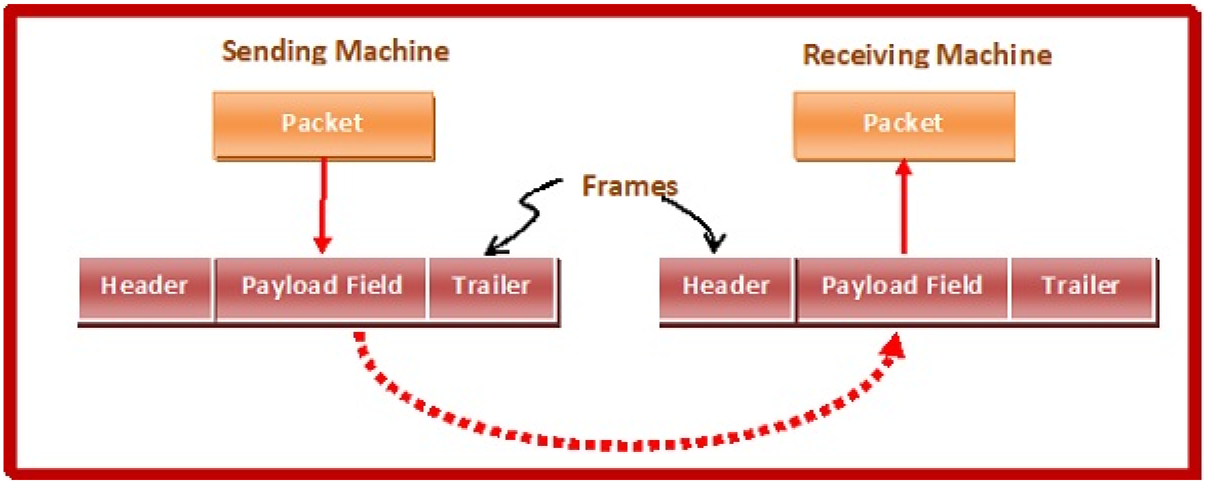
Unit-2
Topic-1

Data link layer framing methods

In the physical layer, data transmission involves synchronised transmission of bits from the source to the destination. The data link layer packs these bits into frames.

Data-link layer takes the packets from the Network Layer and encapsulates them into frames. If the frame size becomes too large, then the packet may be divided into small sized frames. Smaller sized frames makes flow control and error control more efficient.

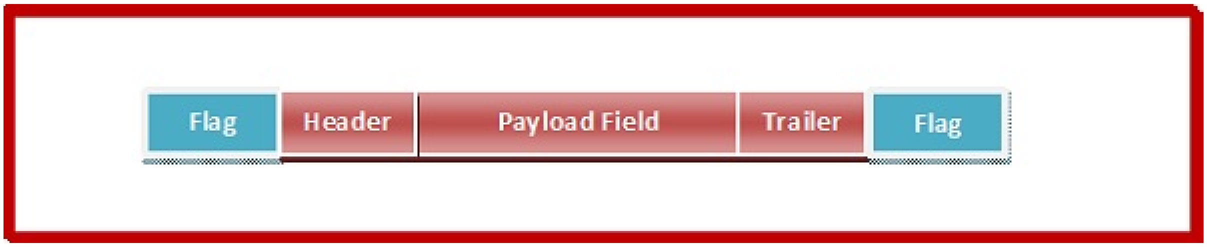
Then, it sends each frame bit-by-bit on the hardware. At receiver's end, data link layer picks up signals from hardware and assembles them into frames.



Parts of a Frame

A frame has the following parts −

- **Frame Header** − It contains the source and the destination addresses of the frame.
- **Payload field** − It contains the message to be delivered.
- **Trailer** − It contains the error detection and error correction bits.
- **Flag** − It marks the beginning and end of the frame.



Types of Framing

Framing can be of two types, fixed sized framing and variable sized framing.

Fixed-sized Framing

Here the size of the frame is fixed and so the frame length acts as delimiter of the frame. Consequently, it does not require additional boundary bits to identify the start and end of the frame.

Example − ATM cells.

Variable – Sized Framing

Here, the size of each frame to be transmitted may be different. So additional mechanisms are kept to mark the end of one frame and the beginning of the next frame.

It is used in local area networks.

Two ways to define frame delimiters in variable sized framing are −

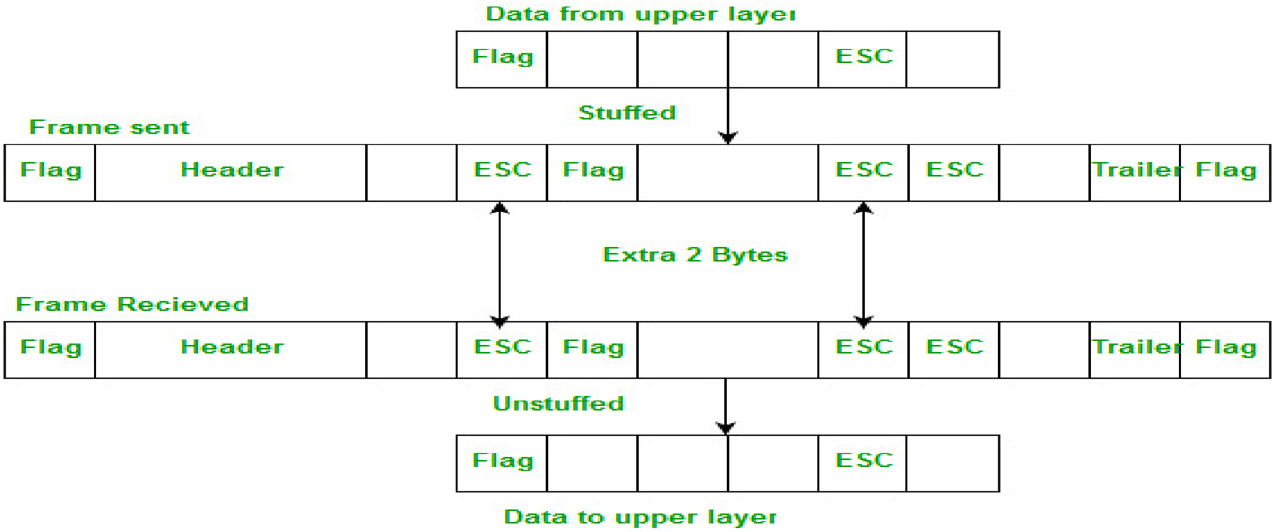
- **Length Field** − Here, a length field is used that determines the size of the frame. It is used in Ethernet (IEEE 802.3).
- **End Delimiter** − Here, a pattern is used as a delimiter to determine the size of frame. It is used in Token Rings. If the pattern occurs in the message, then two approaches are used to avoid the situation −
  - **Byte – Stuffing** − A byte is stuffed in the message to differentiate from the delimiter. This is also called character-oriented framing.
  - **Bit – Stuffing** − A pattern of bits of arbitrary length is stuffed in the message to differentiate from the delimiter. This is also called bit – oriented framing.

