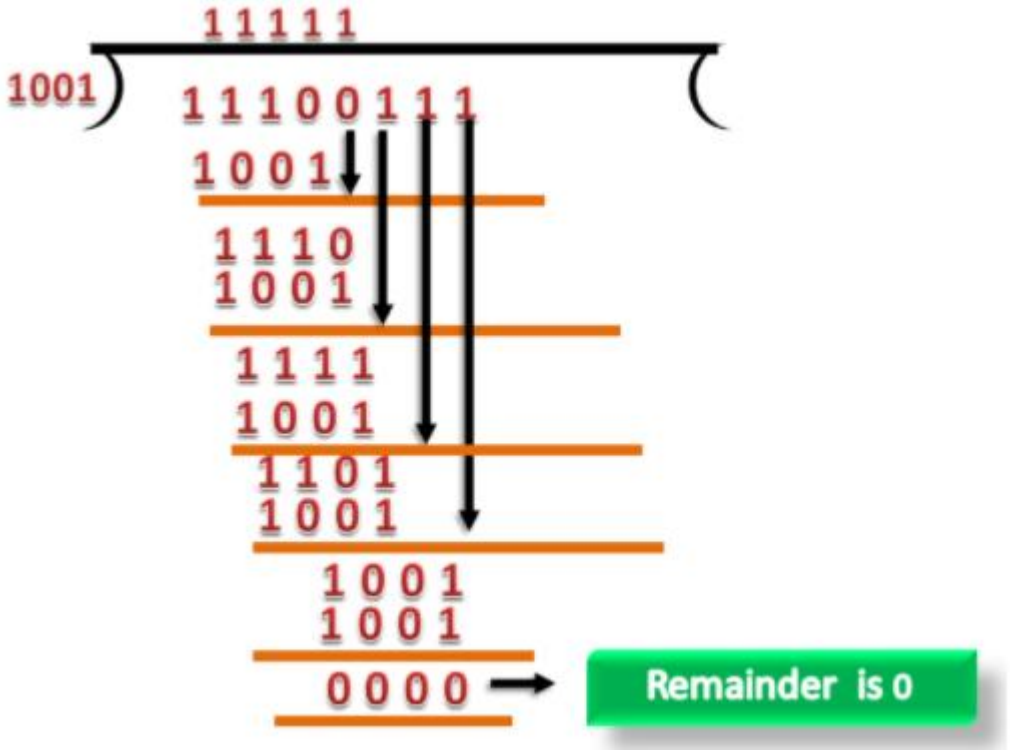
- CRC Generator A CRC generator uses a modulo-2 division. Firstly, three zeroes are appended at the end of the data as the length of the divisor is 4 and we know that the length of the string 0s to be appended is always one less than the length of the divisor.

- Now, the string becomes 11100000, and the resultant string is divided by the divisor 1001.



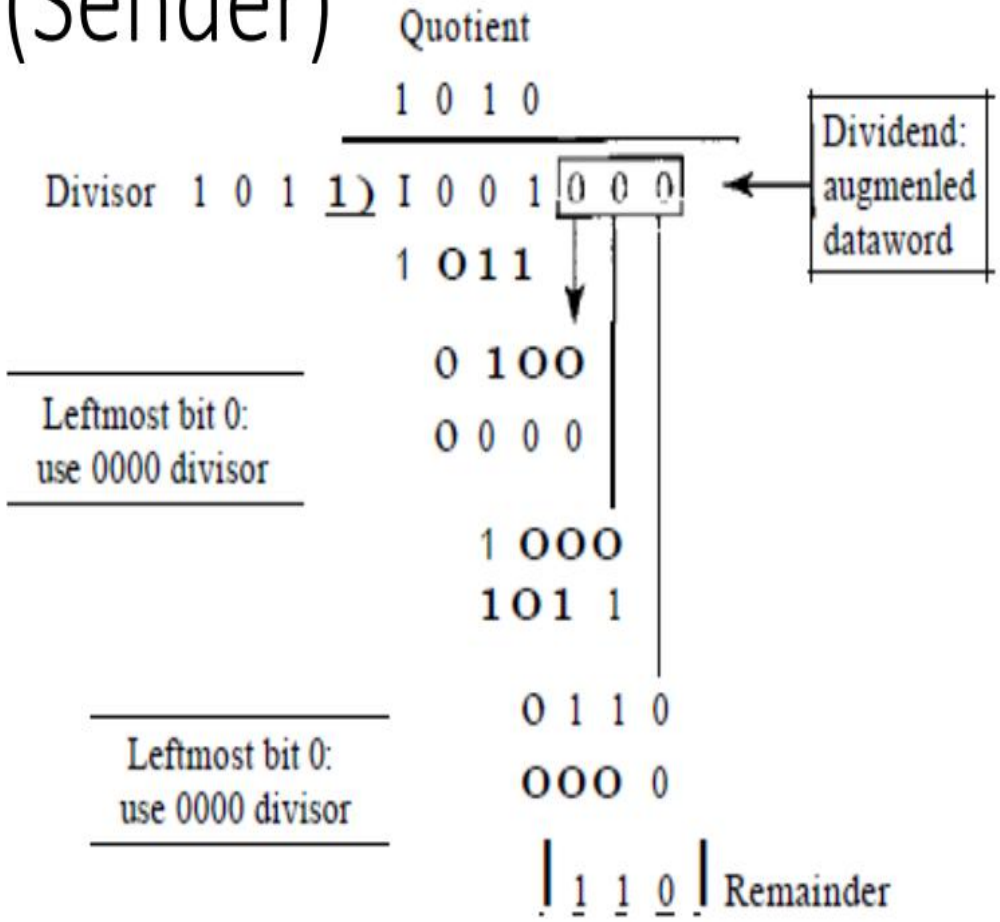
CRC Checker

The functionality of the CRC checker is similar to the CRC generator. When the string 11100111 is received at the receiving end, then CRC checker performs the modulo-2 division.

