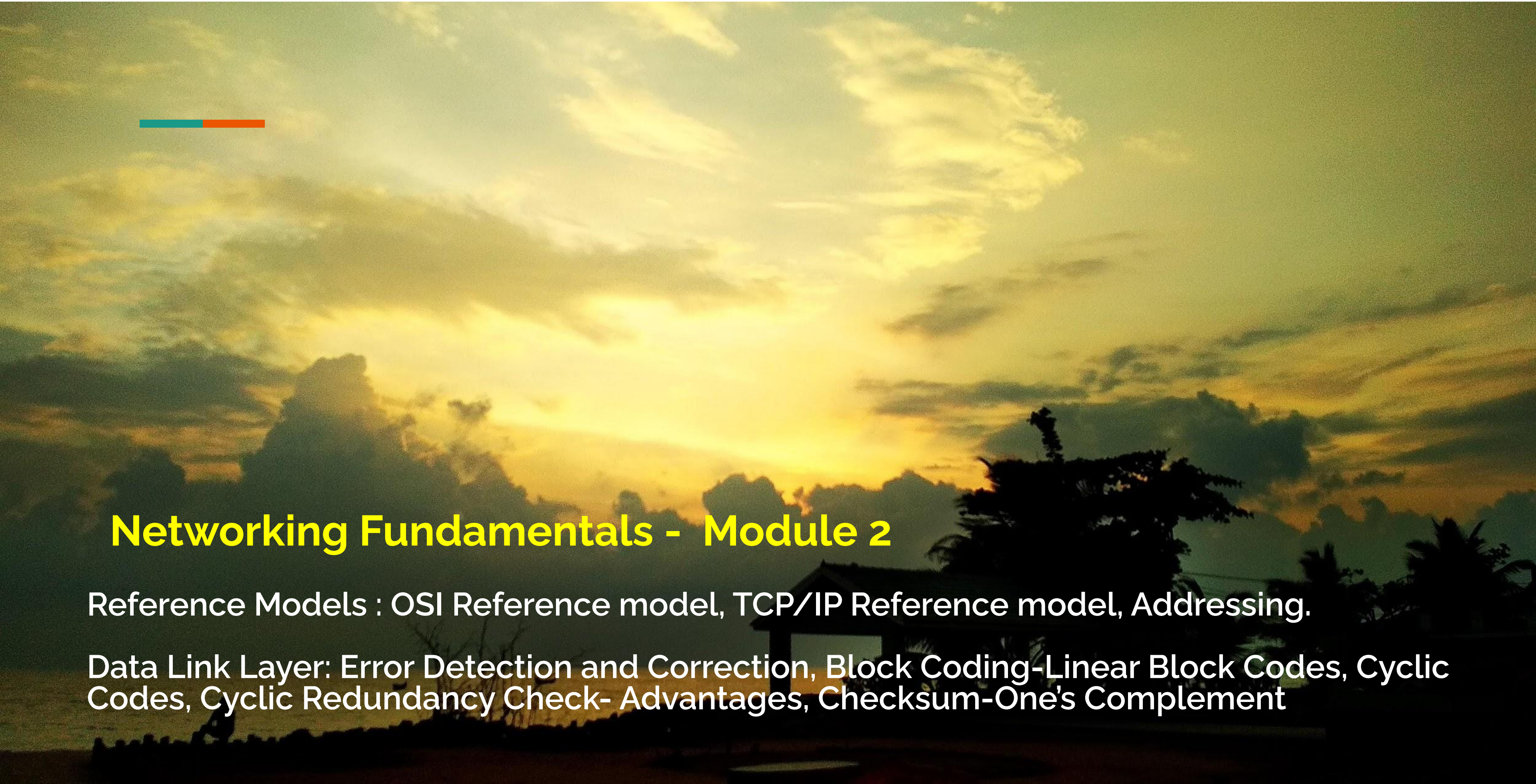


Networking Fundamentals - Module 2

Reference Models : OSI Reference model, TCP/IP Reference model, Addressing.

Data Link Layer: Error Detection and Correction, Block Coding-Linear Block Codes, Cyclic Codes, Cyclic Redundancy Check- Advantages, Checksum-One's Complement

Network Reference Model

During 1970s, computer networks were built using different types of hardware & software implementations. Since the computers were of different architecture, it became difficult for networks to communicate with each other. To solve this problem, a standardised model or a reference model wa required.

The International Organization for Standardization (ISO) [an organization which publishes the international standards], researched various schemes, and developed a reference model known as **ISO-OSI Model** in 1984. This was developed to ensure worldwide data communication.

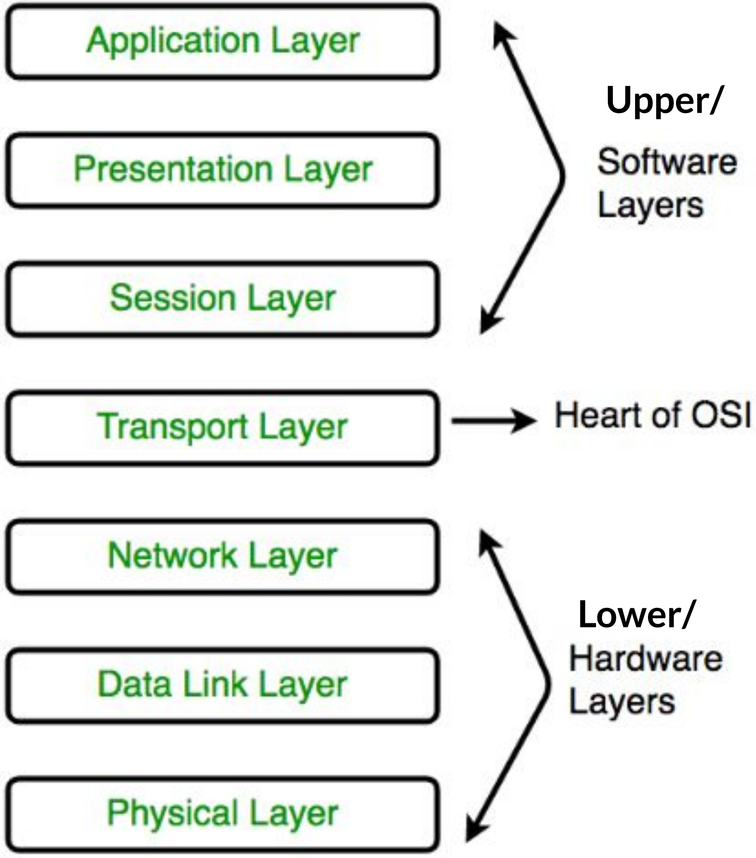
OSI (Open System Interconnection) is a reference/ theoretical model that describes how an ideal computer network should be! How devices should communicate; ie how information from a software application in one computer moves through a physical medium to the software application in another computer. All vendors started to design their devices according to the rules specified in this model and thus world wide data communication became possible.

ISO International Organization for Standardization

OSI Model

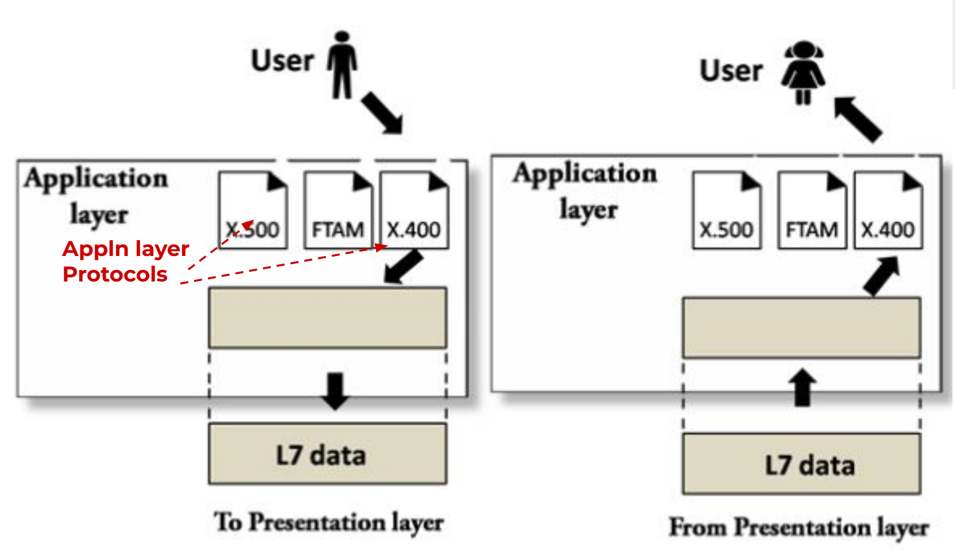
- This model describes how data is transmitted from one system to another system irrespective of distance and location.
- OSI model consists of a seven-layer architecture. It divides the whole task into seven smaller and manageable tasks. Each layer is independent & self-contained; ie a particular task is assigned to each layer.
- The concept that the OSI Model should be a seven-layer structure, was proposed by Charles Bachman at Honeywell Information Systems
- All seven layers work together to transmit data from one system to another system.
- Using this model, troubleshooting has become easier as the error can be detected at different levels
- The OSI model is never fully implemented, even though it explains all the aspects of network communication.

OSI Model



- The seven layers of OSI Model are Physical Layer, Data-Link Layer, Network Layer, Transport Layer, Session Layer, Presentation Layer & Application Layer.
- Each layer is independent and has its own protocols by which it processes the data. Each layer communicates with the layer above or below it.
- The OSI model is divided into two : upper layers and lower layers.
- The upper layer of the OSI model mainly deals with the application related issues, and they are implemented only in the software. The application layer is closest to the end user. Both the end user and the application layer interact with the software applications.
- The lower layer of the OSI model deals with the data transport issues. The data link layer and the physical layer are implemented in hardware and software. The physical layer is the lowest layer of the OSI model and is closest to the physical medium. The physical layer is mainly responsible for placing the information on the physical medium.

Layer 7 - Application Layer



The application layer is not an application. It is a process or service that works behind an application used for networking.

For eg:, when a browser requires a web page, it sends a request for the page to the server using HTTP & the server sends the requested page back. Similarly if you use Outlook to check your mails, the SMTP / POP protocol works behind it. These form the application layer

- It is the top most layer of OSI Model; also called Desktop Layer
- It is layer where actual communication is initiated; ie this layer interacts with the user
- The application layer is used by end-user softwares such as web browsers and email clients. It provides protocols that allow software to send and receive information and present meaningful data to users.
- A few examples of application layer protocols are the
  - Hypertext Transfer Protocol (**HTTP**)
  - File Transfer Protocol (**FTP**)
  - Post Office Protocol (**POP**)
  - Simple Mail Transfer Protocol (**SMTP**)
  - Domain Name System (**DNS**)
  - Teletype NETwork (**TELNET**)
- The Application layer takes the help of layers below it to transfer data to and from the remote host.

Layer 6 - Presentation Layer

- This layer is also known as the Translation layer, since it serves as a data translator for the network.
- The data received from the Application Layer is extracted and converted to the format required to transmit over the network. It encodes the messages from the user dependent format to the common format and vice versa.
- The main responsibility of this layer is to define how two devices should encode, encrypt, and compress data so that it is received correctly on the other end.

Functions of Presentation Layer :

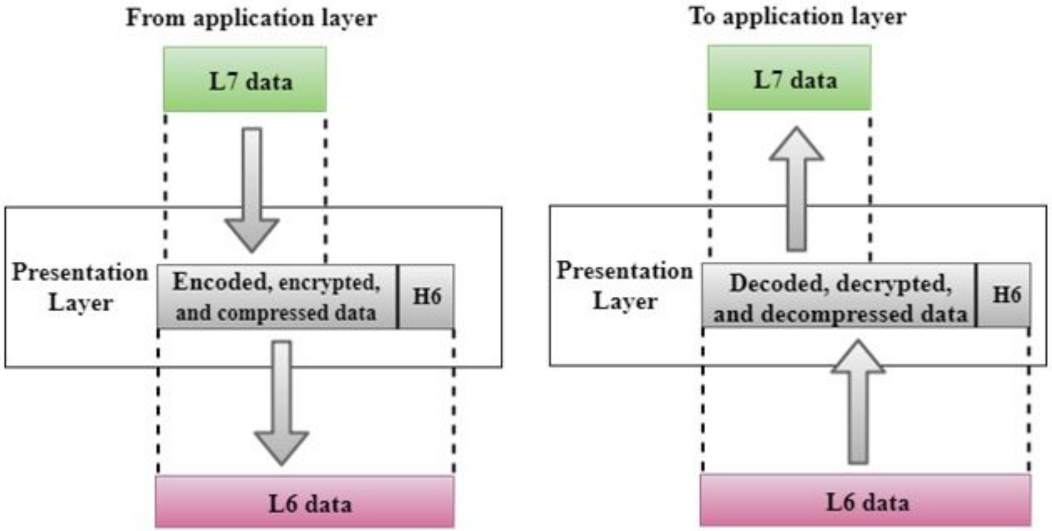
1. **Translation:** As the translator, the presentation layer converts the data received from the application layer of the transmitting node into a standardized format.