Byte stuffing

A byte (usually escape character(ESC)), which has a predefined bit pattern is added to the data section of the frame when there is a character with the same pattern as the flag. Whenever the receiver encounters the ESC character, it removes from the data section and treats the next character as data, not a flag.

But the problem arises when the text contains one or more escape characters followed by a flag. To solve this problem, the escape characters that are part of the text are marked by another escape character i.e., if the escape character is part of the text, an extra one is added to show that the second one is part of the text.

Example:



Note – Point-to-Point Protocol (PPP) is a byte-oriented protocol.

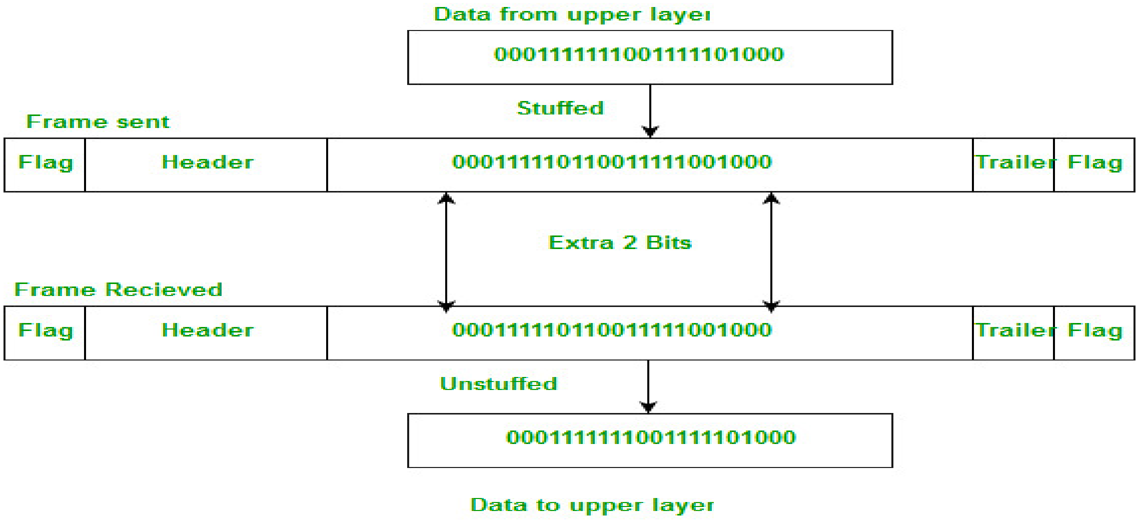
Bit stuffing

Mostly flag is a special 8-bit pattern “01111110” used to define the beginning and the end of the frame.

Problem with the flag is the same as that was in case of byte stuffing. So, in this protocol what we do is, if we encounter 0 and five consecutive 1 bits, an extra 0 is added after these bits. This extra stuffed bit is removed from the data by the receiver.

The extra bit is added after one 0 followed by five 1 bits regardless of the value of the next bit. Also, as the sender side always knows which sequence is data and which is flag it will only add this extra bit in the data sequence, not in the flag sequence.

Example:



Note – High-Level Data Link Control(HDLC) is a bit-oriented protocol.

Problems in Framing –

- **Detecting start of the frame:** When a frame is transmitted, every station must be able to detect it. Station detect frames by looking out for special sequence of bits that marks the beginning of the frame i.e. SFD (Starting Frame Delimeter).
- **How do station detect a frame:** Every station listen to link for SFD pattern through a sequential circuit. If SFD is detected, sequential circuit alerts station. Station checks destination address to accept or reject frame.
- **Detecting end of frame:** When to stop reading the frame.

https://www.youtube.com/watch?v=Re3QshTqmaM
https://www.youtube.com/watch?v=EwyPY_TSRcs
https://www.youtube.com/watch?v=dCll7RJn4c0
https://www.youtube.com/watch?v=u8xsvAGhwGs

Unit-2
Topic-2

Error Detecting Codes:

Error

A condition when the receiver's information does not match with the sender's information. During transmission, digital signals suffer from noise that can introduce errors in the binary bits travelling from sender to receiver. That means a 0 bit may change to 1 or a 1 bit may change to 0.

Error Detecting Codes (Implemented either at Data link layer or Transport Layer of OSI Model) Whenever a message is transmitted, it may get scrambled by noise or data may get corrupted. To avoid this, we use error-detecting codes which are additional data added to a given digital message to help us detect if any error has occurred during transmission of the message.

Basic approach used for error detection is the use of redundancy bits, where additional bits are added to facilitate detection of errors.

Some popular techniques for error detection are:

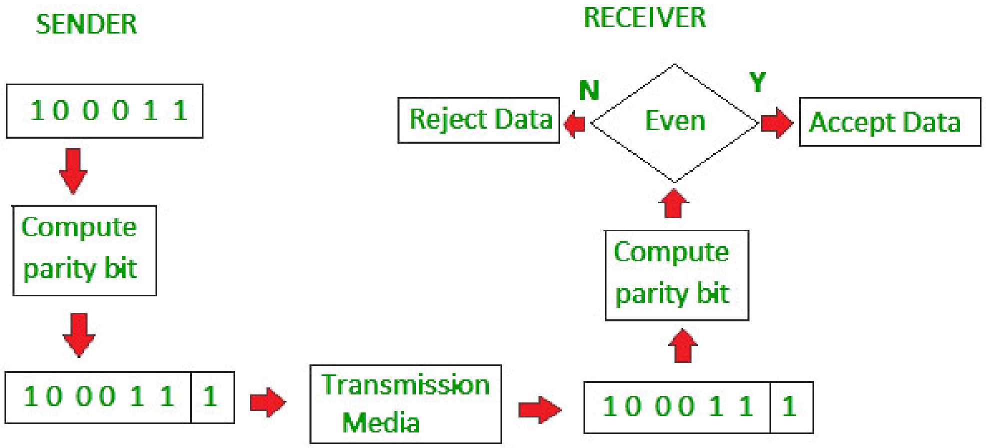
1. Simple Parity check
2. Two-dimensional Parity check
3. Checksum
4. Cyclic redundancy check

1**. Simple Parity check**

Blocks of data from the source are subjected to a check bit or parity bit generator form, where a parity of :