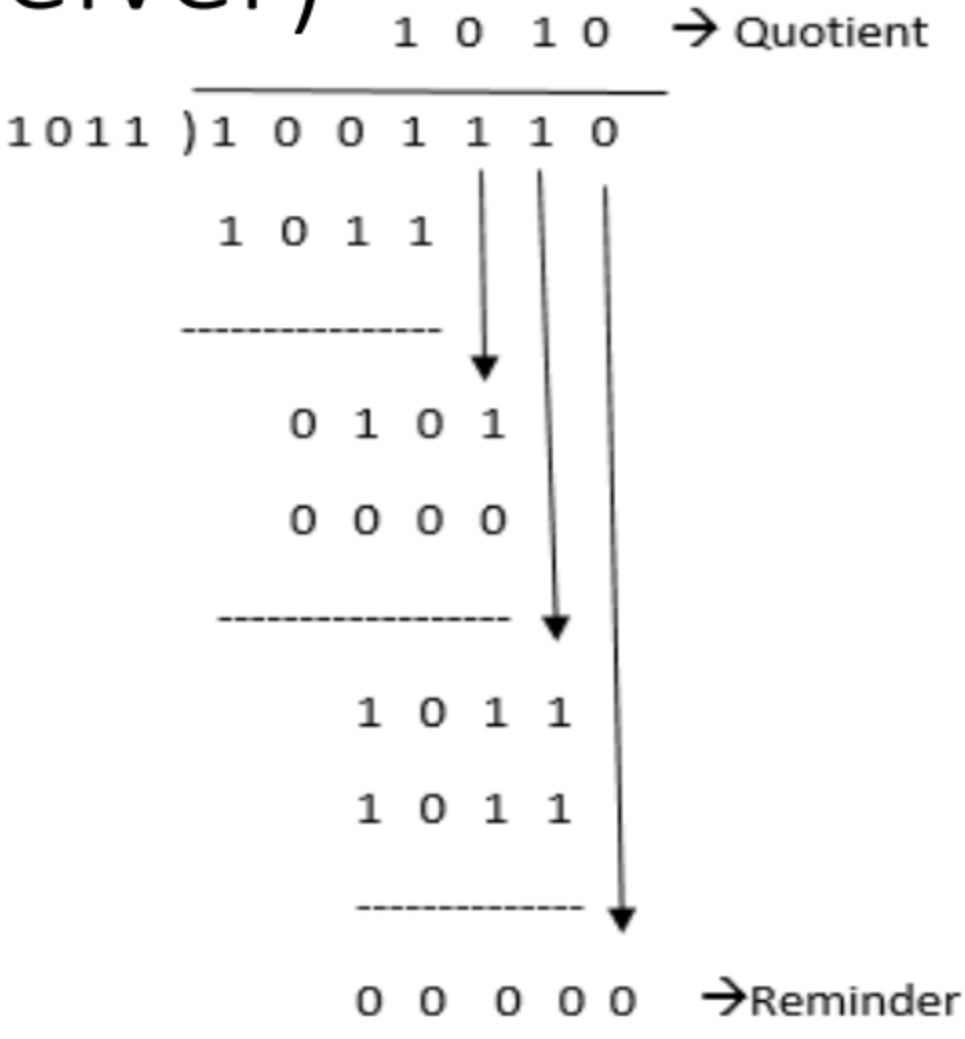


In this case, CRC checker generates the remainder of zero. Therefore, the data is accepted.

Encoder (Sender)



Decoder (Receiver)



4.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)

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

Hamming Code

- Linear Block Code

- Use for error detection and correction
  - Block length $n = 2^m - 1$
  - No of message bits, $k = 2^m - m - 1$
  - No of binary bits, $(n-k) = m$
  - Efficiency of code = k/n

Hamming Code Structure

- Parity bits are inserted in between data bits
- Commonly 7 bits Hamming code is used

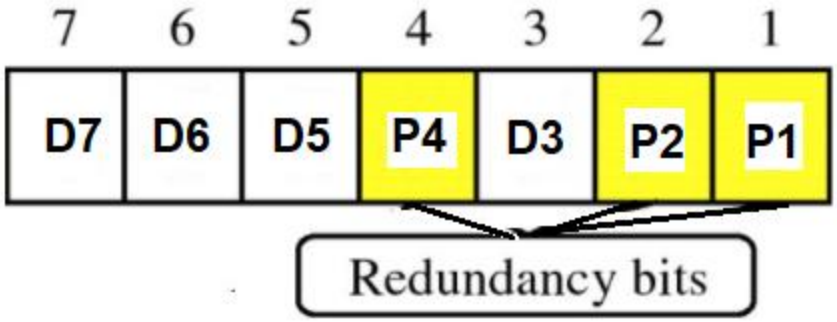


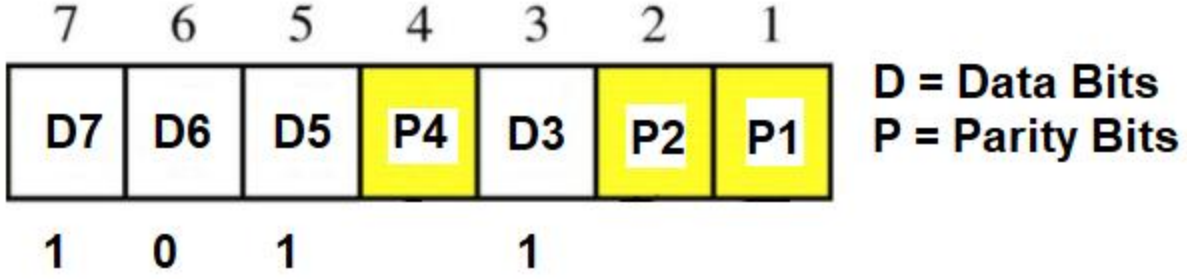
Fig : 7 bit Hamming code

D = Data Bits
P = Parity Bits

Selection:

P1 = 1,3,5,7
P2 = 2,3,6,7
P3 = 4,5,6,7

For bit 1011 to be transmitted, construct even parity 7-bit Hamming code



Calculate

P1 = 1,3,5,7 = (P1,1,1,1) = 1 to be even

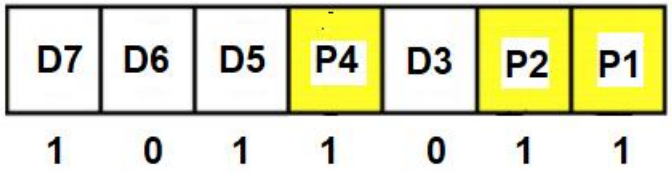
P2 = 2,3,6,7 = (P2,1,0,1) = 0

P4 = 4,5,6,7 = (P4,1,0,1) = 0

1	0	1	0	1	0	1

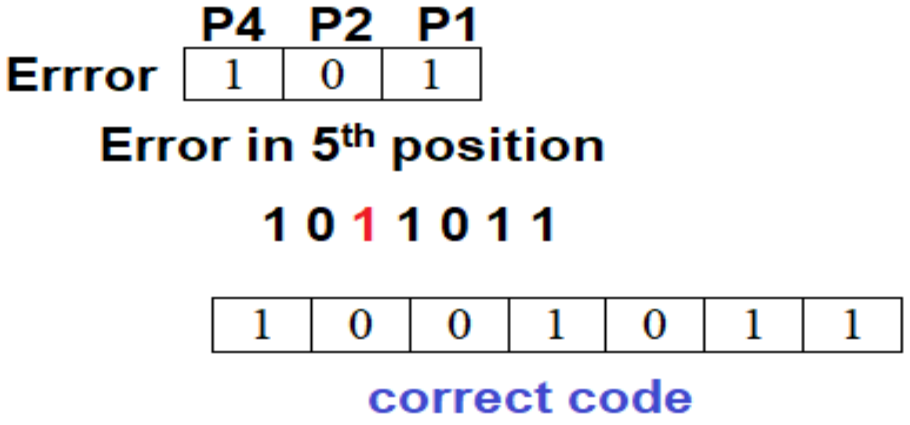
Codeword receiver=1011011; Assume even parity, state whether received codeword is correct or incorrect. If incorrect locate the bit error

Solution:



P1 = 1,3,5,7 = 1011 = odd Parity; error so put P1 =1
P2 = 2,3,6,7 = 1001= Even Parity ; no error so put P2 =0
P4 = 4,5,6,7 = 1101 = odd Parity ; error so put P4 =1

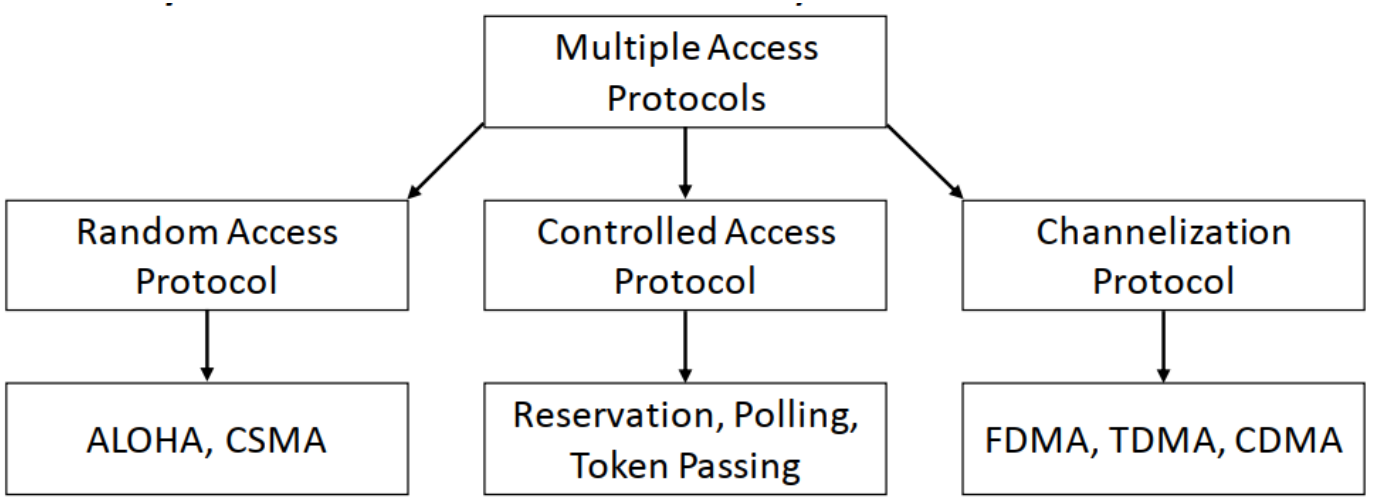
Thus error exist



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)



3.5 CHANNEL ALLOCATION TECHNIQUE

RANDOM ACCESS PROTOCOLS

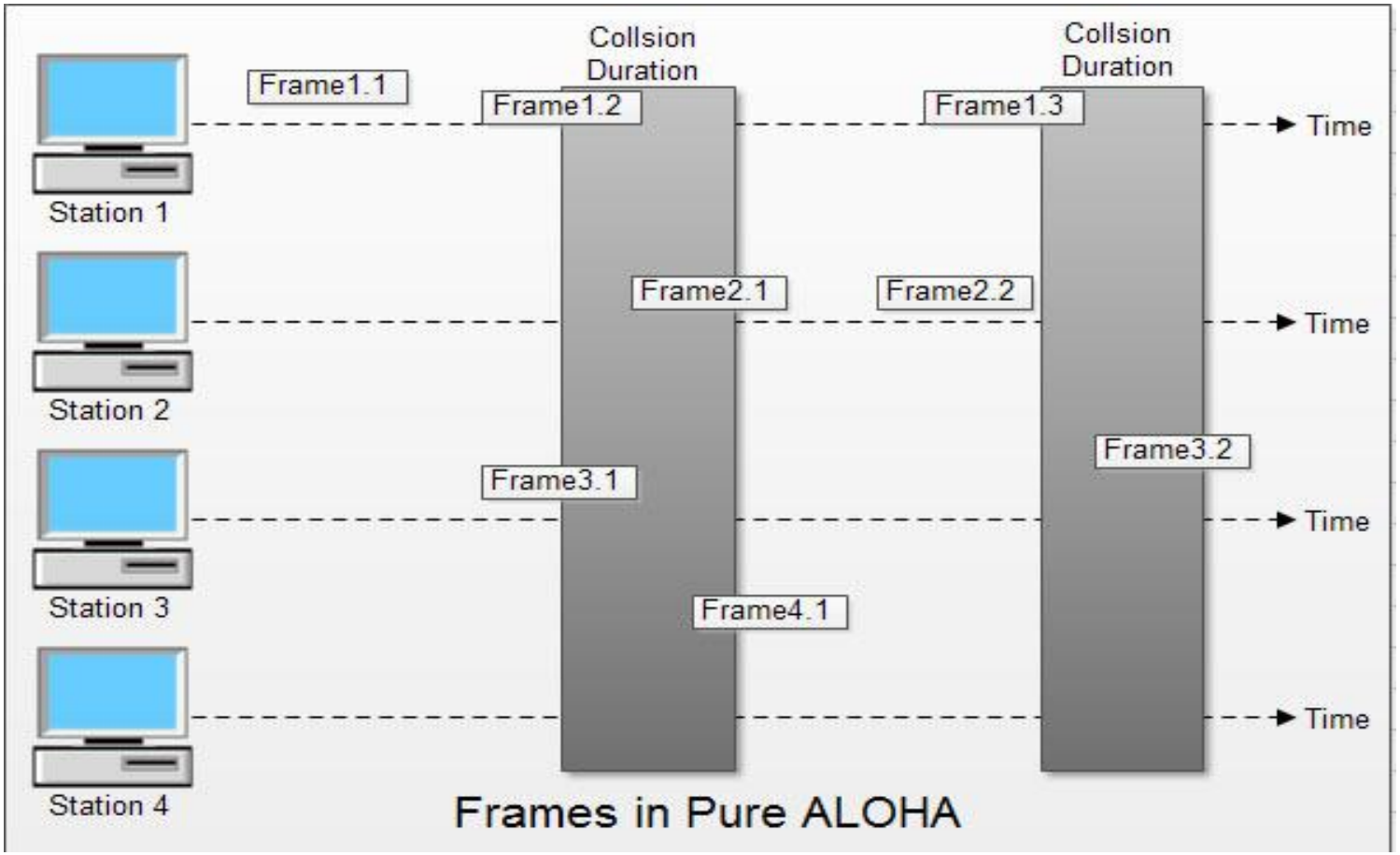
- system for **a shared communication** Networks channel.
- requires a method of **handling collisions**
- occur when **two or more systems** attempt to transmit on the **channel** at the **same time.**
- In the **ALOHA system**, a node transmits **whenever data is available** to send.
- If **another node transmits at the same time**, a collision occurs, and the frames that were transmitted are lost.