2. **Encryption:** Encryption enhances the security of data. At the sender side data is encrypted and at the receiver side, data is decrypted. Data Encryption translates the data into another format, which can be decrypted only by the receiver. The encrypted data is known as **cipher text** and the decrypted data is known as **plain text.** A key is required for encrypting and decrypting data. SSL (Secure Socket Layer) protocol is used in the Presentation layer for encryption & decryption.

Layer 6 - Presentation Layer

3. **Compression:** Data Compression reduces the bandwidth of the data to be transmitted. ie Compression reduces the no. of bits to be transmitted. Since the file size is reduced, it can be received at the destination in very less time. Thus, Data transmission becomes faster. It is very helpful in real-time video or audio streaming.

After translation, encryption & compression, the Presentation layer transfers data to the Session Layer.



Layer 5 - Session Layer

- The Session Layer is the 5th layer in the Open System Interconnection (OSI) model. The session layer creates communication channels called sessions, between devices. ie It allows users on different machines to establish active communication sessions between them.
- This layer is responsible for establishing, managing, synchronizing and terminating sessions between end-user application processes.
- The session layer can also set checkpoints during a data transfer. If the session is interrupted, devices can resume data transfer from the last checkpoint.
- Session layers functions are done using certain APIs(Application Program Interface). NETBIOS (NETwork Basic Input Output System) is an API which allows applications on different computers to communicate with each other

Functions of Session Layer

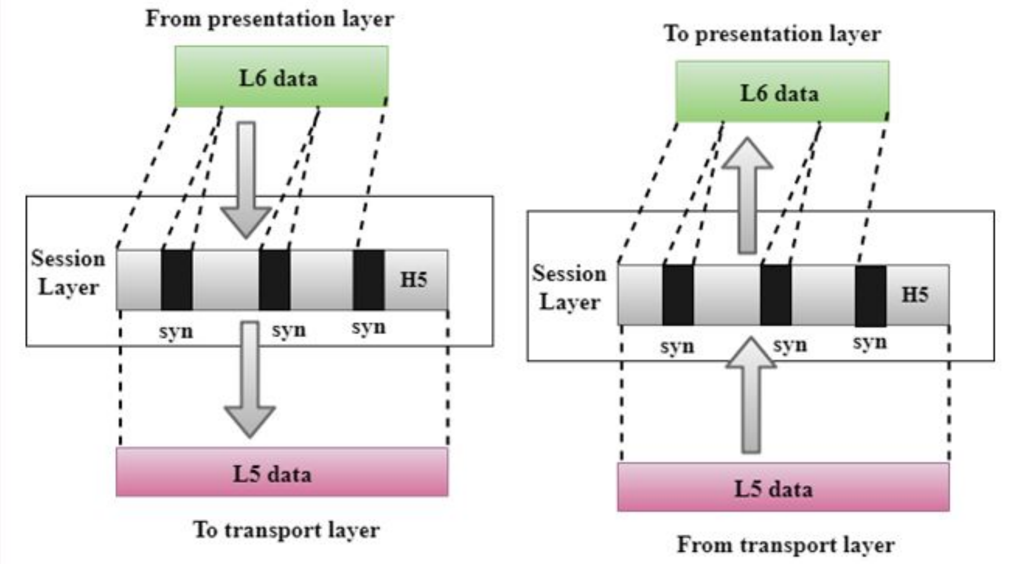
1. **Dialog Control :** Session Layer works as a dialog controller through which it allows systems to communicate in either half-duplex mode or full duplex mode of communication. It helps you to establish, maintain and terminate the connections between end-user application processes.

[*When two devices are communicating, if only one device can communicate at a time, it is half duplex. If both devices can communicate at the same time, then it is full duplex.*]

Functions of Session Layer

2. **Authentication and Authorization :** Authentication and authorization are services that are provided by the session layer. Authentication is the process of verifying who you are. Logging on to a server with a username and password is authentication. The main purpose of authentication is security. After authentication is successful, authorization is checked. Authorization is the process used by the server to determine if you have permissions to access the resources of the server. i.e. verifying what you are able to do.

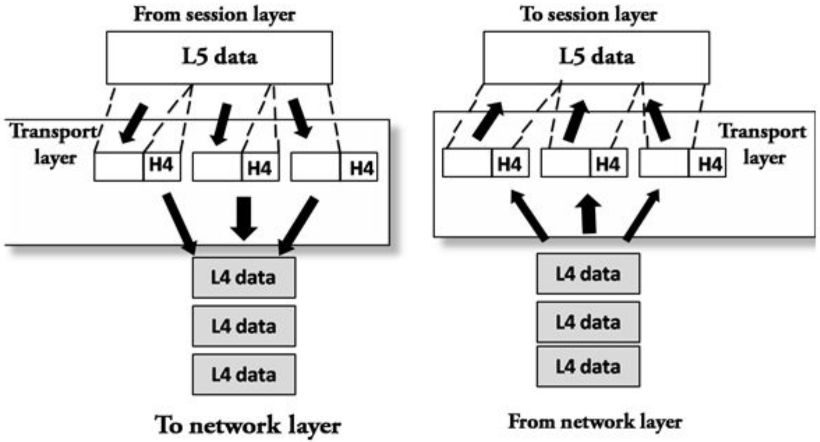
All the functions of Session, Presentation & Application layer are performed by Network applications such as your Web Browsers

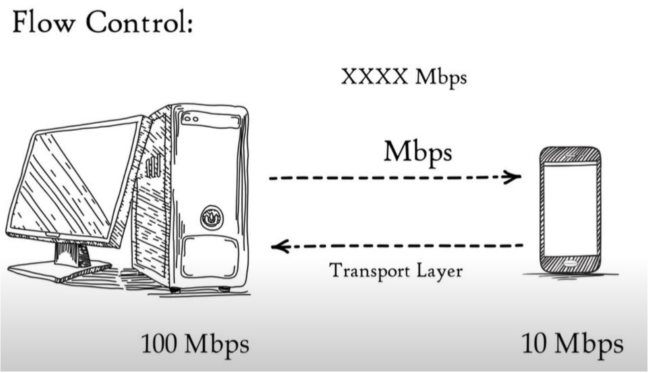


3. **Synchronization :** Once communication starts, Session Layer protocols help to ensure that communications can be restarted or resumed if it gets interrupted or contains errors. This process is called synchronization. To achieve synchronization, the Session layer adds some checkpoints [synchronization points] when transmitting the data in a sequence. This ensures that data up to a particular checkpoint is received. In case if the session is interrupted due to connection failure, devices can resume data transfer from the last checkpoint.

Layer 4 - Transport Layer

- The Transport Layer is called the **Heart of OSI model .**
- The transport layer accepts data from the session layer. It divides the message into smaller units known as **segments** to transfer it over the network. A sequence number along with the port address is added to the header of each segment.
- The main responsibility of the transport layer is to deliver a message from a specific process of one computer to a specific process in another computer.
- At the receiving end, this layer is responsible for reassembling the segments, and turning it back into data that can be used by the receiver's session layer.
- The transport layer carries out **flow control**- ie sending data at a rate that matches the connection speed of the receiving device, and **error control** - checking if data is received correctly, if not received correctly, it is requested to resend it again.
- It identifies errors like damaged packets, lost packets, and duplication of packets, and provides adequate error-correction techniques.





Functions of Transport Layer :

1. **Segmentation and reassembly:** When the transport layer receives the message from the upper layer, it divides the message into multiple segments, and each segment is assigned with a sequence number that uniquely identifies each segment. When the message reaches its destination, the transport layer of the receiver, reassembles the message based on their sequence numbers.
2. **Service-point addressing:** Computers often transmit multiple segments from different applications across a network. Due to this reason, the transport layer adds a specific type of address known as the **service-point address or port address** to the header of each segment. The transport layer at the receiving end, receives data segments from its network layer and delivers the data to the appropriate processes running on the receiver's system.

3. **Flow control:** The transport layer is also responsible for flow control. The sending device may transmit data at a faster rate than the processing capacity of the receiving device. When the receiver is unable to process the incoming data, it gets overwhelmed with data. To avoid such situations, the transport layer manages the flow control to an acceptable rate.

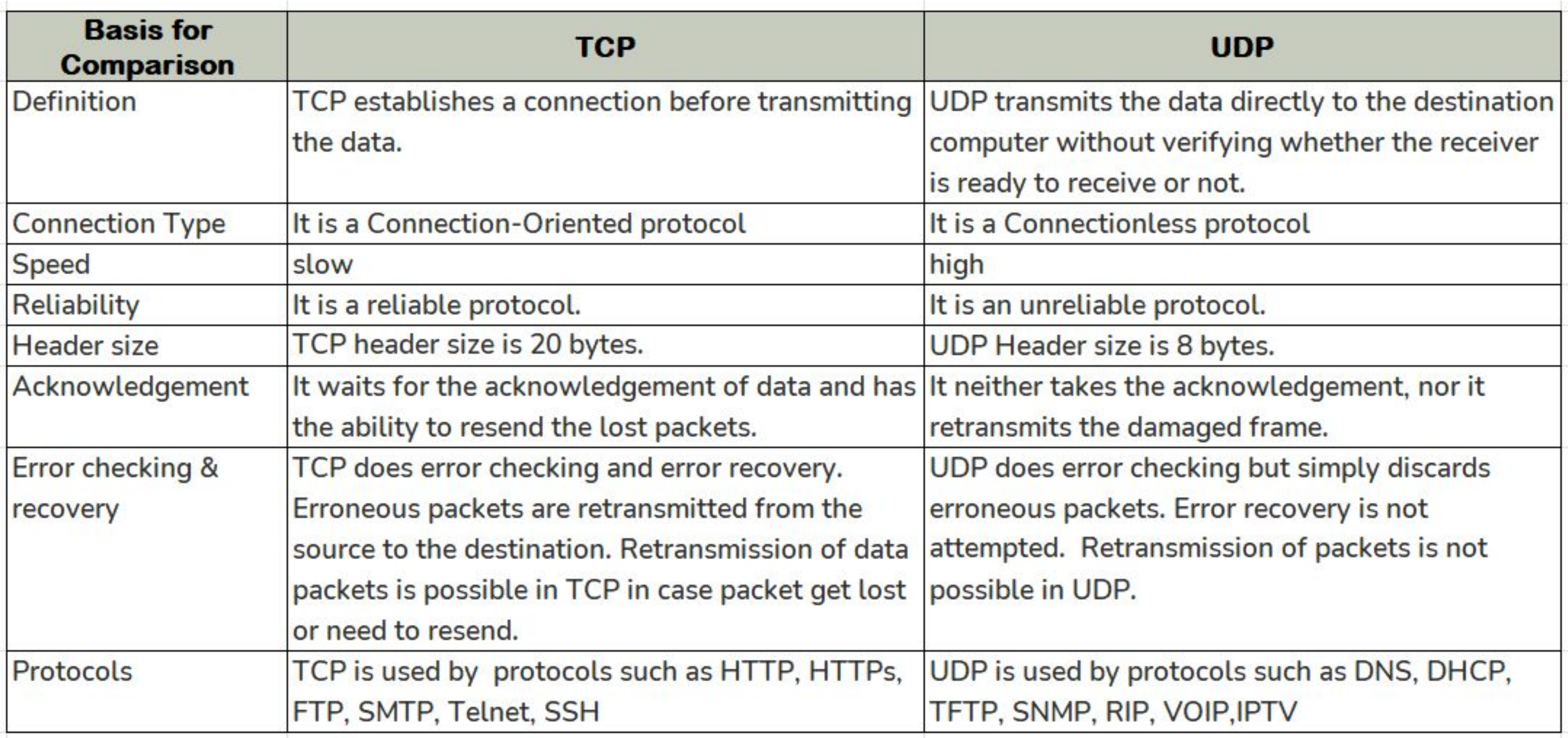
4. **Error control:** The transport layer is also responsible for Error control. Error control is performed end-to-end rather than across the single link. The sender transport layer ensures that messages reach the destination without any error such as data loss, damage , duplication etc. Error correction is achieved through retransmission of the packet.

5. **Connection control:** Transport layer provides two services - **Connection-oriented** service and **Connectionless service**.
    a. A **connection- oriented service** makes a connection with the transport layer of the destination machine before transmitting the packets. Here, all the packets travel in the single route. When data transmission is completed, the connection is terminated. This type of transmission is more reliable than connectionless service. eg: **TCP**
    b. A **connectionless service** treats each segment as an individual packet, and they all travel in different routes to reach the destination. In this type of transmission, the receiver does not send an acknowledgment to the sender about the receipt of the packet. Therefore it is a faster communication technique. Eg: **UDP**

The two important protocols used in this layer are **TCP** and **UDP**. They are the different methods to send information across the internet.