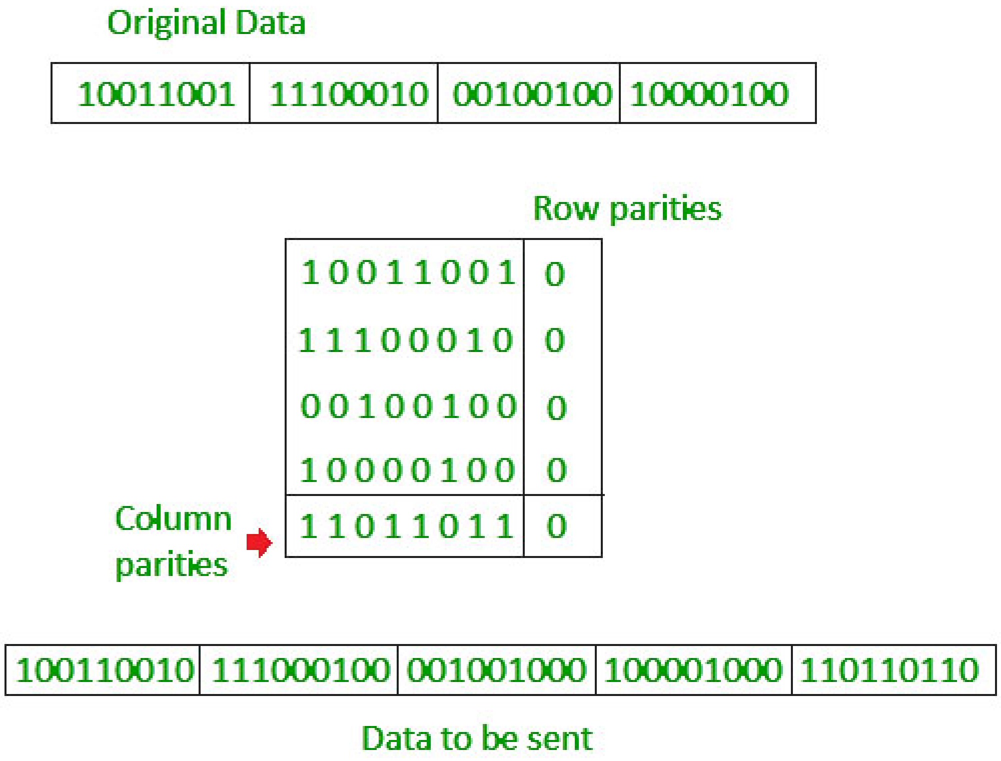
- 1 is added to the block if it contains odd number of 1's, and
- 0 is added if it contains even number of 1's

This scheme makes the total number of 1's even, that is why it is called even parity checking.



2. **Two-dimensional Parity check**

Parity check bits are calculated for each row, which is equivalent to a simple parity check bit. Parity check bits are also calculated for all columns, then both are sent along with the data. At the receiving end these are compared with the parity bits calculated on the received data.



Original Data

10011001	11100010	00100100	10000100

Row parities

10011001	0
11100010	0
00100100	0
10000100	0
11011011	0

Column parities

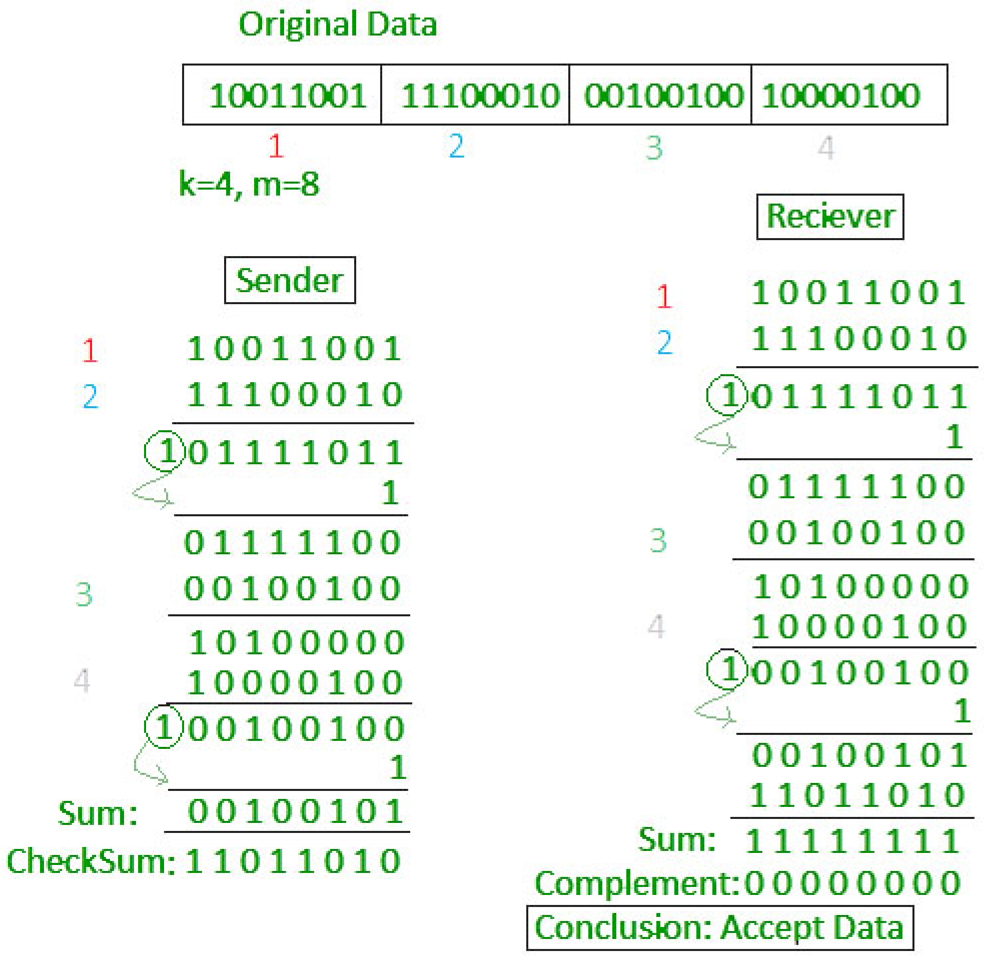
100110010	111000100	001001000	100001000	110110110

Data to be sent

3. **Checksum**

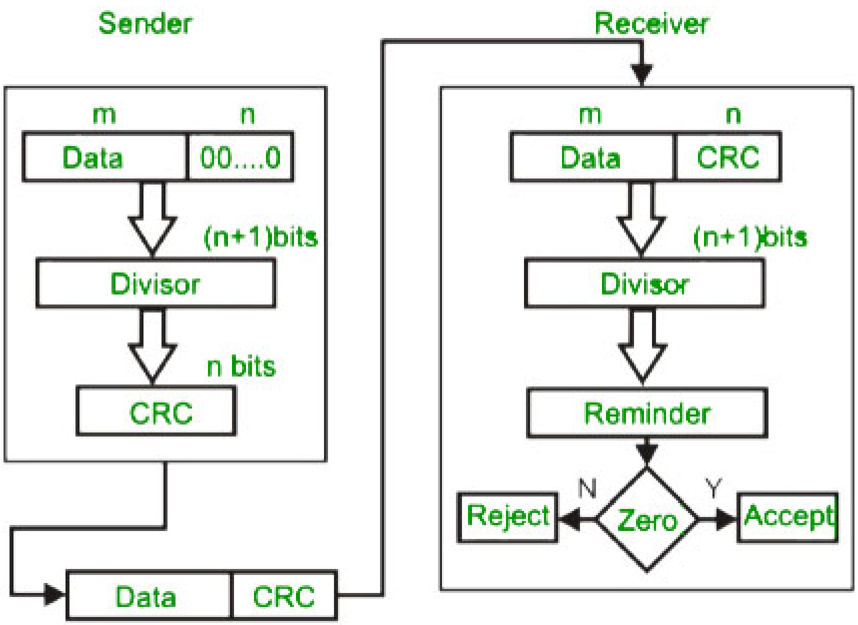
- In checksum error detection scheme, the data is divided into k segments each of m bits.
- In the sender's end the segments are added using 1's complement arithmetic to get the sum. The sum is complemented to get the checksum.