There are two different version/types of ALOHA:

(i) Pure ALOHA
(ii) Slotted ALOHA

Pure ALOHA

- In pure ALOHA, **whenever any station transmits a frame**, it expects the **acknowledgement** from the receiver.
- If **acknowledgement is not received** within specified time, the station assumes that the **frame has been destroyed**.
- If the **frame is destroyed** because of collision the station **waits** for a **random amount of time** and **sends it again.**
- This **waiting time must be random** otherwise same **frames will collide again and again.**
- Figure shows an example of frame collisions in pure ALOHA.

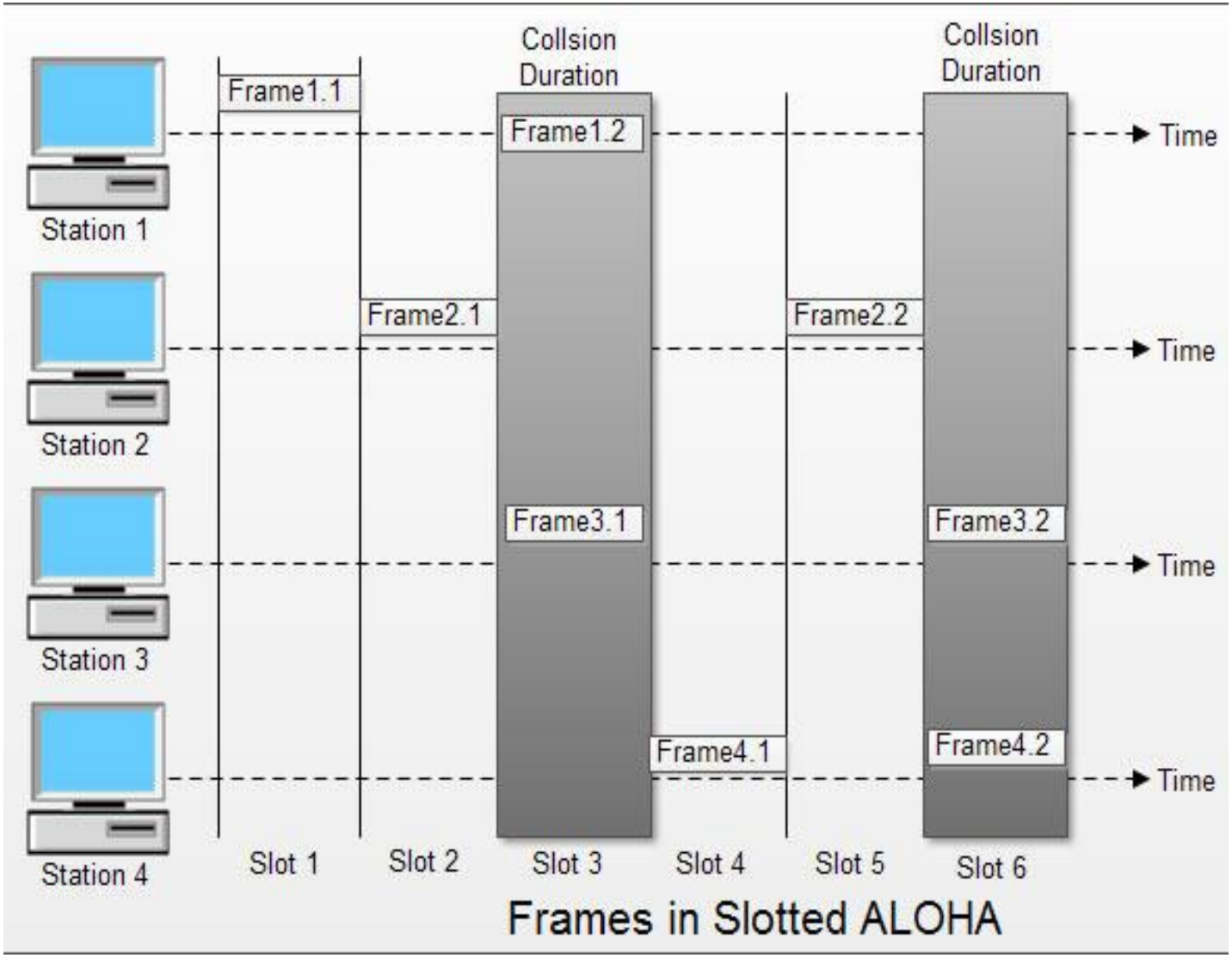

Collsion
Duration
Collsion
Duration
Frame1.1
Frame1.2
Frame1.3
Time
Station 1
Frame2.1
Frame2.2
Time
Station 2
Frame3.2
Frame3.1
Time
Station 3
Frame4.1
Time
Station 4
Frames in Pure ALOHA

Slotted ALOHA

- Slotted ALOHA was **invented to improve the efficiency of pure ALOHA** as chances of collision in pure ALOHA are very high.