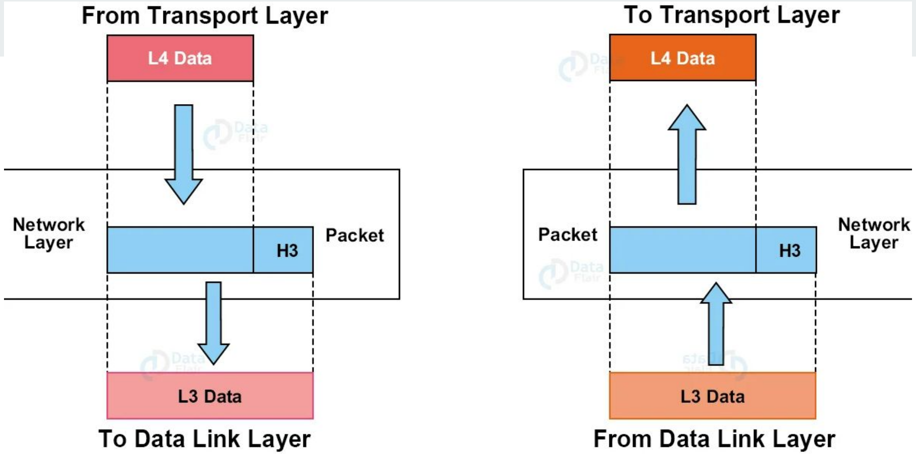
- **TCP - Transmission Control Protocol**
  - TCP is a layer 4 protocol that allows data to be sent from one computer to another.
  - It is a connection-oriented protocol, i.e. the computers first establish a connection and then it does the communication.
  - It provides acknowledgement of the received packets and is also reliable as it resends the lost packets.
  - TCP includes error-checking, and guarantees that data will be delivered in the order it was sent, making it the perfect protocol for transferring information like still images, data files, and web pages.
  - It is used by application protocols like HTTP, FTP etc.

- **UDP - User Datagram Protocol**
  - Its working is similar to the TCP as it is also used for sending and receiving the message.
  - User Datagram Protocol is a simple connectionless transport layer protocol.
  - It is an unreliable transport protocol as in this case the receiver does not send any acknowledgment when the packet is received, the sender does not wait for any acknowledgment. Therefore, this makes a protocol unreliable.
  - It is used in video and voice streaming.



Basis for Comparison	TCP	UDP
Definition	TCP establishes a connection before transmitting the data.	UDP transmits the data directly to the destination computer without verifying whether the receiver is ready to receive or not.
Connection Type	It is a Connection-Oriented protocol	It is a Connectionless protocol
Speed	slow	high
Reliability	It is a reliable protocol.	It is an unreliable protocol.
Header size	TCP header size is 20 bytes.	UDP Header size is 8 bytes.
Acknowledgement	It waits for the acknowledgement of data and has the ability to resend the lost packets.	It neither takes the acknowledgement, nor it retransmits the damaged frame.
Error checking & recovery	TCP does error checking and error recovery. Erroneous packets are retransmitted from the source to the destination. Retransmission of data packets is possible in TCP in case packet get lost or need to resend.	UDP does error checking but simply discards erroneous packets. Error recovery is not attempted. Retransmission of packets is not possible in UDP.
Protocols	TCP is used by protocols such as HTTP, HTTPs, FTP, SMTP, Telnet, SSH	UDP is used by protocols such as DNS, DHCP, TFTP, SNMP, RIP, VOIP,IPTV

Layer 3 - Network Layer



- The main aim of this layer is transferring of data from one host to another in a network.
- If the source & destination systems are connected to the same network, there is no need for a network layer. The data link layer can handle the delivery of data between systems on the same network.
- The network layer accepts data from the transport layer above, divides and encapsulates it into packets and sends it to the data link layer. The reverse procedure is done during receiving data.

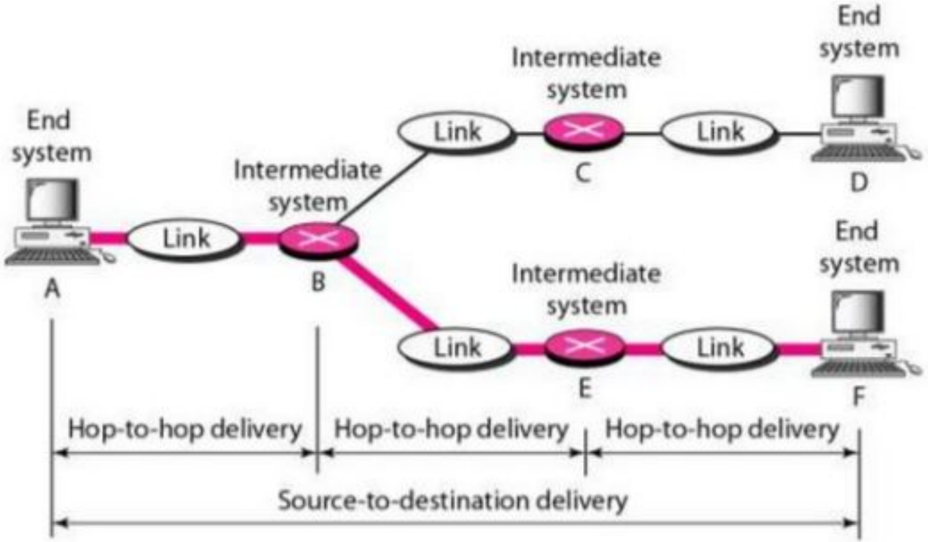
- The network layer is responsible for routing packets from the source host to the destination host. It selects the shortest path to transmit the packet to its destination, from the available different routes.
- Many networks are partitioned into sub-networks or subnets. The network layer controls the operations of the subnets. Network devices called routers operate in this layer to forward packets between the subnets or the different networks.
- The lower layers assign the physical address locally. When the data packets are routed to remote locations, a logical addressing scheme is required to differentiate the source system and the destination system. This is provided by the network layer.
- This layer also provides mechanisms for congestion control, in situations when too many packets overload the subnets.
- The network layer tackles issues like transmission delays, transmission time, avoidance of jitters etc.

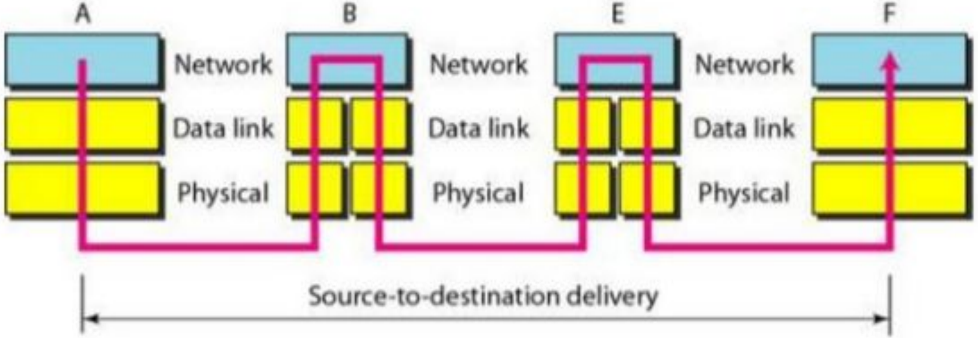
Functions of Network Layer :

Packetizing : The network layer receives data segments from the transport layer above. In case the data is too large to be transported across, it is broken down into smaller fragments to facilitate the easy flow of data. It then adds headers and encapsulates it into packets and then send to the data link layer. The packets are decapsulated after reaching the destination.

Addressing: A Network layer adds the source and destination address to the packet header. This layer follows an addressing scheme to find the correct IP address of the destination device on the internet.

Routing: Routing is the major function of the network layer, and it determines the best optimal path out of the multiple paths from source to the destination. It determines the best path based on the network conditions, the priority of service, and other factors.

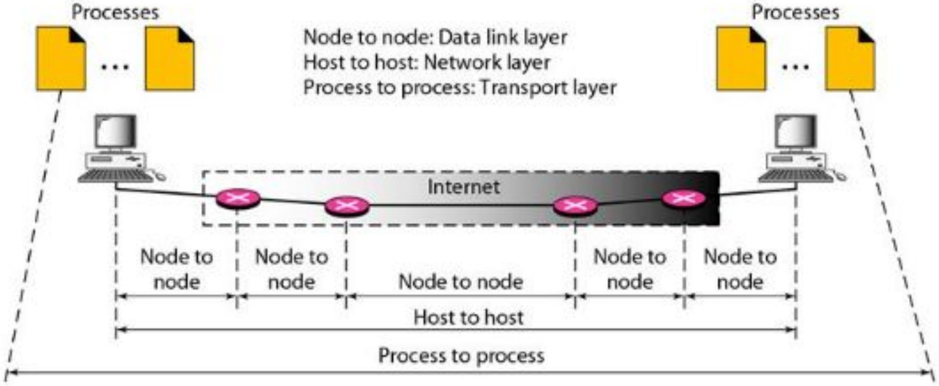




Layer 2 - Data Link Layer (DLL)

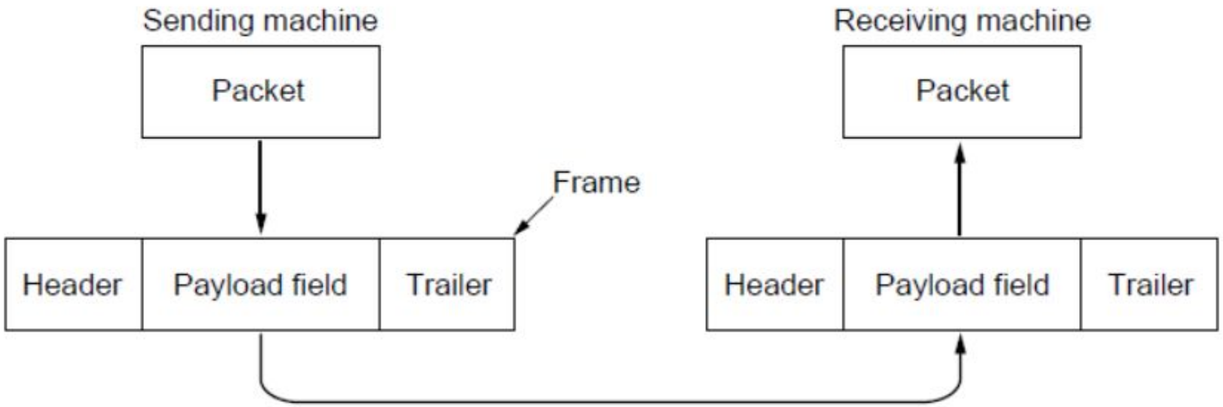
- Data link layer is the second layer of the OSI Model. It is responsible for the node-to-node delivery of data. The goal of the DLL is to provide reliable & efficient communication between adjacent nodes connected by a single communication channel.
- This is considered the most complex layer of the OSI model as it hides all the underlying complexities of the hardware from the other above layers
- The packet received from the Network layer is further divided into frames depending on the frame size of NIC(Network Interface Card). DLL also adds the Sender's and Receiver's MAC address in the header. A MAC (Media Access Control) address is the unique identifier that is assigned by the manufacturer to a piece of network hardware (like a wireless card or an ethernet card).
- When a packet or message reaches, it is the responsibility of Data Link Layer to transmit it to the Host using its MAC address.
- The receiver's MAC address is obtained by placing an ARP request (Address Resolution Protocol), asking " Who has this IP Address?" The destination host will reply with its MAC address.

- Data link layer ensures that data transfer is error-free from one node to another over the physical layer.
- Data Link Layer devices are Switch & Bridges.



Functions of Data Link Layer :

Framing : Framing is the function of DLL. At the sender side, It receives packets from the Network layer, divides it into data units called frames , adds the header and trailer at the beginning and end of each frame as shown in figure.