- 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.

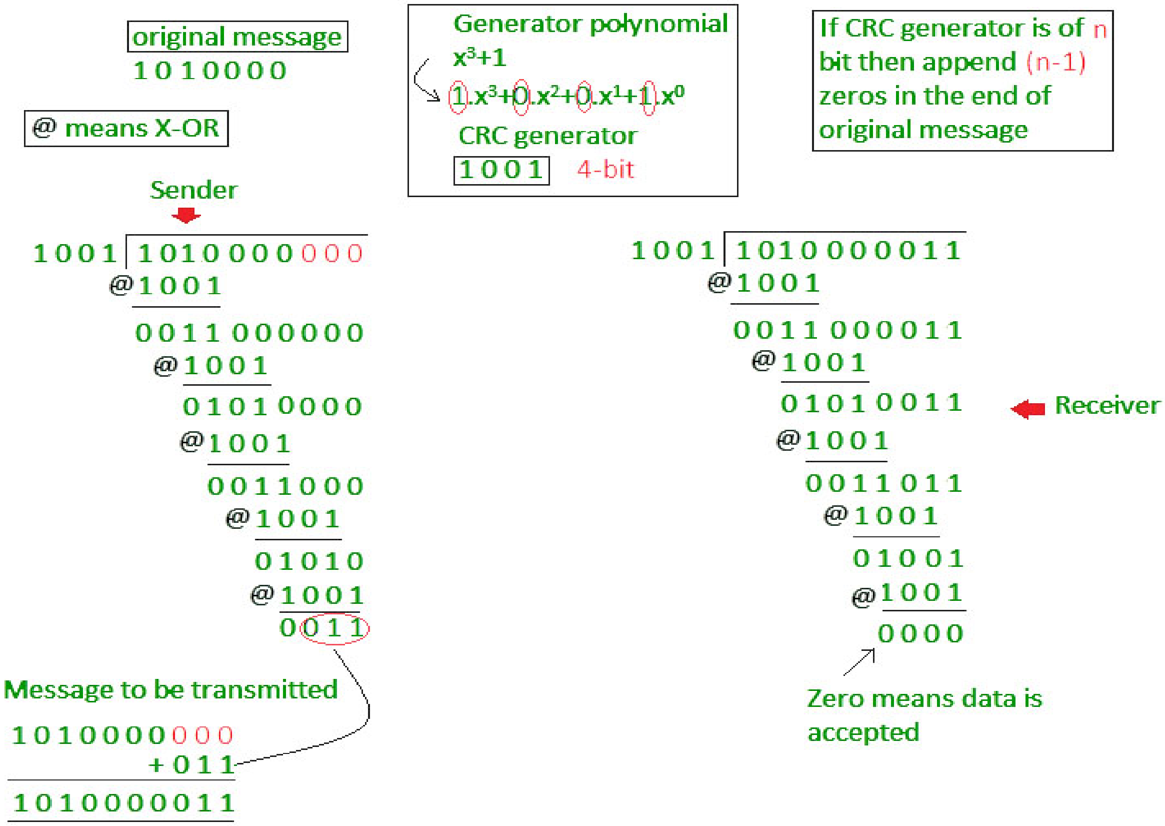


4. **Cyclic redundancy check (CRC)**

- Unlike checksum scheme, which is based on addition, CRC is based on binary division.
- In CRC, a sequence of redundant bits, called cyclic redundancy check bits, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number.
- At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted.
- A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.



Example :



original message

1010000

@ means X-OR

Generator polynomial

x^3+1

$1.x^3+0.x^2+0.x^1+1.x^0$

CRC generator

1001 4-bit

If CRC generator is of n bit then append (n-1) zeros in the end of original message

Sender

```
1001 | 1010000000
      @1001
       0011000000
        @1001
         01010000
         @1001
          0011000
           @1001
            01010
            @1001
             0011
```

Message to be transmitted

```
1010000000
      +011
1010000011
```

```
1001 | 1010000011
      @1001
       0011000011
        @1001
         01010011
         @1001
          0011011
           @1001
            01001
            @1001
             0000
```

Receiver

Zero means data is accepted

Unit-2
Topic-3

Error Correction

In the digital world, error correction can be done in two ways:

- **Backward Error Correction** When the receiver detects an error in the data received, it requests back the sender to retransmit the data unit.
- **Forward Error Correction** When the receiver detects some error in the data received, it executes error-correcting code, which helps it to auto-recover and to correct some kinds of errors.

The first one, Backward Error Correction, is simple and can only be efficiently used where retransmitting is not expensive. For example, fiber optics. But in case of wireless transmission retransmitting may cost too much. In the latter case, Forward Error Correction is used.

To correct the error in data frame, the receiver must know exactly which bit in the frame is corrupted. To locate the bit in error, redundant bits are used as parity bits for error detection.For example, we take ASCII words (7 bits data), then there could be 8 kind of information we need: first seven bits to tell us which bit is error and one more bit to tell that there is no error.

For m data bits, r redundant bits are used. r bits can provide 2r combinations of information. In m+r bit codeword, there is possibility that the r bits themselves may get corrupted. So the number of r bits used must inform about m+r bit locations plus no-error information, i.e. m+r+1.

$$2^r >= m+r+1$$

Unit-2
Topic-4

Block Coding

Conversion of Digital Data to Digital Signal involves three techniques:

1. Line Coding
2. Block Coding
3. Scrambling

Out of which Line coding is always needed, block coding and scrambling may or may not be needed.

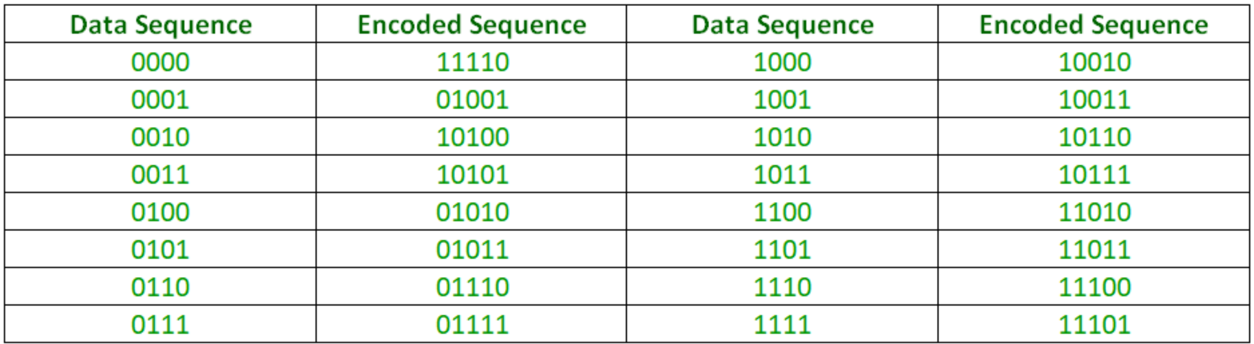
Block coding helps in error detection and re-transmission of the signal. It is normally referred to as mB/nB coding as it replaces each m-bit data group with an n-bit data group (where n>m). Thus, its adds extra bits (redundancy bits) which helps in synchronization at receiver’s and sender’s end and also providing some kind of error detecting capability.