- In slotted ALOHA, **the time of the shared channel is divided into discrete intervals called slots.**

- The stations can send **a frame only at the beginning of the slot** and only **one frame is sent in each slot**.


Collsion Duration
Collsion Duration
Frame1.1
Frame1.2
Frame2.1
Frame2.2
Frame3.1
Frame3.2
Frame4.1
Frame4.2
Time
Time
Time
Time
Station 1
Station 2
Station 3
Station 4
Slot 1
Slot 2
Slot 3
Slot 4
Slot 5
Slot 6
Frames in Slotted ALOHA

- In slotted ALOHA, **if any station is not able to place the frame** onto the channel at the beginning of the slot

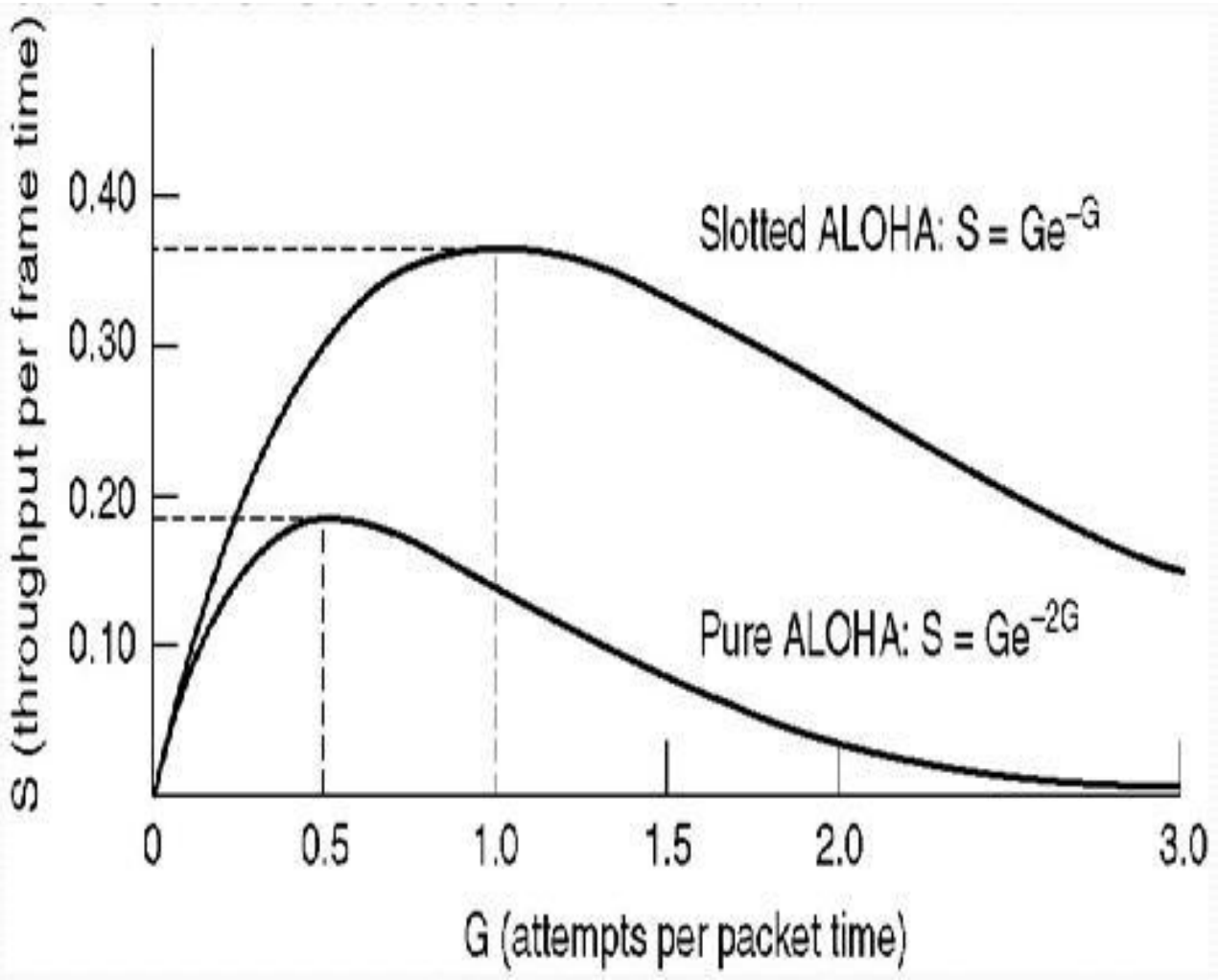
- then the station has to **wait until the beginning of the next time slot**.

- In slotted ALOHA, there is **still a possibility of collision** if **two stations try to send at the beginning of the same time slot**

- Slotted ALOHA still has an edge over pure ALOHA as **chances of collision are reduced to one-half.**

Key Differences Between Pure ALOHA and Slotted ALOHA

- Pure ALOHA was introduced by Norman and his associates at the university of Hawaii in 1970.
- On the other hand, Slotted ALOHA was introduced by Roberts in 1972.
- In pure ALOHA, whenever a station has data to send it transmits it without waiting whereas, in slotted ALOHA a user wait till the next time slot beings to transmit the data.
- In pure ALOHA the time is continuous whereas, in Slotted ALOHA the time is discrete and divided into slots.
- In pure ALOHA the probability of successful transmission is S=G*e^-2G. On the other hand, in slotted ALOHA the probability of successful transmission is S=G*e^-G.
- The maximum throughput occurs at G=1/2 which is 18 % whereas, the maximum throughput occurs at G=1 which is 37%.