A frame contains

1. A frame header
2. A payload field for holding the packet
3. A frame trailer

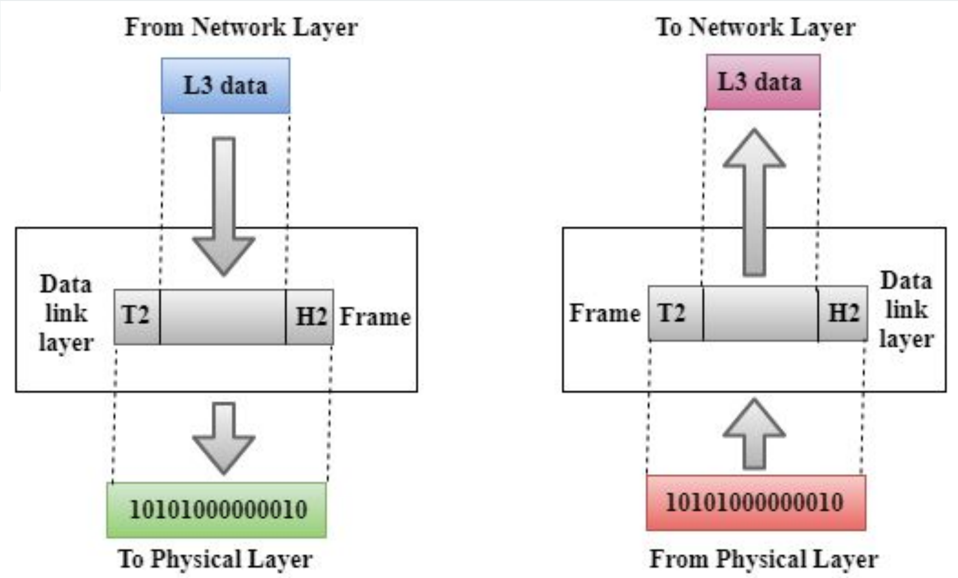
After including header & trailer information, It sends each frame bit-by-bit to the physical layer. At the receiving end, the DLL there, picks up signals from the hardware & assembles them into frames and sends them to the Network layer.

Addressing : The data link layer encapsulates the source and destination MAC address/ physical address in the header of each frame to ensure node-to-node delivery. MAC address is the unique hardware address that is assigned to the device while manufacturing.

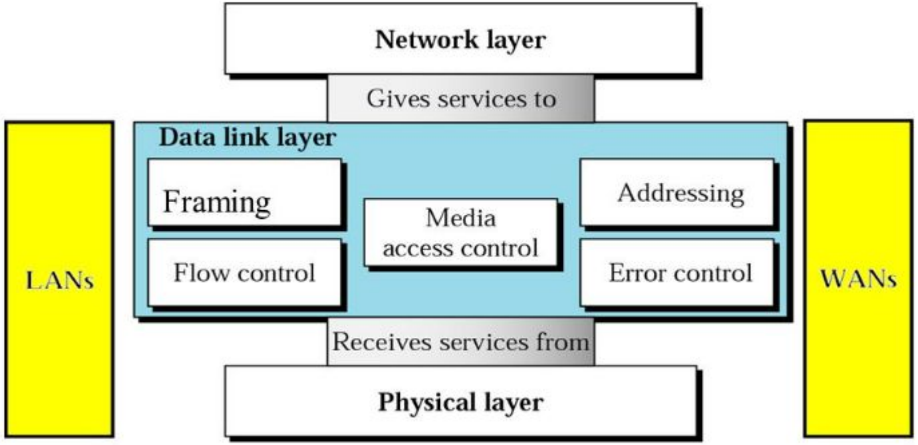
Error Control : Data can get corrupted due to various reasons like noise, attenuation, etc. So, it is the responsibility of the data link layer, to detect the error in the transmitted data and correct it using error detection and correction techniques. The sender calculates the checksum & sends the checksum together with the data. These extra bits are added into the frame's trailer. The receiver recomputes the checksum & compares it with the received value. If they differ, an error has occurred and the frame is discarded. The error control protocol returns a positive or negative acknowledgment to the sender.

Flow Control : It is a technique through which the constant data rate is maintained between the sender & the receiver, so that no data gets corrupted. The sender may be a server with high processing speed and the receiver may be a low end machine. So, it's the responsibility of DLL to synchronize the sender's and receiver's speeds and establish flow control between them. Some of the flow control techniques are Stop-and-wait, Go-Back-N,Selective Repeat etc

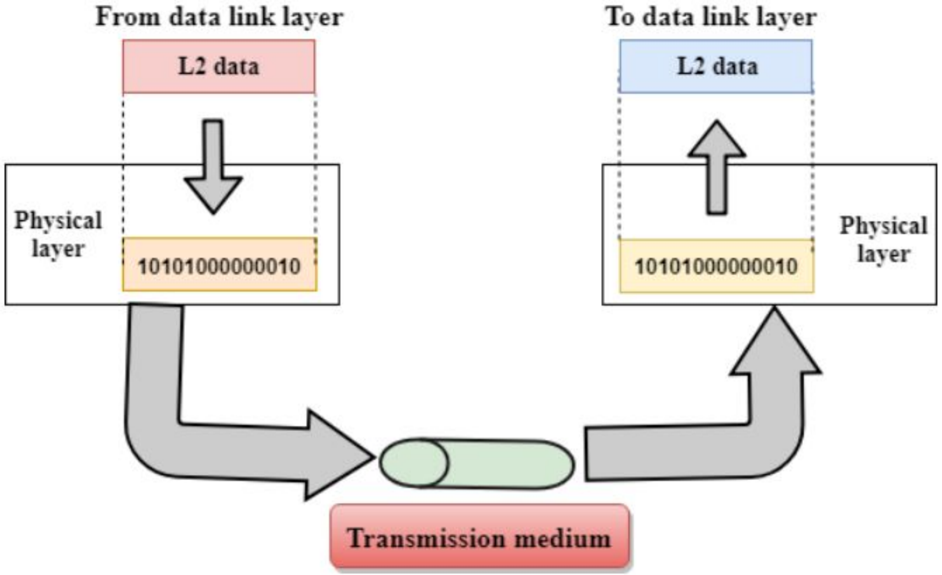
Access Control : When multiple devices share the same communication channel, there is a high probability of collision. So it's the responsibility of DLL protocols to determine which device has control over the channel at any point of time.



Data Link Layer Functions



Layer 1 - Physical Layer



Modes of Transmission Medium :

Simplex mode: *In this mode, the devices, either can only transmit the data or only receive the data. Example- Input from keyboards, monitors, TV broadcasting, Radio broadcasting, etc.*

Half Duplex mode: *In this mode, out of two devices, both devices can send and receive the data but only one at a time not simultaneously. Example- Walkie-Talkie*

Full Duplex mode: *In this mode, both devices can send and receive the data simultaneously. Example- Telephone System, Chatting applications, etc.*

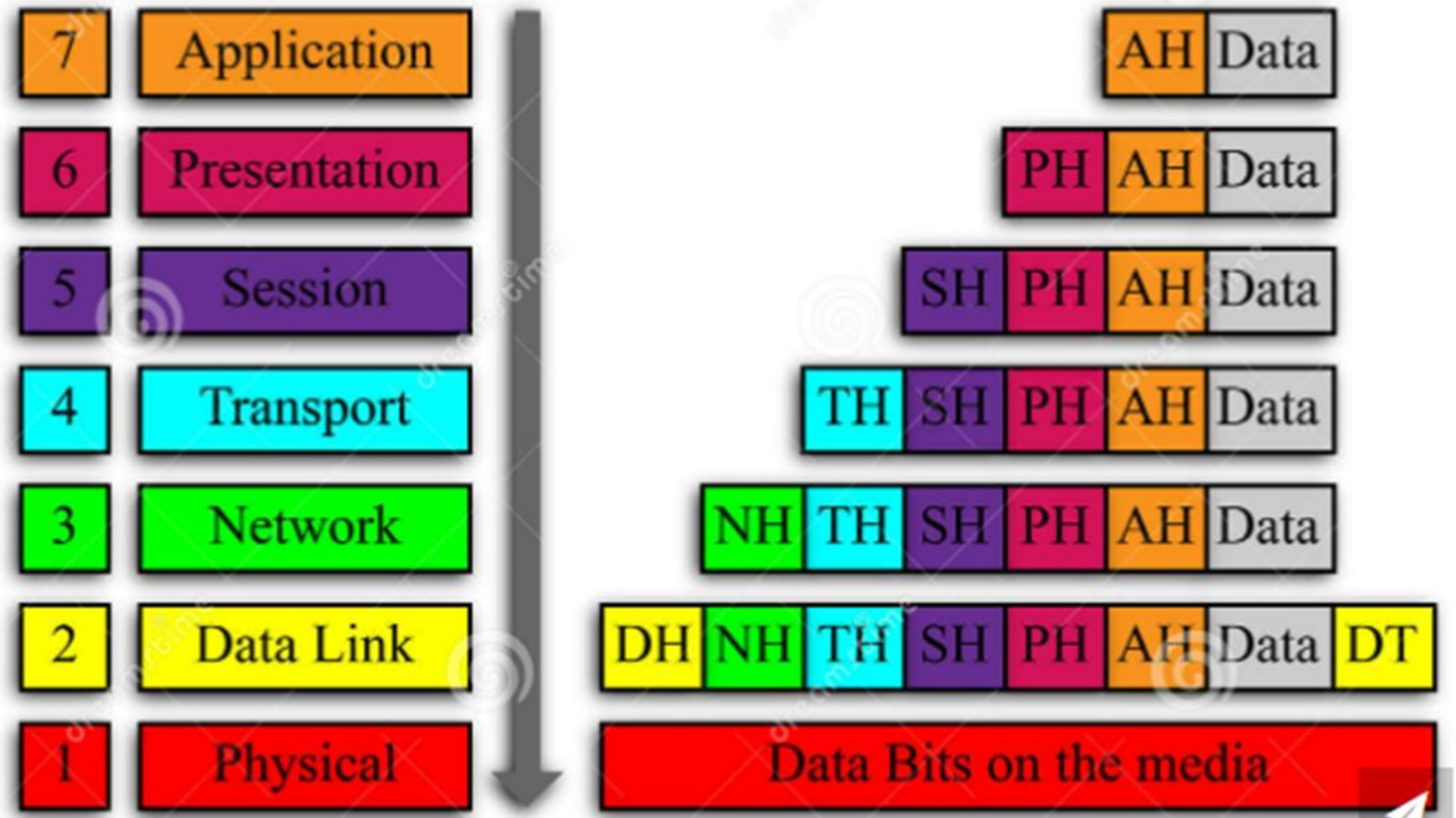
- The physical layer is the lowest layer of OSI Model.
- It deals with the setup of *physical connection* to the network and with transmission and reception of signals.
- Physical layer sends data bits from one device to another device.
- It consists of various network components such as power plugs, connectors, cable types, hubs, repeaters, network adapters, modems etc.
- Physical Layer defines the types of encoding (i.e. how the 0's and 1's are encoded in a signal).
- It establishes, maintains & deactivates the physical connection.
- Physical layer decides the transmission mode between two devices, ie full-duplex/half-duplex

Functions of Physical Layer :

1. **Data Rate Control:** This layer defines the rate of data transmission ie How many bits a sender can send per second.

2. **Representation of Bits:** The data in this layer consists of a stream of bits. These bits must be encoded into signals for transmission. This layer defines the type of encoding (i.e. how 0's and 1's are changed to signal) . The purpose of the Physical layer is to create the electrical, optical, or wave signal that represents the bits in each frame. These signals are then sent on the media one at a time.
3. **Bit Synchronization:** The physical layer provides bit synchronization for synchronous serial communications. It deals with the synchronization of the transmitter and receiver. The physical layer provides the synchronization of the bits by providing a clock. This clock controls both sender and receiver thus providing synchronization at bit level.
4. **Line Configuration:** It defines the way how two or more devices can be connected physically. Ie Point-to-point connection or Multi-point connection
5. **Topologies:** Physical layer specifies the way in which the different, devices/nodes are arranged in a network i.e. bus, star, or mesh topology.
6. **Data Transmission Modes:** Physical layer also defines the way in which the data flows between the two connected devices. The way in which data is transmitted from one device to another is known as transmission mode. The various transmission modes possible are Simplex, half-duplex and full-duplex.
7. **Multiplexing and Demultiplexing** : Multiplexing is a technique to mix and send multiple data streams over a single medium.To avoid collisions, we can use time division or wavelength division. At receiver end, demultiplexing is used to convert a single signal into 'n' signals.