It normally involves three steps: division, substitution, and combination. In the division step,a sequence of bits is divided into groups of m-bits. In the substitution step, we substitute an m-bit group for an n-bit group. Finally, the n-bit groups are combined together to form a stream which has more bits than the original bits.

Examples of mB/nB coding:

4B/5B (four binary/five binary) –

This coding scheme is used in combination with NRZ-I. The problem with NRZ-I was that it has a synchronization problem for long sequences of zeros. So, to overcome it we substitute the bit stream from 4-bit to 5-bit data group **before encoding it with NRZ-I**. So that it does not have a long stream of zeros. The block-coded stream does not have more than three consecutive zeros (see encoding table).



Data Sequence	Encoded Sequence	Data Sequence	Encoded Sequence
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

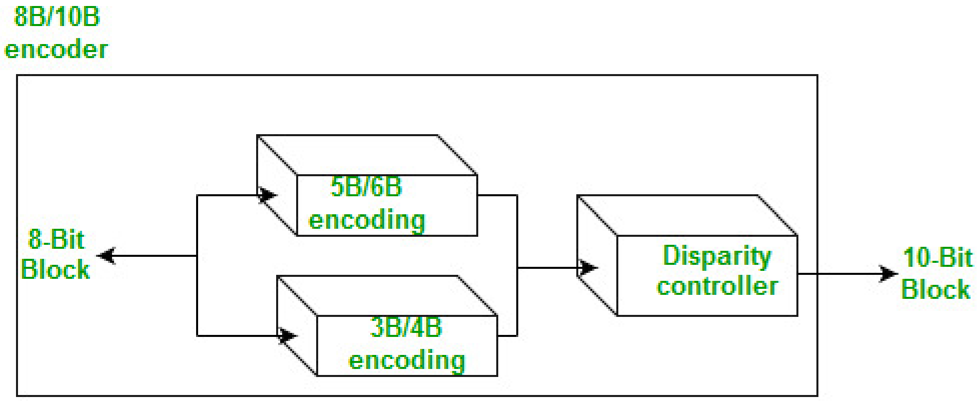
At the receiver, the NRZ-I encoded digital signal is first decoded into a stream of bits and then decoded again to remove the redundancy bits.

Drawback – Though 4B/5B encoding solves the problem of synchronization,it increases the signal rate of NRZ-L.Moreover,it does not solve the DC component problem of NRZ-L.

8B/10B (eight binary/ten binary) –

This encoding is similar to 4B/5B encoding except that a group of 8 bits of data is now substituted by a 10-bit code and it provides greater error detection capability than 4B/5B.

It is actually a combination of 5B/6B and 3B/4B encoding.The most five significant bits of a 10-bit block is fed into the 5B/6B encoder; the least 3 significant bits is fed into a 3B/4B encoder. The split is done to simplify the mapping table.



A group of 8 bits can have 2^8 different combinations while a group of 10 bits can have 2^10 different combinations. This means that there are 2^10-2^8=768 redundant groups that are not used for 8B/10B encoding and can be used for error detection and disparity check.
Thus, this technique is better than 4B/5B because of better error-checking capability and better synchronization.

Unit-2
Topic-5

Hamming Code

Hamming code is a set of error-correction codes that can be used to **detect and correct the errors** that can occur when the data is moved or stored from the sender to the receiver. It is **technique developed by R.W. Hamming for error correction**.

Redundant bits –

Redundant bits are extra binary bits that are generated and added to the information-carrying bits of data transfer to ensure that no bits were lost during the data transfer.
The number of redundant bits can be calculated using the following formula:
2^r ≥ m + r + 1
where, r = redundant bit, m = data bit
Suppose the number of data bits is 7, then the number of redundant bits can be calculated using:
= 2^4 ≥ 7 + 4 + 1
Thus, the number of redundant bits= 4

Parity bits –

A parity bit is a bit appended to a data of binary bits to ensure that the total number of 1's in the data is even or odd. Parity bits are used for error detection. There are two types of parity bits:

1. **Even parity bit:**
   In the case of even parity, for a given set of bits, the number of 1's are counted. If that count is odd, the parity bit value is set to 1, making the total count of occurrences of 1's

an even number. If the total number of 1's in a given set of bits is already even, the parity bit's value is 0.

2. **Odd Parity bit –**
   In the case of odd parity, for a given set of bits, the number of 1's are counted. If that count is even, the parity bit value is set to 1, making the total count of occurrences of 1's an odd number. If the total number of 1's in a given set of bits is already odd, the parity bit's value is 0.

General Algorithm of Hamming code –

The Hamming Code is simply the use of extra parity bits to allow the identification of an error.

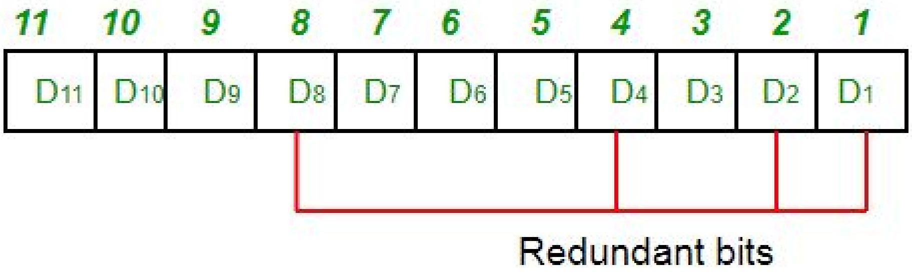
1. Write the bit positions starting from 1 in binary form (1, 10, 11, 100, etc).
2. All the bit positions that are a power of 2 are marked as parity bits (1, 2, 4, 8, etc).
3. All the other bit positions are marked as data bits.
4. Each data bit is included in a unique set of parity bits, as determined its bit position in binary form.
   a. Parity bit 1 covers all the bits positions whose binary representation includes a 1 in the least significant
   position (1, 3, 5, 7, 9, 11, etc).
   b. Parity bit 2 covers all the bits positions whose binary representation includes a 1 in the second position from
   the least significant bit (2, 3, 6, 7, 10, 11, etc).
   c. Parity bit 4 covers all the bits positions whose binary representation includes a 1 in the third position from
   the least significant bit (4–7, 12–15, 20–23, etc).
   d. Parity bit 8 covers all the bits positions whose binary representation includes a 1 in the fourth position from
   the least significant bit bits (8–15, 24–31, 40–47, etc).
   e. In general, each parity bit covers all bits where the bitwise AND of the parity position and the bit position is
   non-zero.
5. Since we check for even parity set a parity bit to 1 if the total number of ones in the positions it checks is
   odd.
6. Set a parity bit to 0 if the total number of ones in the positions it checks is even.

Determining the position of redundant bits –

These redundancy bits are placed at the positions which correspond to the power of 2.