S (throughput per frame time)
0.40
0.30
0.20
0.10
Slotted ALOHA: S = Ge^-G
Pure ALOHA: S = Ge^-2G
0
0.5
1.0
1.5
2.0
3.0
G (attempts per packet time)

3.6. ETHERNET STANDARDS

1. CSMA, or listen with random access carrier

- CSMA (Carrier Sense Multiple Access) is to **listen to the channel before transmitting.**
- This significantly **reduces the risk of collision**, but **does not eliminate them completely.**
- If more remote stations, **a coupler does not detect the transmission** of a frame, and there may be **signal superposition**.
- it is necessary to subsequently **retransmit lost frames.**

2. 802.3 CSMA / CD

- **(Carrier Sense Multiple Access / Collision Detection)**

- a set of rules determining **how network devices respond when two devices attempt to use a data channel simultaneously** (called a *collision*).

- networks use CSMA/CD to **physically monitor the traffic** on the line at participating stations.

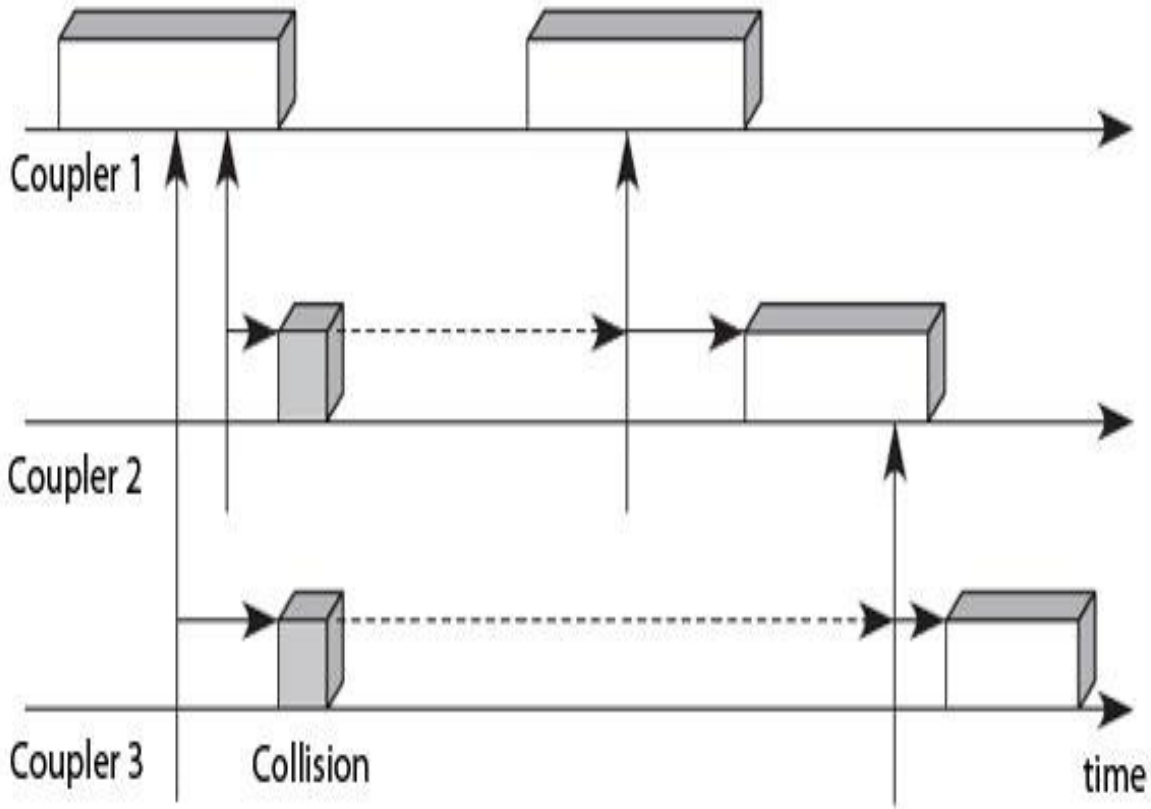
- If **no transmission** is taking place at the time, **the particular station can transmit.**

- If **two stations** attempt to **transmit simultaneously**, this **causes a collision**, which is **detected by all participating stations.**

- After a **random time interval**, the stations that collided **attempt to transmit again.**

- If **another collision occurs**, the **time intervals from which the random waiting time is selected are increased step by step.**

- This is known as **exponential back off**



Operating Principle of CSMA / CD

- **CSMA/CD Algorithm**

- **Step 1**: Before starting the transmission, the station senses the medium for the presence of any transmission from other stations.
- **Step 2**: If the medium is idle, the station starts transmission and goes to step 4; otherwise, it goes to step 3.
- **Step 3**: If the medium is busy, it continuously listens until the channel becomes idle, then it starts immediately.
- **Step 4**: If a collision detected during the transmission, a brief jamming signal is transmitted to inform all the stations to stop transmission immediately.
- **Step 5**: After transmitting the jamming signal, the station waits for a random time and attempts to start from step 1 again.

This CSMA/CD is a significant protocol. It has included in the IEEE standard.

3. CSMA/CA (Collision Avoidance)

CSMA/CA protocol is used in wireless networks because they cannot detect the collision so the only solution is collision avoidance.

CSMA/CA avoids the collisions using three basic techniques.

1. Inter Frame Space
2. Contention window
3. Acknowledgments

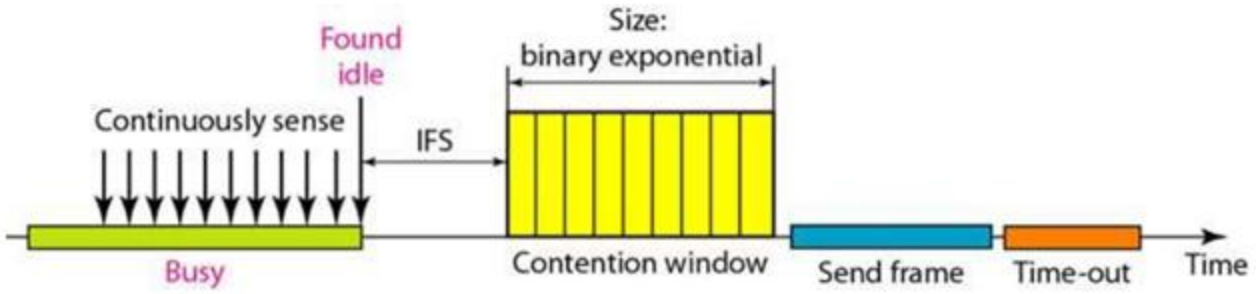


Figure : Timing in CSMA/CA

1. Inter Frame Space (IFS)

- Whenever the **channel is found idle**, the station does not transmit immediately.
- It **waits** for a period of time called **Inter frame space (IFS)**.
- it may be possible that same distant station **may have already started transmitting**
- and the signal of that distant station has not yet reached other stations.