OSI Model



7 Application — AH Data

6 Presentation — PH AH Data

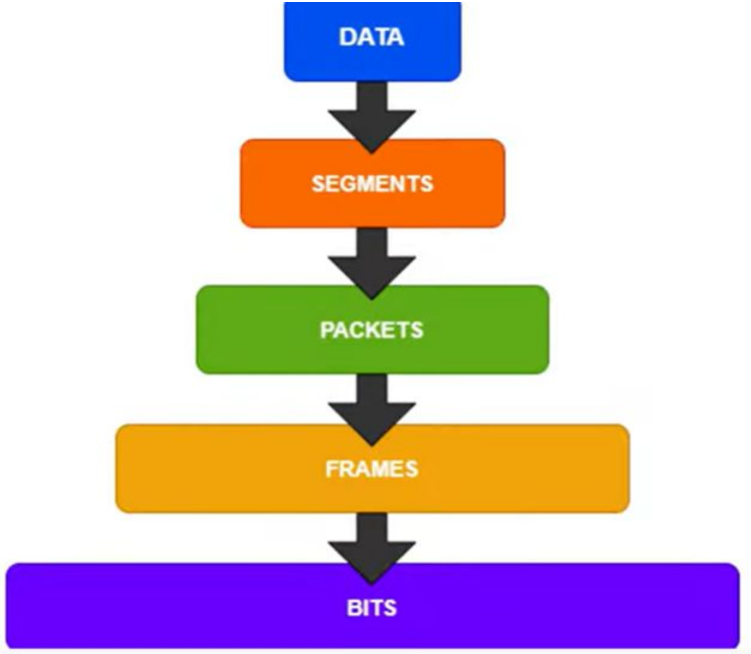
5 Session — SH PH AH Data

4 Transport — TH SH PH AH Data

3 Network — NH TH SH PH AH Data

2 Data Link — DH NH TH SH PH AH Data DT

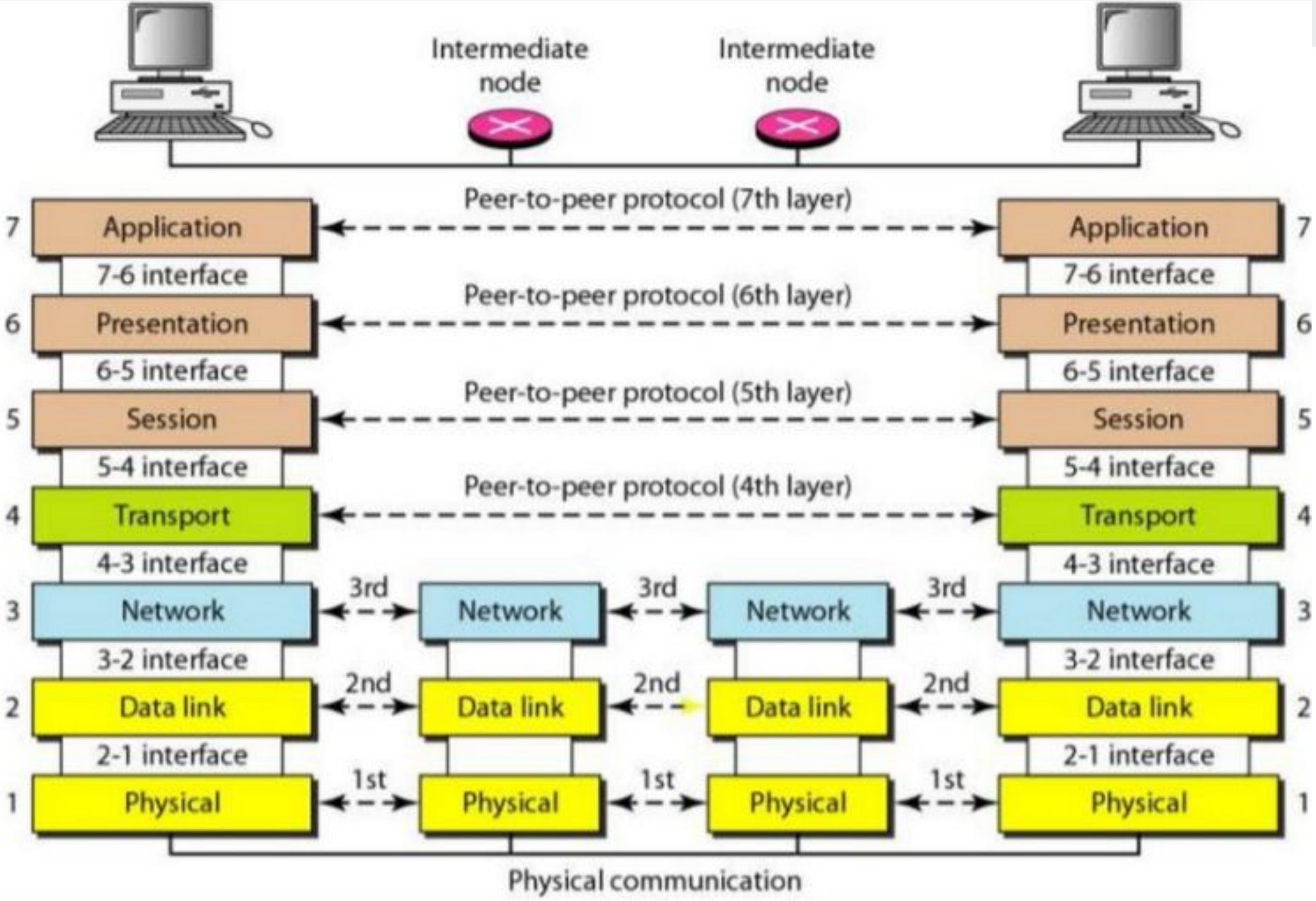
1 Physical — Data Bits on the media



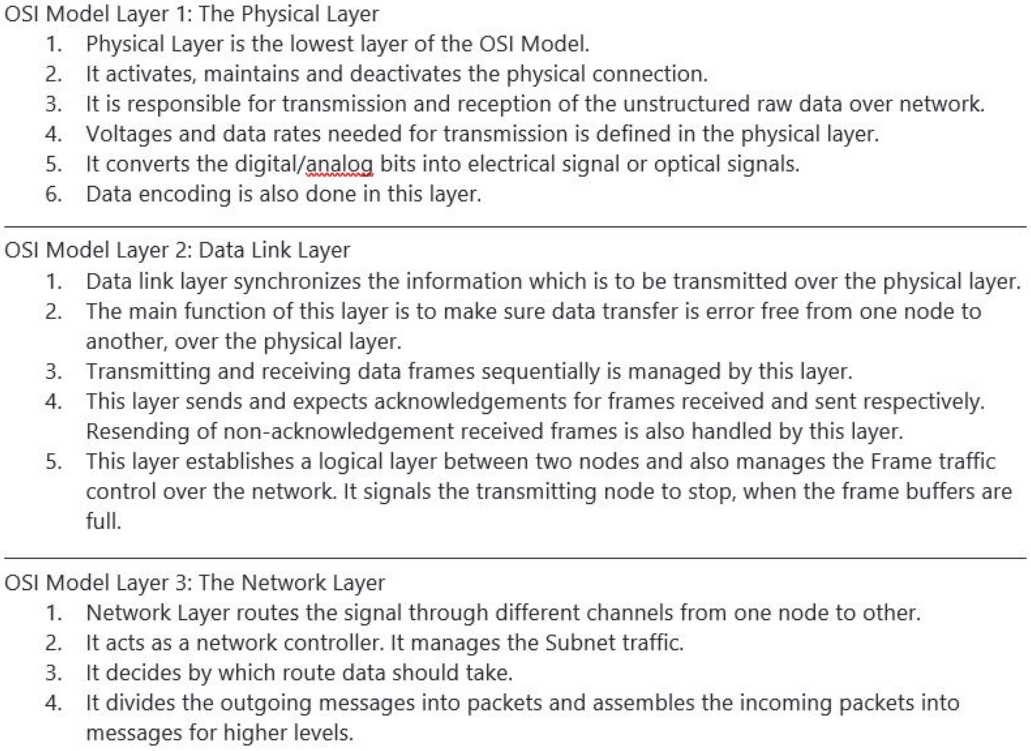
Five Conversion Steps of Data Encapsulation

Data >> **Segments** >> **Packets** >> **Frames** >> **Bits**

1. Upper layers convert and format the information into **data** and send it to the Transport Layer.
2. The Transport layer turns the data into **segments** and adds headers then sends them to the Network layer.
3. The Network layer receives the segments and converts them into **packets** and adds header information (logical addressing) and sends them to the Data Link Layer.
4. The Data Link layer receives the packets and converts them into **frames** and adds header information (physical source and destination addresses) and sends the frames to the Physical Layer.
5. The Physical layer receives the frames and converts them into **bits** to be put on the network medium.


Intermediate node
Intermediate node
Peer-to-peer protocol (7th layer)
Peer-to-peer protocol (6th layer)
Peer-to-peer protocol (5th layer)
Peer-to-peer protocol (4th layer)
7 Application
7-6 interface
6 Presentation
6-5 interface
5 Session
5-4 interface
4 Transport
4-3 interface
3 Network
3-2 interface
2 Data link
2-1 interface
1 Physical
3rd
2nd
1st
Network
Data link
Physical
Network
Data link
Physical
Application 7
7-6 interface
Presentation 6
6-5 interface
Session 5
5-4 interface
Transport 4
4-3 interface
Network 3
3-2 interface
Data link 2
2-1 interface
Physical 1
Physical communication

Functions of Different Layers



OSI Model Layer 1: The Physical Layer

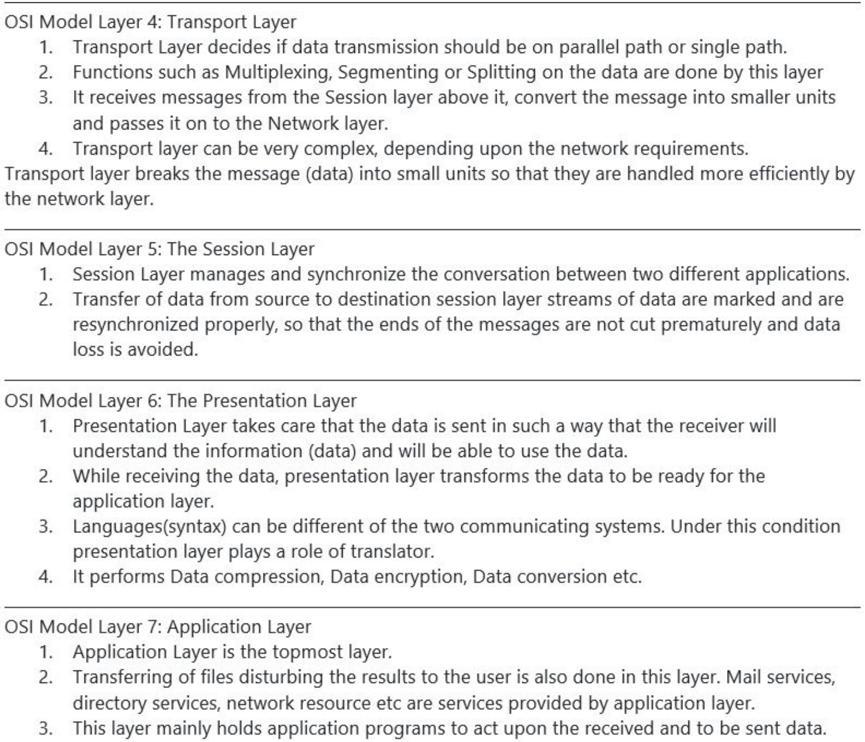
1. Physical Layer is the lowest layer of the OSI Model.
2. It activates, maintains and deactivates the physical connection.
3. It is responsible for transmission and reception of the unstructured raw data over network.
4. Voltages and data rates needed for transmission is defined in the physical layer.
5. It converts the digital/analog bits into electrical signal or optical signals.
6. Data encoding is also done in this layer.

OSI Model Layer 2: Data Link Layer

1. Data link layer synchronizes the information which is to be transmitted over the physical layer.
2. The main function of this layer is to make sure data transfer is error free from one node to another, over the physical layer.
3. Transmitting and receiving data frames sequentially is managed by this layer.
4. This layer sends and expects acknowledgements for frames received and sent respectively. Resending of non-acknowledgement received frames is also handled by this layer.
5. This layer establishes a logical layer between two nodes and also manages the Frame traffic control over the network. It signals the transmitting node to stop, when the frame buffers are full.

OSI Model Layer 3: The Network Layer

1. Network Layer routes the signal through different channels from one node to other.
2. It acts as a network controller. It manages the Subnet traffic.
3. It decides by which route data should take.
4. It divides the outgoing messages into packets and assembles the incoming packets into messages for higher levels.



OSI Model Layer 4: Transport Layer

1. Transport Layer decides if data transmission should be on parallel path or single path.
2. Functions such as Multiplexing, Segmenting or Splitting on the data are done by this layer
3. It receives messages from the Session layer above it, convert the message into smaller units and passes it on to the Network layer.
4. Transport layer can be very complex, depending upon the network requirements.

Transport layer breaks the message (data) into small units so that they are handled more efficiently by the network layer.

OSI Model Layer 5: The Session Layer

1. Session Layer manages and synchronize the conversation between two different applications.
2. Transfer of data from source to destination session layer streams of data are marked and are resynchronized properly, so that the ends of the messages are not cut prematurely and data loss is avoided.

OSI Model Layer 6: The Presentation Layer

1. Presentation Layer takes care that the data is sent in such a way that the receiver will understand the information (data) and will be able to use the data.
2. While receiving the data, presentation layer transforms the data to be ready for the application layer.
3. Languages(syntax) can be different of the two communicating systems. Under this condition presentation layer plays a role of translator.
4. It performs Data compression, Data encryption, Data conversion etc.