As in the above example:

1. The number of data bits = 7
2. The number of redundant bits = 4
3. The total number of bits = 11
4. The redundant bits are placed at positions corresponding to power of 2- 1, 2, 4, and 8



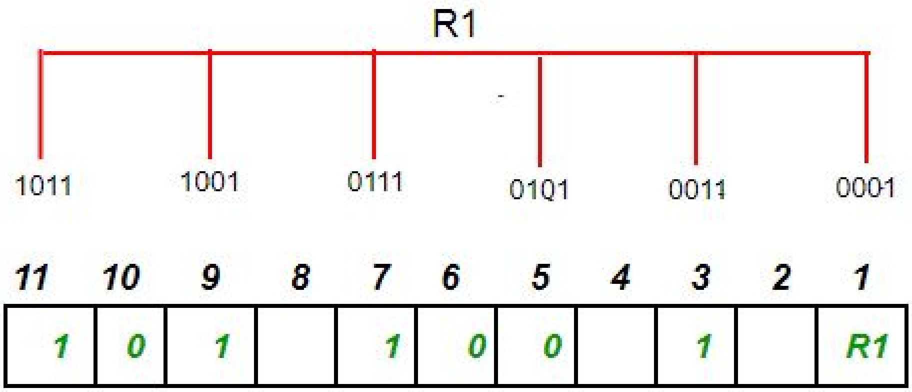
Suppose the data to be transmitted is 1011001, the bits will be placed as follows:

11	10	9	8	7	6	5	4	3	2	1
1	0	1	R8	1	0	0	R4	1	R2	R1

Determining the Parity bits –

1. R1 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the least significant position.

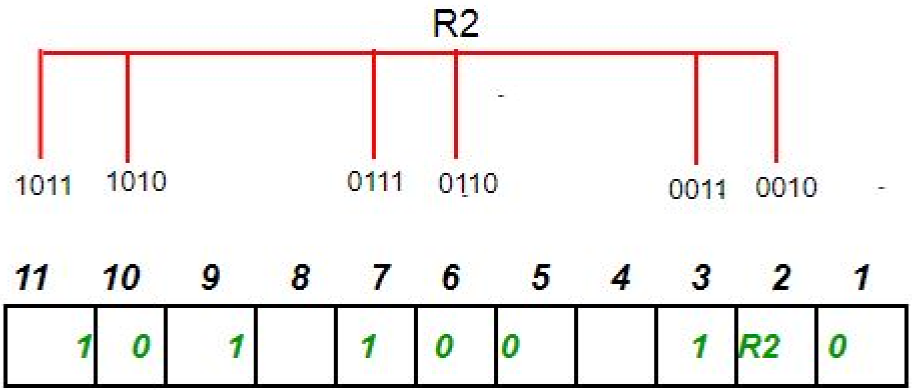
R1: bits 1, 3, 5, 7, 9, 11



To find the redundant bit R1, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R1 is an even number the value of R1 (parity bit's value) = 0

2. R2 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the second position from the least significant bit.

R2: bits 2,3,6,7,10,11

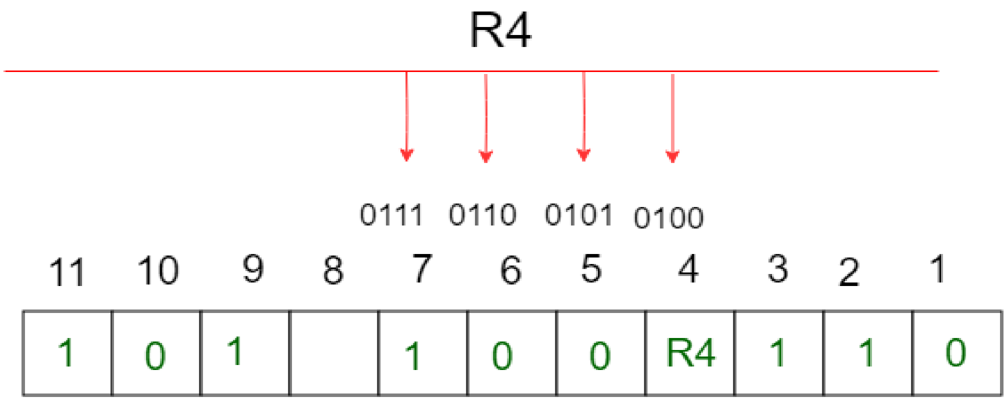


11	10	9	8	7	6	5	4	3	2	1
1	0	1		1	0	0		1	R2	0

To find the redundant bit R2, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R2 is odd the value of R2(parity bit's value)=1

3. R4 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the third position from the least significant bit.

R4: bits 4, 5, 6, 7

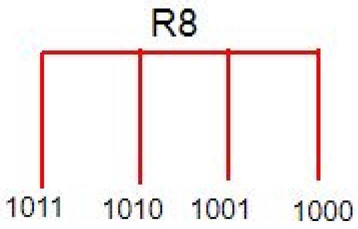


11	10	9	8	7	6	5	4	3	2	1
1	0	1		1	0	0	R4	1	1	0

To find the redundant bit R4, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R4 is odd the value of R4(parity bit's value) = 1

4. R8 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the fourth position from the least significant bit.

R8: bit 8,9,10,11



11	10	9	8	7	6	5	4	3	2	1
1	0	1	R8	1	0	0	1	1	1	0

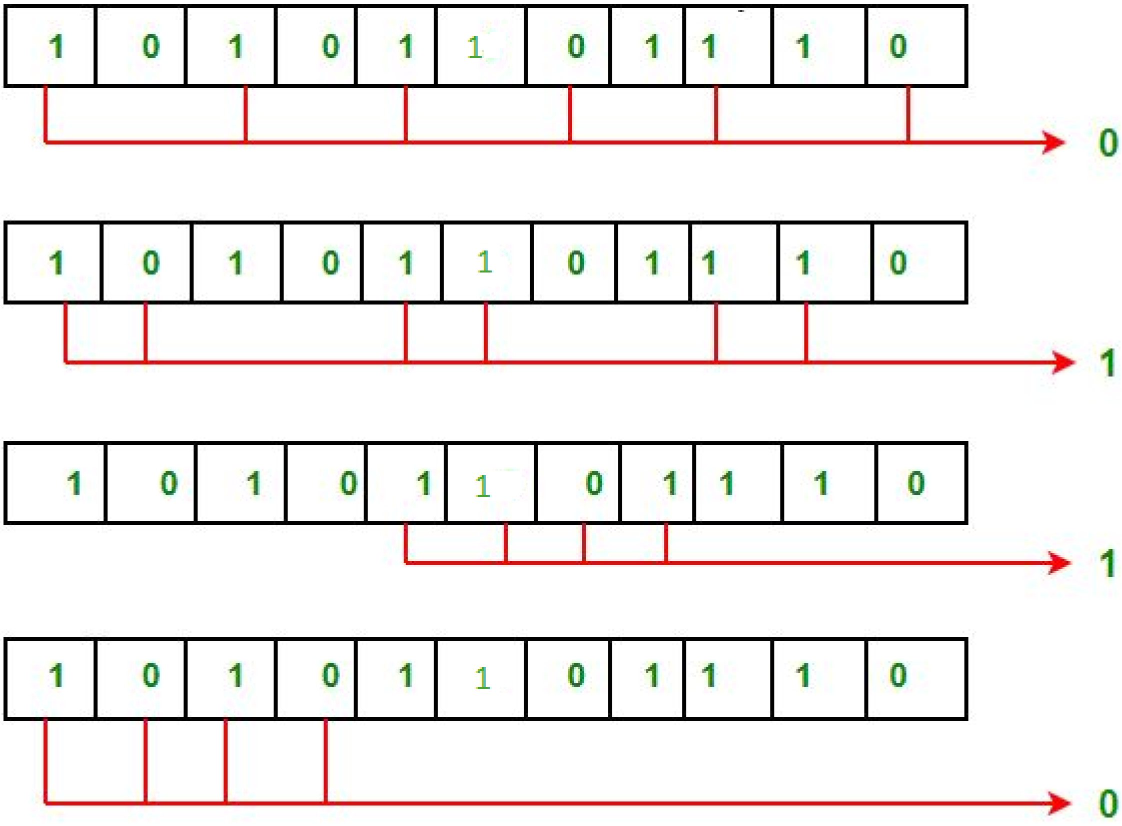
To find the redundant bit R8, we check for even parity. Since the total number of 1's in all the bit positions corresponding to R8 is an even number the value of R8(parity bit's value)=0.