2. Contention Window

- Contention window 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.**
- In contention window the station **needs to sense the channel after each time slot.**

3. Acknowledgement

- Despite all the precautions, **collisions may occur and destroy the data.**
- The **positive acknowledgment and the time-out timer** can help guarantee that receiver has received the frame.

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.

- 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

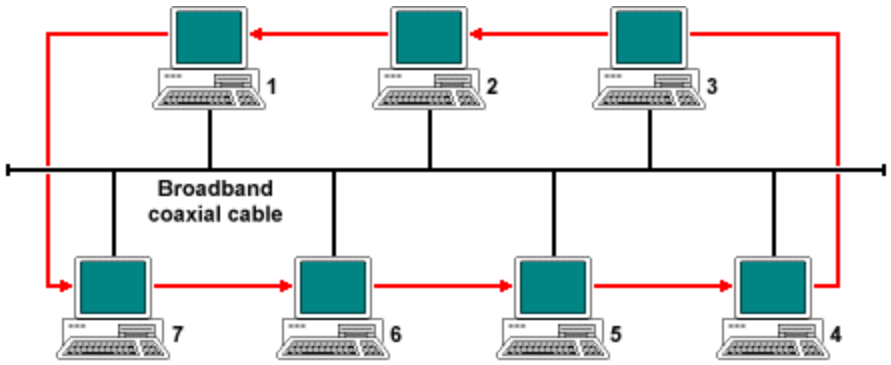


Figure : Token Bus Network (Red Arrow Indicates Token Passing Sequence)

802.5 Token Ring

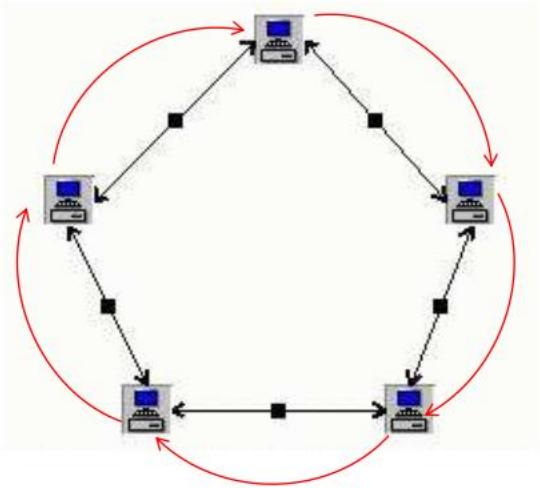


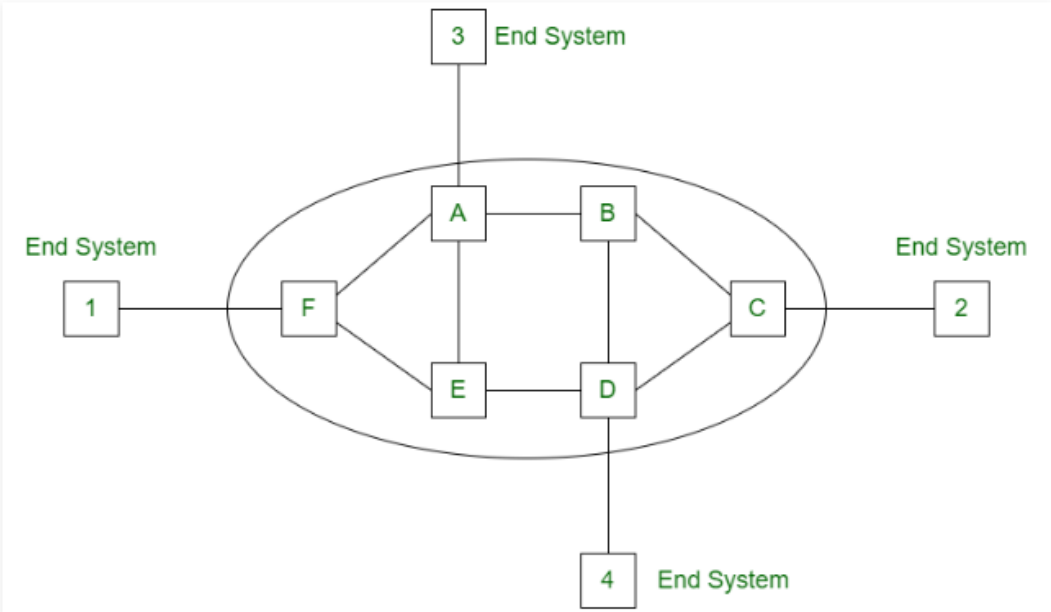
Figure : Token Bus Network (Red Arrow Indicates Token Passing Sequence)

- 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

- 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

3.8 Virtual Circuit

- **Virtual Circuit** is the computer network providing connection-oriented service.
- It is a connection-oriented network.
- In virtual circuit resource are reserve for the time interval of data transmission between two nodes.
- This network is a highly reliable medium of transfer. Virtual circuits are costly to implement.



Working of Virtual Circuit:

- In the first step a medium is set up between the two end nodes.
- Resources are reserved for the transmission of packets.
- Then a signal is sent to sender to tell the medium is set up and transmission can be started.
- It ensures the transmission of all packets.
- A global header is used in the first packet of the connection.
- Whenever data is to be transmitted a new connection is set up.

Asynchronous Transfer Mode (ATM)

- ATM stands for Asynchronous Transfer Mode. It is a switching technique that uses time division multiplexing (TDM) for data communications.

  OR

- Asynchronous transfer mode (ATM) is a switching technique used by telecommunication networks that uses asynchronous time-division multiplexing to encode data into small, fixed-sized cells.

- ATM networks are connection oriented networks for cell relay that supports voice, video and data communications.

- It encodes data into small fixed - size cells so that they are suitable for TDM and transmits them over a physical medium.

- The size of an ATM cell is **53 bytes**: **5 byte header and 48 byte** payload. There are two different cell formats - user-network interface (UNI) and network-network interface (NNI).