OSI Model Layer 7: Application Layer

1. Application Layer is the topmost layer.
2. Transferring of files disturbing the results to the user is also done in this layer. Mail services, directory services, network resource etc are services provided by application layer.
3. This layer mainly holds application programs to act upon the received and to be sent data.

TCP/ IP Model

- The **OSI Model** is a reference/theoretical model. It was designed to describe the functions of the communication system by dividing the communication procedure into smaller and simpler components.

- The TCP/IP model was developed prior to the OSI model.

- TCP/IP is an implemented model; designed and developed by US Dept of Defense -ARPA Advanced Research Projects Agency. In 1982, the United States Dept of Defense declared TCP/IP as the standard for all military computer networking.

- **TCP/IP** stands for Transmission Control Protocol/Internet Protocol. **The two main protocols used in this model are TCP and IP.** The TCP/IP model is a concise version of the OSI model.

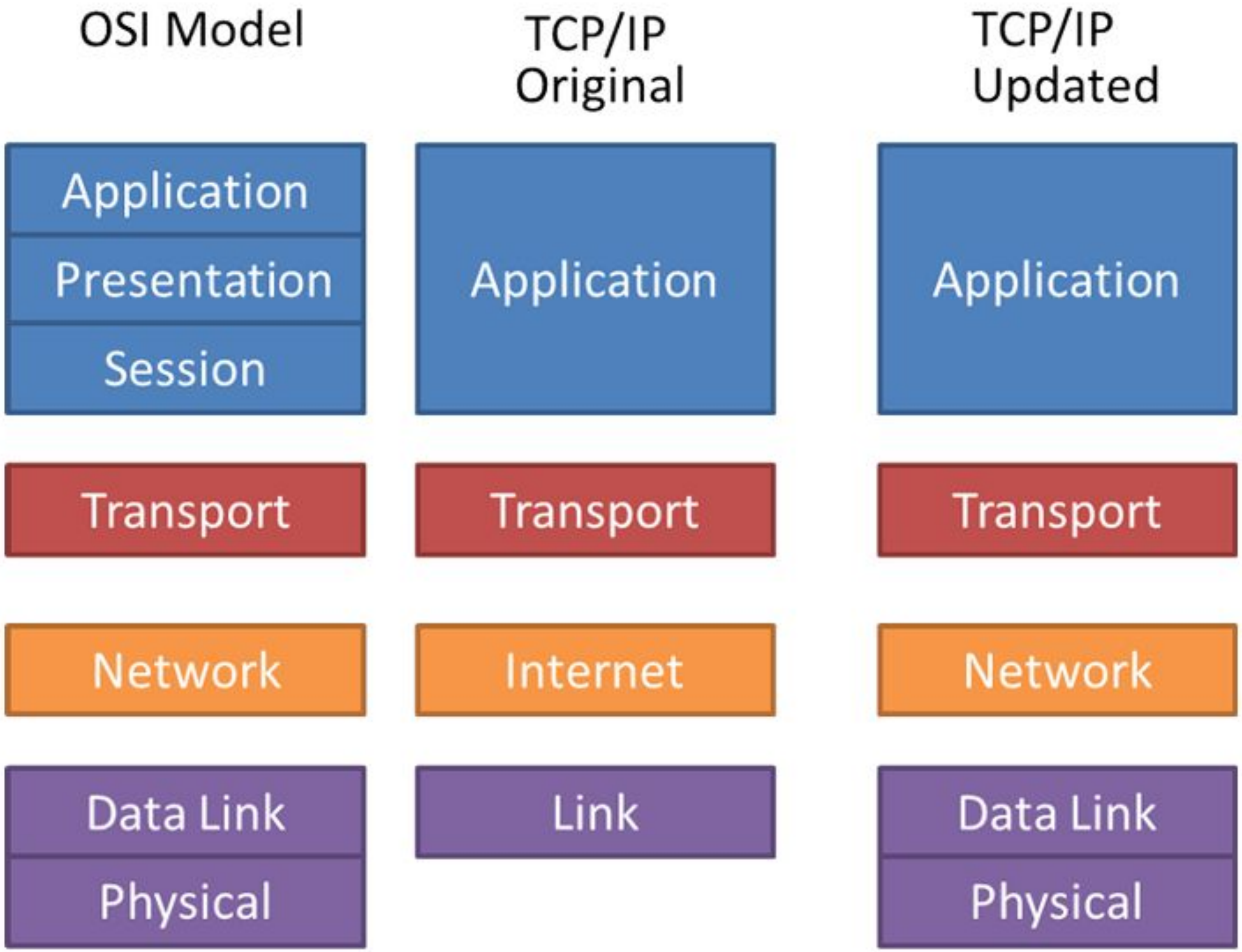
- The TCP/ IP is a protocol suite used in the Internet today. Its ability to connect multiple networks made this model a huge success.

- The original TCP/IP model was defined with only four layers, but the updated TCP/IP model has five layers. Each of these layers provide a specific functionality.

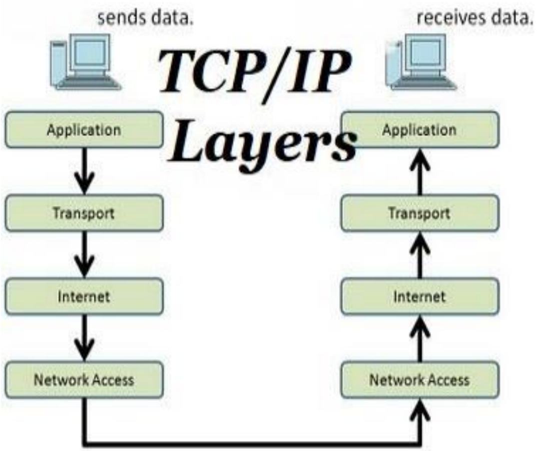
- The four layers of original TCP/IP model are Application Layer, Transport Layer, Internet Layer, Network Access Layer (Host-to-network layer)

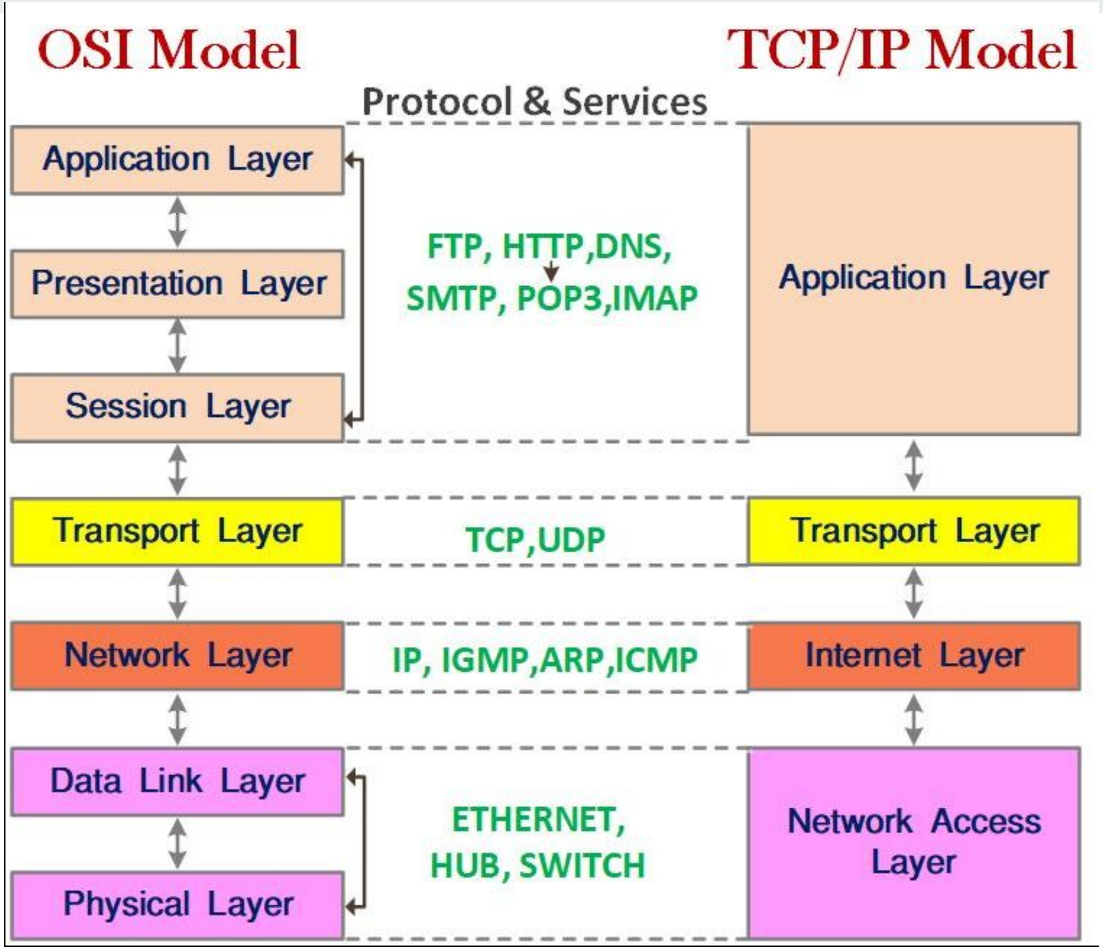
- The 5 layers in the current model are the - application layer, transport layer, network layer, data link layer and physical layer.

- This model supports both Client-Server & Peer-to-peer architecture. It is interoperable with all operating systems and can interface with every other machine.



OSI Model	TCP/IP Original	TCP/IP Updated
Application Presentation Session	Application	Application
Transport	Transport	Transport
Network	Internet	Network
Data Link Physical	Link	Data Link Physical




OSI Model
TCP/IP Model
Protocol & Services
Application Layer
Presentation Layer
Session Layer
FTP, HTTP,DNS,
SMTP, POP3,IMAP
Application Layer
Transport Layer
TCP,UDP
Transport Layer
Network Layer
IP, IGMP,ARP,ICMP
Internet Layer
Data Link Layer
Physical Layer
ETHERNET,
HUB, SWITCH
Network Access Layer

TCP /IP Protocol Suite

Application Layer - It is the topmost layer of the TCP/IP model and it interacts with software applications . Its functions are similar to the combination of the application layer, session layer, and presentation layer of OSI Model. It is responsible for user interface specifications. It contains high level protocols used in the process to process communication across computer networks. Some of the protocols used in this layer are:

HTTP - It stands for Hypertext Transfer Protocol. It permits applications such as browsers to upload & download webpages. HTTPS stands for HTTP-Secure. It is a combination of HTTP with SSL(Secure Socket Layer). It is efficient in cases where the browser need to fill out forms, sign in, authenticate and carry out bank transactions.

SMTP - It stands for Simple Mail Transfer Protocol. It is used for sending and receiving electronic mails.

TELNET - This is a virtual terminal protocol which allows the user in one machine to log into another remote machine over a network.

FTP - It stands for File Transfer Protocol. It is a standard communication protocol used for transferring files from one computer to another over a network. It uses TCP at Transport layer.

DNS - Domain Name System. Allows the network determine the IP address from name. & viceversa.

Transport Layer -

This layer is similar to the transport layer of the OSI model. Transport layer builds on the network layer in order to provide data transport from a process on a source system machine to a process on a destination system.

The transport layer is responsible for the reliability, flow control, and correction of data which is being sent over the network. It determines how much data should be sent where and at what rate. This layer builds on the message which are received from the application layer. It helps ensure that data units are delivered error-free and in sequence. This layer communicates data between two applications making use of Port Numbers.

Important functions of Transport Layers:

- It divides the message received from the application llayer into segments and numbers them to make a sequence.
- Transport layer makes sure that the message is delivered to the correct process on the destination machine.
- It also makes sure that the entire message arrives without any error else it should be retransmitted.

The main protocols of this layer are -

TCP - It stands for Transmission Control Protocol. It is a connection-oriented protocol. At the sending end, TCP divides the whole message into smaller units known as segments, and each segment contains a sequence number which is required for reordering the segments to form an original message. At the receiving end, TCP collects all the segments and reorders them based on sequence numbers. It provides reliable communication and error-free delivery of data from the source to the destination host. i.e. It gives accurate delivery rather than timely delivery. It is used by many internet applications including World Wide Web(WWW), email.

UDP - It stands for User Datagram Protocol. It provides simple, fast, cost-effective & connectionless service. It is an unreliable protocol since performs error checking, but it discards erroneous packets. The User Datagram Protocol (UDP) was developed for use by application protocols that do not require reliability, acknowledgment or flow control features at the transport layer. It prioritizes speed over the accuracy of delivery. It is also compatible with packet broadcasting (ie for sending all over the network) and multicasting.

Internet Layer -

An internet layer is the third layer of the TCP/IP model. It is also known as a network layer.