Thus, the data transferred is:

11	10	9	8	7	6	5	4	3	2	1
1	0	1	0	1	0	0	1	1	1	0

Error detection and correction –

Suppose in the above example the 6th bit is changed from 0 to 1 during data transmission, then it gives new parity values in the binary number:



‘

The bits give the binary number as 0110 whose decimal representation is 6. Thus, the bit 6 contains an error. To correct the error the 6th bit is changed from 1 to 0.
https://www.youtube.com/watch?v=1A_NcXxdoCc

Unit-2
Topic-6

Flow Control

When a data frame (Layer-2 data) is sent from one host to another over a single medium, it is required that the sender and receiver should work at the same speed. That is, sender sends at a speed on which the receiver can process and accept the data. What if the speed (hardware/software) of the sender or receiver differs? If sender is sending too fast the receiver may be overloaded, (swamped) and data may be lost.
Two types of mechanisms can be deployed to control the flow:

Stop and Wait
Sliding Window

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 helps 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.

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.

There are three types of techniques available which Data-link layer may deploy to control the errors by Automatic Repeat Requests (ARQ):

Stop-and-wait ARQ

Go-Back-N ARQ

Selective Repeat ARQ

https://www.youtube.com/watch?v=ReQiSK8W3Ag

https://www.youtube.com/watch?v=WYIPXZLmaDM

Unit-2

Topic-7

LAN Technologies | ETHERNET

Local Area Network (LAN) is a data communication network connecting various terminals or computers within a building or limited geographical area. The connection among the devices could be wired or wireless. Ethernet, Token Ring and Wireless LAN using IEEE 802.11 are examples of standard LAN technologies.

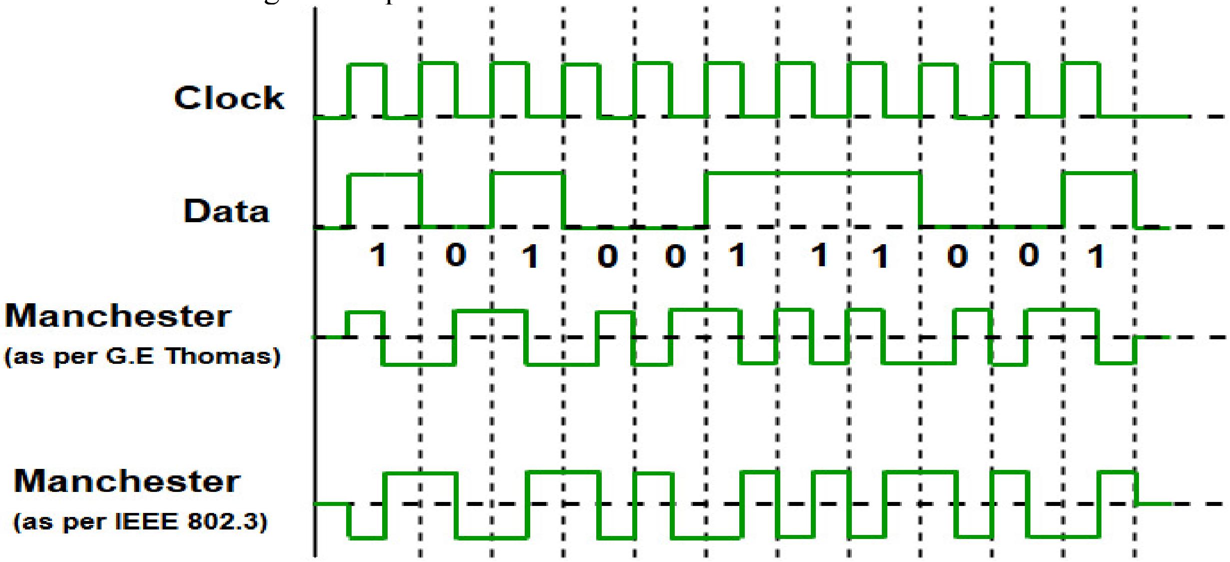
LAN has following topologies:

- Star Topology
- Bus Topology
- Ring Toplology
- Mesh Topology
- Hybrid Topology
- Tree Topology

Ethernet :-

Ethernet is most widely used LAN Technology, which is defined under IEEE standards 802.3. The reason behind its wide usability is Ethernet is easy to understand, implement, maintain and allows low-cost network implementation. Also, Ethernet offers flexibility in terms of topologies which are allowed.Ethernet generally uses Bus Topology. Ethernet operates in two layers of the OSI model, Physical Layer, and Data Link Layer. For Ethernet, the protocol data unit is Frame since we mainly deal with DLL. In order to handle collision, the Access control mechanism used in Ethernet is CSMA/CD.

Manchester Encoding Technique is used in Ethernet.



Since we are talking about IEEE 802.3 standard Ethernet therefore, 0 is expressed by a high-to-low transition, a 1 by the low-to-high transition. In both Manchester Encoding and Differential Manchester, Encoding Baud rate is double of bit rate.

Baud rate = 2* Bit rate

Ethernet LANs consist of network nodes and interconnecting media or link. The network nodes can be of two types:

Data Terminal Equipment (DTE):- Generally, DTEs are the end devices that convert the user information into signals or reconvert the received signals. DTEs devices are: personal computers, workstations, file servers or print servers also referred to as end stations. These devices are either the source or the destination of data frames. The data terminal equipment may be a single piece of equipment or multiple pieces of equipment that are interconnected and perform all the required functions to allow the user to communicate. A user can interact to DTE or DTE may be a user.

Data Communication Equipment (DCE):- DCEs are the intermediate network devices that receive and forward frames across the network. They may be either standalone devices such as repeaters, network switches, routers or maybe communications interface units such as interface cards and modems. The DCE performs functions such as signal conversion, coding and may be a part of the DTE or intermediate equipment.