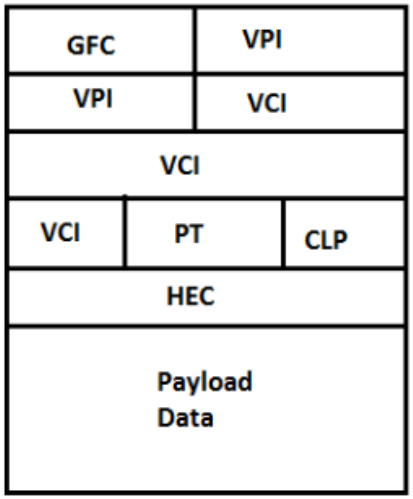
i. UNI Header: This is used within private networks of ATMs for communication between ATM endpoints and ATM switches. It includes the Generic Flow Control (GFC) field.

ii. NNI Header: is used for communication between ATM switches, and it does not include the Generic Flow Control(GFC) instead it includes a Virtual Path Identifier (VPI) which occupies the first 12 bits.

GFC: Generic Flow Control

VPI: Virtual Path Identifier

VCI: Virtual Channel Identifier

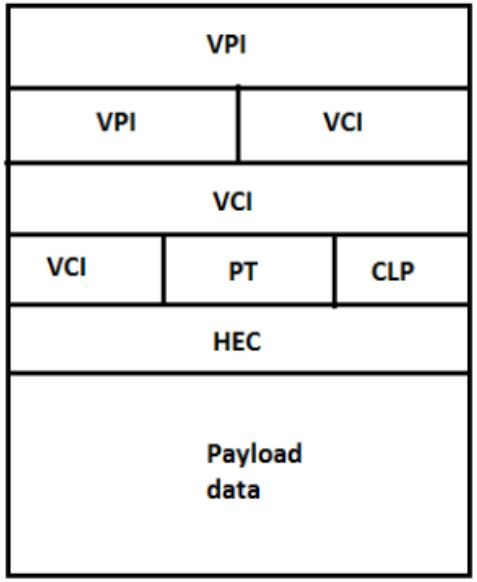


UNI Frame format

PT: Payload type

CLP: Cell loss priority

HEC: Header Error control



NNI Frame format

- **Header Format**
- **GFC:** Provides local functions such as identifying multiple stations that share a single ATM interface.
- **VPI:** In conjunction with the VCI, identifies the next destination of a cell as it passes through a series of ATM switches.
- **VCI:** In conjunction with the VPI, identifies the next destination of a cell as it passes through a series of ATM switches. The Virtual Channel Identifier (VCI) is used for routing to and from the end user.
- **PT:** Indicates in the first bit whether the cell contains user data or control data. The second bit indicates congestion. The third bit indicates the cell is the last in a series of cells that represent a single AAL5 frame.
- **CLP:** Indicates whether cell should be discarded if it encounters extreme congestion as it moves through the network.
- **HEC:** Calculates checksum only on the first 4 bytes of the header.The Header Error Control (HEC) field is used for both error control and synchronization, as explained subsequently.

Benefits of ATM Networks are

- It provides the dynamic bandwidth that is particularly suited for bursty traffic.
- Since all data are encoded into identical cells, data transmission is simple, uniform and predictable.
- Uniform packet size ensures that mixed traffic is handled efficiently.
- Small sized header reduces packet overload, thus ensuring effective bandwidth usage.
- ATM networks are scalable both in size and speed.

3.9 Data Link Layer Protocols

i. Point - to - Point Protocol (PPP)

- It is a communication protocol of the data link layer that is used to transmit multiprotocol data between two directly connected (point-to-point) computers.
- It is a byte - oriented protocol that is widely used in broadband communications having heavy loads and high speeds.

Services Provided by PPP

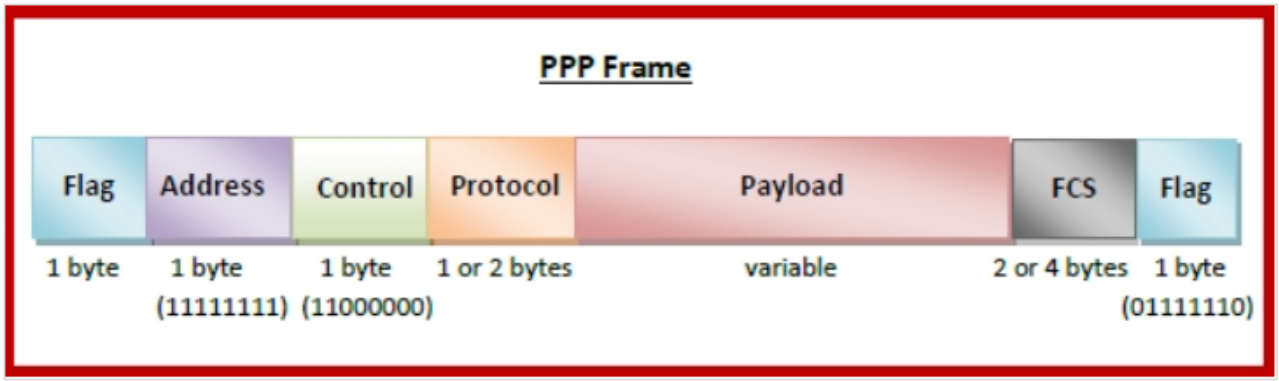
The main services provided by Point - to - Point Protocol are –

- Defining the frame format of the data to be transmitted.
- Defining the procedure of establishing link between two points and exchange of data.
- Stating the method of encapsulation of network layer data in the frame.
- Stating authentication rules of the communicating devices.
- Providing address for network communication.
- Providing connections over multiple links.
- Supporting a variety of network layer protocols by providing a range of services.

PPP Frame

PPP is a byte - oriented protocol where each field of the frame is composed of one or more bytes. The fields of a PPP frame are −

- **Flag** − 1 byte that marks the beginning and the end of the frame. The bit pattern of the flag is 01111110.
- **Address** − 1 byte which is set to 11111111 in case of broadcast.
- **Control** − 1 byte set to a constant value of 11000000.
- **Protocol** − 1 or 2 bytes that define the type of data contained in the payload field.
- **Payload** − This carries the data from the network layer. The maximum length of the payload field is 1500 bytes. However, this may be negotiated between the endpoints of communication.
- **FCS** − It is a 2 byte or 4 bytes frame check sequence for error detection. The standard code used is CRC (cyclic redundancy code)



Assignment

1. Frame Relay
2. HDLC Protocol

Thank YOU