- It is compared to the network layer of the OSI model.
- The main responsibility of the internet layer is to send the packets from any network, and they arrive at the destination irrespective of the route they take.
- The internet layer encapsulates the segments from the transport layer into envelopes called packets and then determines the path to take it across the network.

Layer-management protocols that belong to the network layer are:

Routing protocols
Multicast group management
Network-layer address assignment.

The main protocols used in this layer are: At the network layer (or, more accurately, the internetwork layer), TCP/IP supports the Internetworking Protocol. IP, in turn, uses four supporting protocols: ARP, RARP, ICMP, and IGMP.

IP (Internet Protocol) - It is responsible for the transmission of data packets from the source to the destination host. It is the most significant part of the entire TCP/IP suite. It is implemented in two versions, IPv4 and IPv6.

The main responsibilities of the IP protocol are-

- **Data Encapsulation and Formatting:** An IP protocol accepts the data from the transport layer protocol. It ensures that the data is sent and received securely, it encapsulates the data into message known as IP datagram.
- **IP Addressing:** This protocol implements logical host addresses known as IP addresses. The IP addresses are used by the internet and higher layers to identify the device .
- **Host-to-host communication:** It determines the path through which the data is to be transmitted.
- **Fragmentation and Reassembly:** The limit imposed on the size of the IP datagram by data link layer protocol is known as Maximum Transmission unit (MTU). If the size of IP datagram is greater than the MTU unit, then the IP protocol splits the datagram into smaller units so that they can travel over the local network. Fragmentation can be done by the sender or intermediate router. At the receiver side, all the fragments are reassembled to form an original message.
- **Routing:** When IP datagram is sent over the same local network, it is known as direct delivery. When source and destination are on the distant network, then the IP datagram is sent indirectly. This can be accomplished by routing the IP datagram through various devices such as routers.

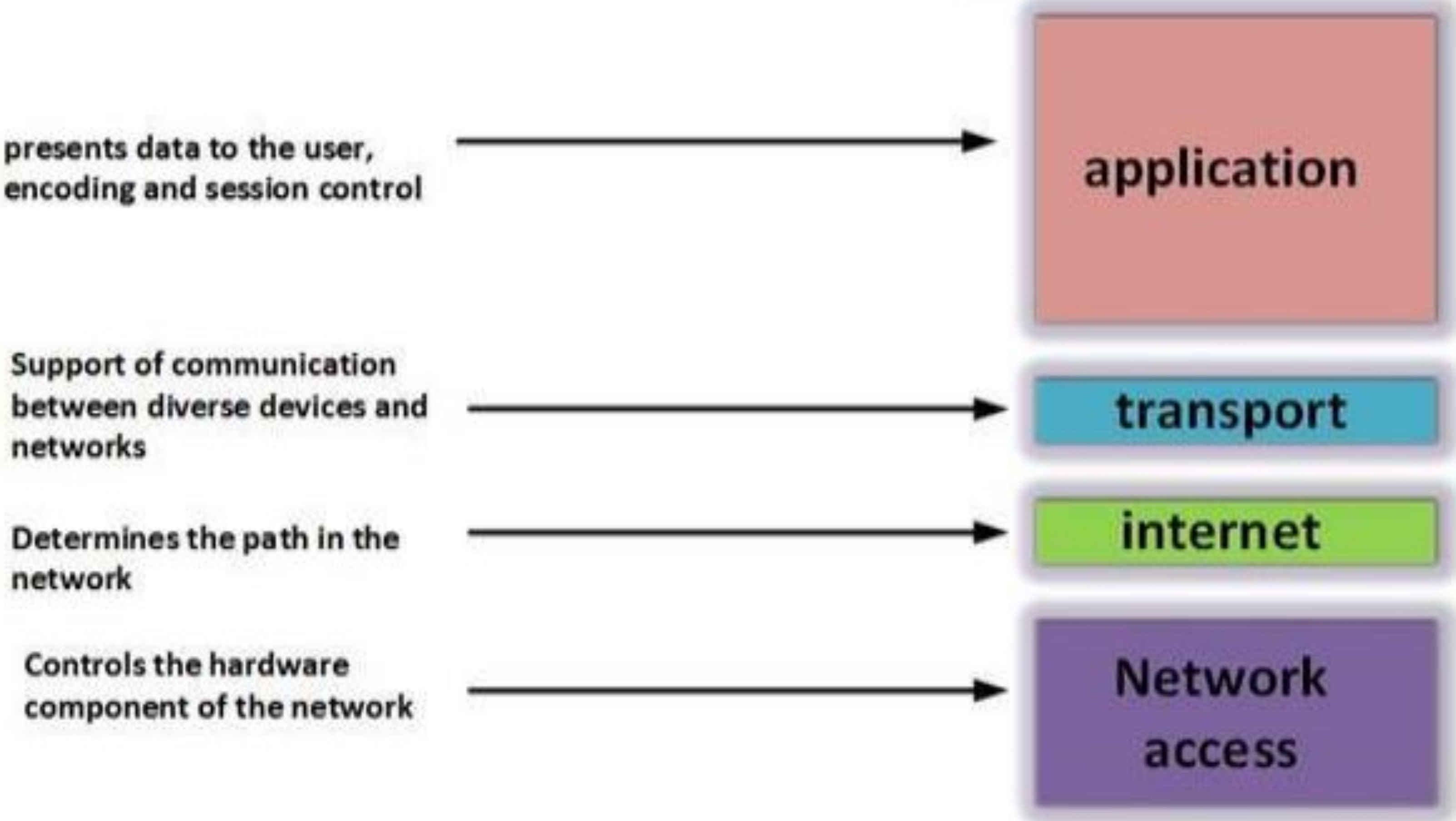
ARP (Address Resolution Protocol)- Its main responsibility is to find the physical or hardware address of the host using the IP address.
ICMP (Internet Control Message Protocol)- It is a mechanism used by the hosts or routers to send notifications regarding datagram problems back to the sender.
IGMP (Internet Group Message Protocol)- It is used for the transmission of data to a group of networks. For eg. online streaming.

Link Layer / Network Access Layer -

Network Access Layer is the fourth-layer of TCP/IP model and it is the interface to the actual network hardware. So this layer is also called a network interface layer. It helps you to define details of how data should be sent using the network. The link layer is the lowest layer in the TCP/IP model.

- It is the combination of the Physical layer and Data Link layer defined in the OSI reference model.
- At this layer,TCP/IP does not specifically define any protocol here but supports all the standard protocols.
- It defines how the data should be sent physically through the network.
- The functions carried out by this layer are encapsulating the IP datagram into frames transmitted by the network and mapping of IP addresses into physical addresses.