Currently, these data rates are defined for operation over optical fibers and twisted-pair cables:

i) Fast Ethernet

Fast Ethernet refers to an Ethernet network that can transfer data at a rate of 100 Mbit/s.

ii) Gigabit Ethernet

Gigabit Ethernet delivers a data rate of 1,000 Mbit/s (1 Gbit/s).

iii) 10 Gigabit Ethernet

10 Gigabit Ethernet is the recent generation and delivers a data rate of 10 Gbit/s (10,000 Mbit/s). It is generally used for backbones in high-end applications requiring high data rates.

https://www.youtube.com/watch?v=HLziLmaYsO0 https://www.youtube.com/watch?v=5u52wbqBgEY

Unit-2
Topic-8

Wireless LAN

Wireless LAN stands for **Wireless Local Area Network**. It is also called LAWN (**Local Area Wireless Network**). WLAN is one in which a mobile user can connect to a Local Area Network (LAN) through a wireless connection.

The IEEE 802.11 group of standards defines the technologies for wireless LANs. For path sharing, 802.11 standard uses the Ethernet protocol and CSMA/CA (carrier sense multiple access with collision avoidance). It also uses an encryption method i.e. wired equivalent privacy algorithm.

Wireless LANs provide high speed data communication in small areas such as building or an office. WLANs allow users to move around in a confined area while they are still connected to the network.

In some instance wireless LAN technology is used to save costs and avoid laying cable, while in other cases, it is the only option for providing high-speed internet access to the public. Whatever the reason, wireless solutions are popping up everywhere.

Examples of WLANs that are available today are NCR's waveLAN and Motorola's ALTAIR.

Types of WLANS

WLANs, as standardized by IEEE 802.11, operates in two basic modes, infrastructure, and ad hoc mode.

- **Infrastructure Mode** − Mobile devices or clients connect to an access point (AP) that in turn connects via a bridge to the LAN or Internet. The client transmits frames to other clients via the AP.
- **Ad Hoc Mode** − Clients transmit frames directly to each other in a peer-to-peer fashion.

Advantages of WLANs

- They provide clutter-free homes, offices and other networked places.
- The LANs are scalable in nature, i.e. devices may be added or removed from the network at greater ease than wired LANs.
- The system is portable within the network coverage. Access to the network is not bounded by the length of the cables.
- Installation and setup are much easier than wired counterparts.
- The equipment and setup costs are reduced.

Disadvantages of WLANs

- Since radio waves are used for communications, the signals are noisier with more interference from nearby systems.
- Greater care is needed for encrypting information. Also, they are more prone to errors. So, they require greater bandwidth than the wired LANs.
- WLANs are slower than wired LANs.

https://www.youtube.com/watch?v=m_qTs_FQ_jU

Unit-2
Topic-9

Network Devices (Hub, Repeater, Bridge, Switch, Router, Gateways and Brouter)

1. Repeater – A repeater operates at the physical layer. Its job is to regenerate the signal over the same network before the signal becomes too weak or corrupted so as to extend the length to which the signal can be transmitted over the same network. An important point to be noted about repeaters is that they do not amplify the signal. When the signal becomes weak, they copy the signal bit by bit and regenerate it at the original strength. It is a 2 port device.

2. Hub – A hub is basically a multiport repeater. A hub connects multiple wires coming from different branches, for example, the connector in star topology which connects different stations. Hubs cannot filter data, so data packets are sent to all connected devices. In other words, collision domain of all hosts connected through Hub remains one. Also, they do not have intelligence to find out best path for data packets which leads to inefficiencies and wastage.

Types of Hub

- **Active Hub:-** These are the hubs which have their own power supply and can clean, boost and relay the signal along with the network. It serves both as a repeater as well as wiring centre. These are used to extend the maximum distance between nodes.
- **Passive Hub :-** These are the hubs which collect wiring from nodes and power supply from active hub. These hubs relay signals onto the network without cleaning and boosting them and can't be used to extend the distance between nodes.

3. Bridge – A bridge operates at data link layer. A bridge is a repeater, with add on the functionality of filtering content by reading the MAC addresses of source and destination. It is also used for interconnecting two LANs working on the same protocol. It has a single input and single output port, thus making it a 2 port device.

Types of Bridges

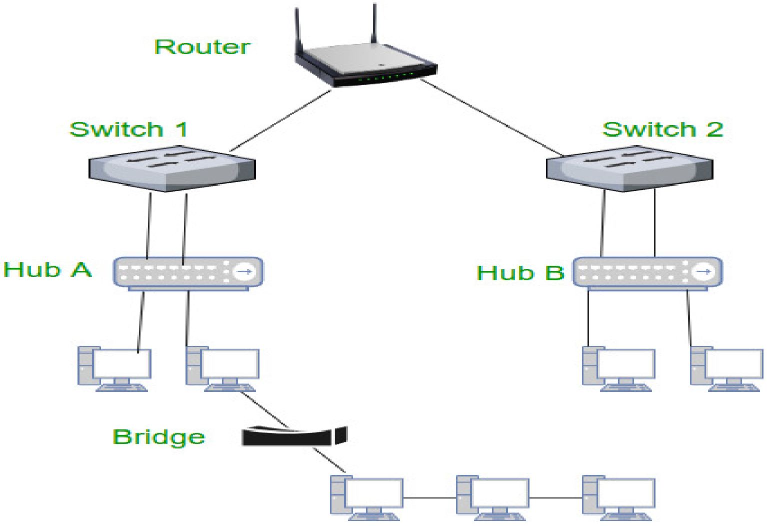
- **Transparent Bridges:-** These are the bridge in which the stations are completely unaware of the
  bridge's existence i.e. whether or not a bridge is added or deleted from the network, reconfiguration of

the stations is unnecessary. These bridges make use of two processes i.e. bridge forwarding and bridge learning.

- **Source Routing Bridges:-** In these bridges, routing operation is performed by source station and the frame specifies which route to follow. The hot can discover frame by sending a special frame called discovery frame, which spreads through the entire network using all possible paths to destination.

4. Switch – A switch is a multiport bridge with a buffer and a design that can boost its efficiency(a large number of ports imply less traffic) and performance. A switch is a data link layer device. The switch can perform error checking before forwarding data, that makes it very efficient as it does not forward packets that have errors and forward good packets selectively to correct port only. In other words, switch divides collision domain of hosts, but broadcast domain remains same.

5. Routers – A router is a device like a switch that routes data packets based on their IP addresses. Router is mainly a Network Layer device. Routers normally connect LANs and WANs together and have a dynamically updating routing table based on which they make decisions on routing the data packets. Router divide broadcast domains of hosts connected through it.



6. Gateway – A gateway, as the name suggests, is a passage to connect two networks together that may work upon different networking models. They basically work as the messenger agents that take data from one system, interpret it, and transfer it to another system. Gateways are also called protocol converters and can operate at any network layer. Gateways are generally more complex than switch or router.

7. Brouter – It is also known as bridging router is a device which combines features of both bridge and router. It can work either at data link layer or at network layer. Working as router, it is capable of routing packets across networks and working as bridge, it is capable of filtering local area network traffic.

https://www.youtube.com/watch?v=eMamgWllRFY

https://www.youtube.com/watch?v=1z0ULvg_pW8