TCP/IP model



presents data to the user, encoding and session control

application

Support of communication between diverse devices and networks

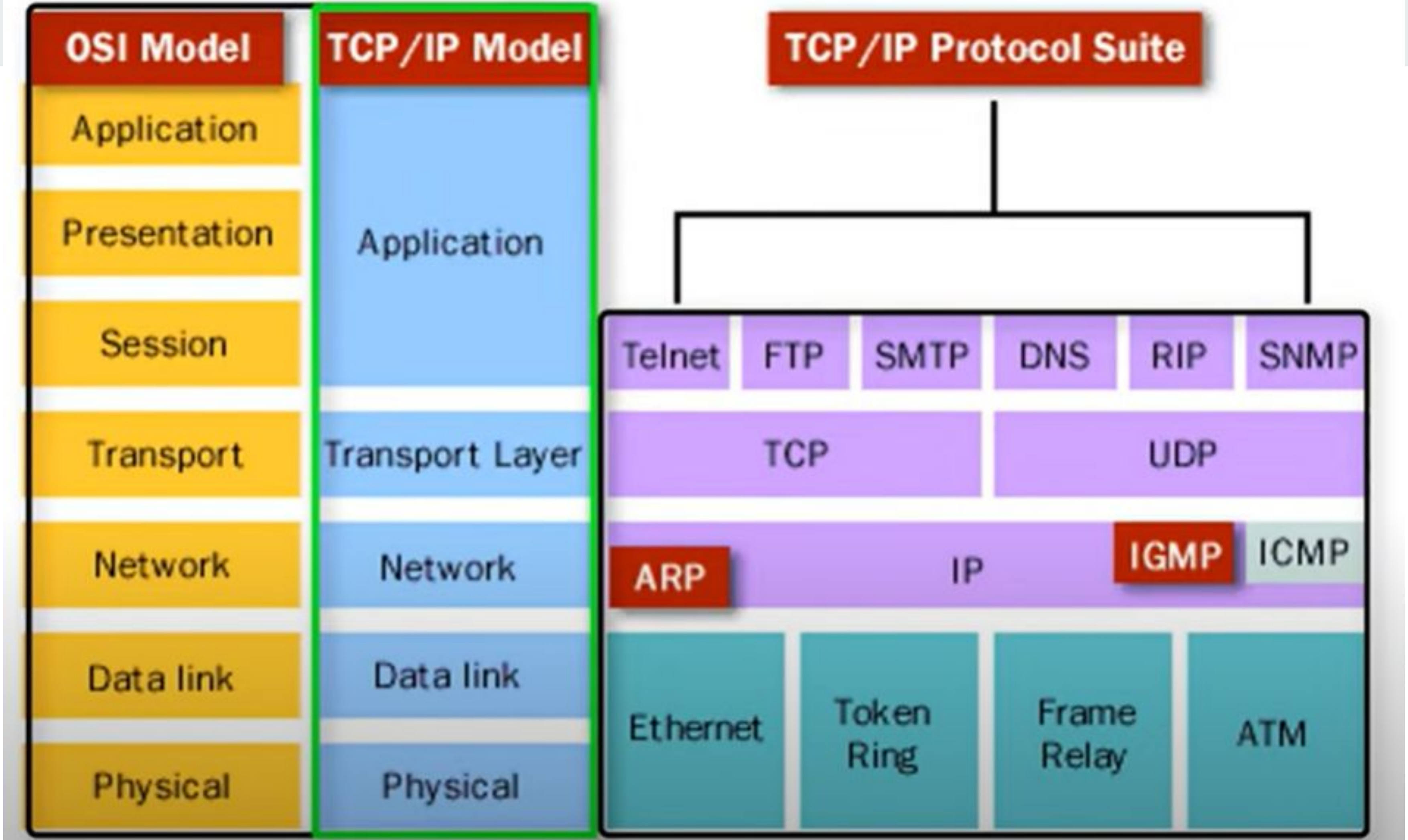
transport

Determines the path in the network

internet

Controls the hardware component of the network

Network access



OSI Model

Application

Presentation

Session

Transport

Network

Data link

Physical

TCP/IP Model

Application

Transport Layer

Network

Data link

Physical

TCP/IP Protocol Suite

Telnet

FTP

SMTP

DNS

RIP

SNMP

TCP

UDP

ARP

IP

IGMP

ICMP

Ethernet

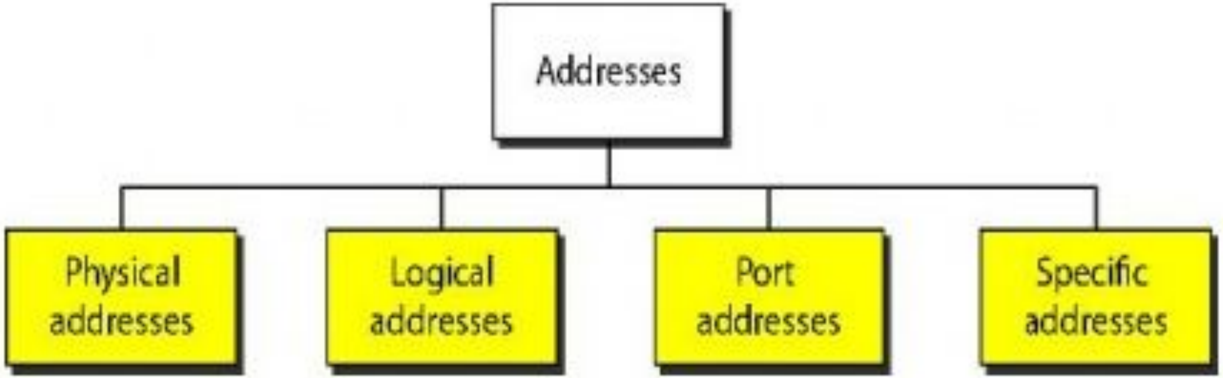
Token Ring

Frame Relay

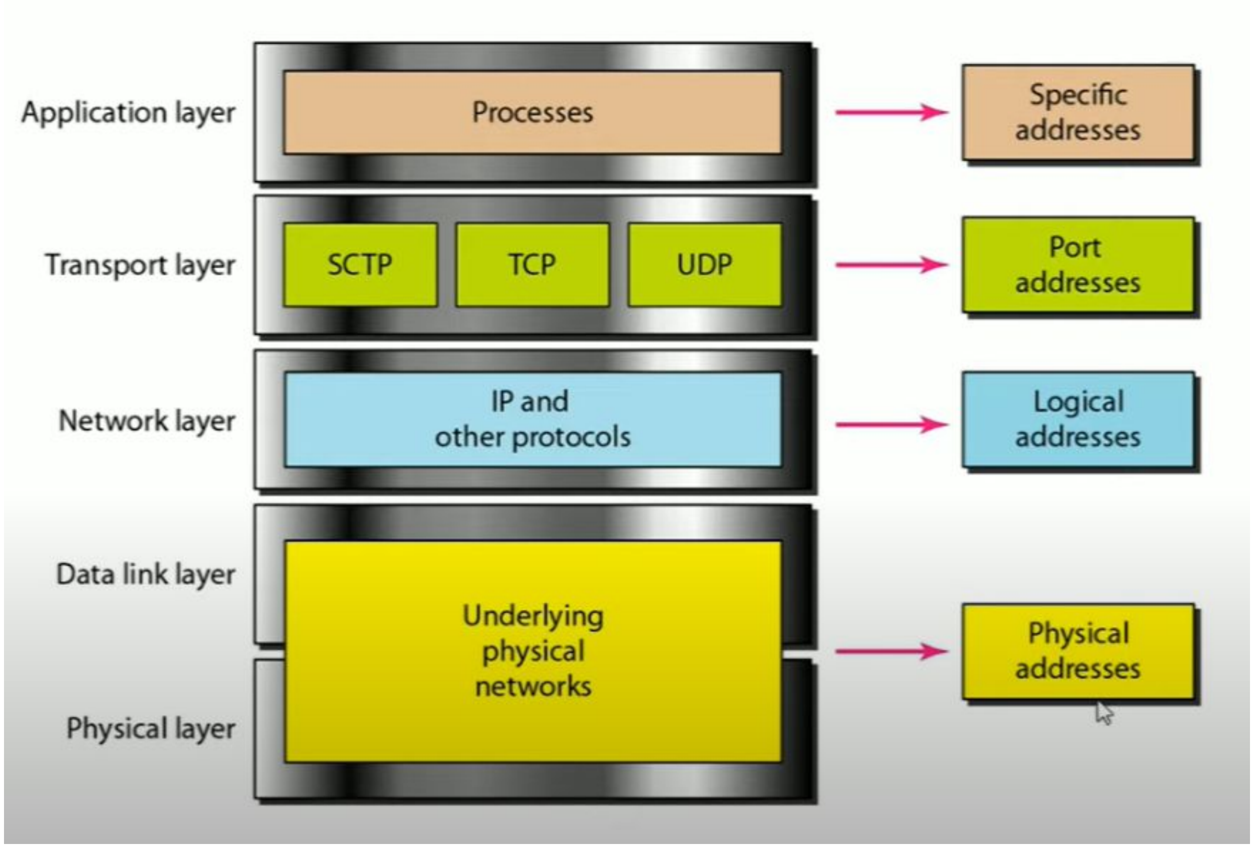
ATM

ADDRESSING:

Four levels of addresses are used in an internet employing the TCP/IP protocols:
physical (link) addresses, logical (IP) addresses, port addresses, and specific addresses.

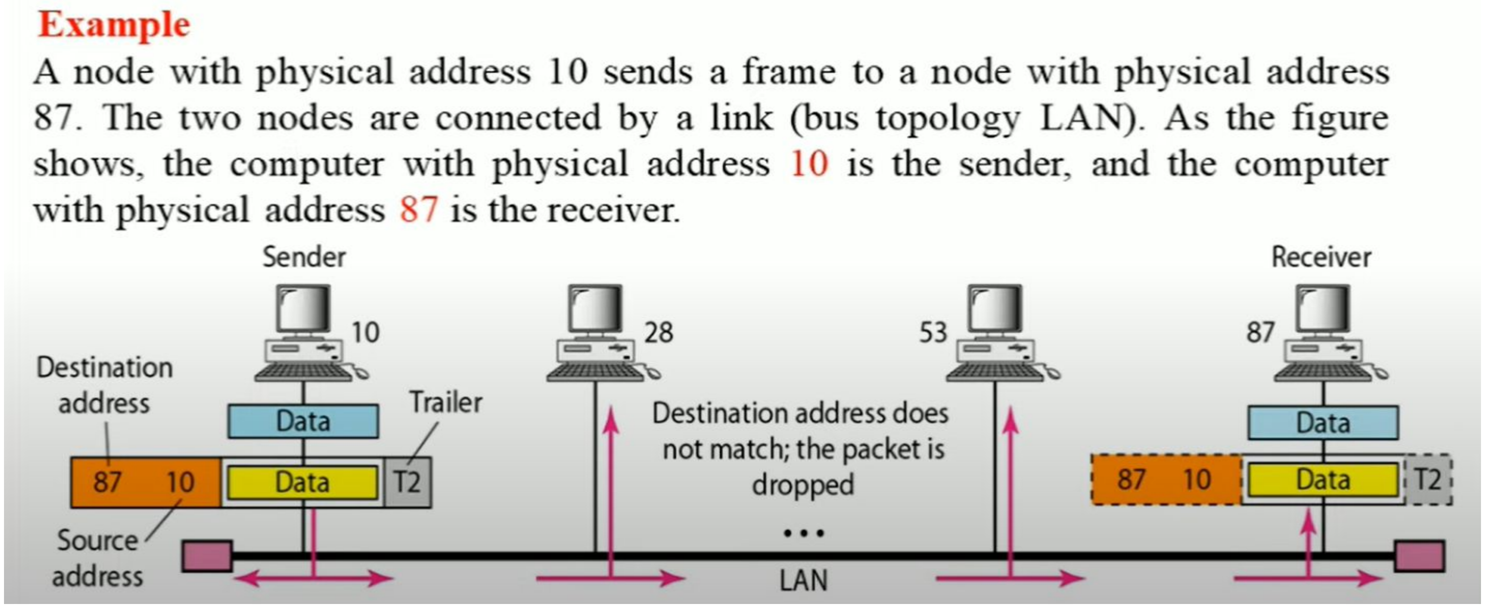


Relationship of layers and addresses in TCP/IP



1. Physical Addresses

- The physical address, also known as the link address/LAN address/MAC address. It is the lowest-level address.
- It is the address of a node as defined by its LAN or WAN.
- It is included in the frame used by the data link layer/ network access layer.
- The size and format of these addresses vary depending on the network. Most Local Area Networks use a **48-bit** (6-byte) physical address with 12 hexadecimal digits ; every byte separated by a colon Eg: 07:2C:05:04:2E:34 . MAC address is imprinted on the network interface card (NIC).



Example

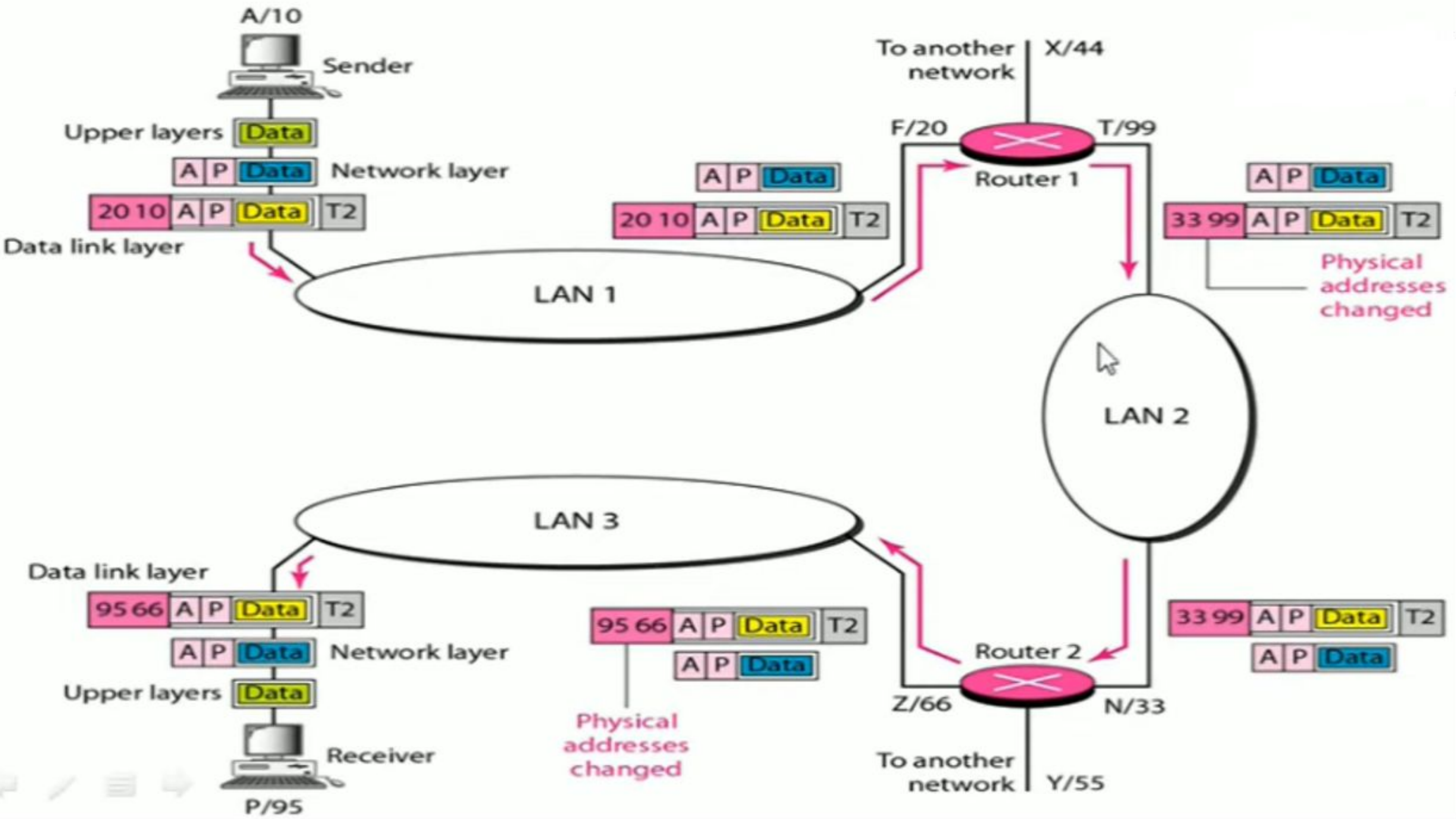
A node with physical address 10 sends a frame to a node with physical address 87. The two nodes are connected by a link (bus topology LAN). As the figure shows, the computer with physical address 10 is the sender, and the computer with physical address 87 is the receiver.

2. Logical Addresses

- Used in Internet / Network layer of TCP/IP. It is also known as IP address.
- Logical addresses are necessary for universal communications.
- Physical addresses are not adequate in an internetwork environment where different networks can have different physical address formats.
- A universal addressing scheme is needed, where each host can be identified uniquely regardless of the underlying physical networks.
- A logical address in the Internet is currently a **32-bit address** that can uniquely define a host connected to the Internet. Eg: 192.168.17.14

Example

The figure shows a part of an internet with two routers connecting three LANs. Each device [computer/ router] has a logical address and a physical address.


A/10
Sender
Upper layers
Data
A P Data
Network layer
20 10 A P Data T2
Data link layer
LAN 1
A P Data
20 10 A P Data T2
To another network
X/44
F/20
T/99
Router 1
A P Data
33 99 A P Data T2
Physical addresses changed
LAN 2
LAN 3
Data link layer
95 66 A P Data T2
A P Data
Network layer
Upper layers
Data
Receiver
P/95
95 66 A P Data T2
A P Data
Physical addresses changed
Router 2
Z/66
N/33
33 99 A P Data T2
A P Data
To another network
Y/55

3. Port Addresses

- Used in Transport layer of TCP/IP.
- The IP address and the physical address are necessary for a quantity of data to travel from a source to the destination host. However, arrival at the destination host is not the final objective of data communications on the Internet.
- Today, computers are devices that can run multiple processes at the same time. The end objective of Internet communication is a process communicating with another process. For example, computer A can communicate with computer C by using TELNET. At the same time, computer A communicates with computer B by using the File Transfer Protocol (FTP). For these processes to receive data simultaneously, we need a method to label the different processes. In other words, they need addresses. **In the TCP/IP architecture, the label assigned to a process is called a port address. A port address in TCP/IP is 16 bits in length. eg: 21, 753 etc.**
- The physical addresses change from hop to hop, but logical & port addresses remain the same from source to destination.

Example

Figure shows two computers communicating via the Internet. The sending computer is running three processes at this time with port addresses a, b, and c. The receiving computer is running two processes at this time with port addresses j and k. Process a in the sending computer needs to communicate with process j in the receiving computer. **Note that although physical addresses change from hop to hop, logical and port addresses remain the same from the source to destination.**

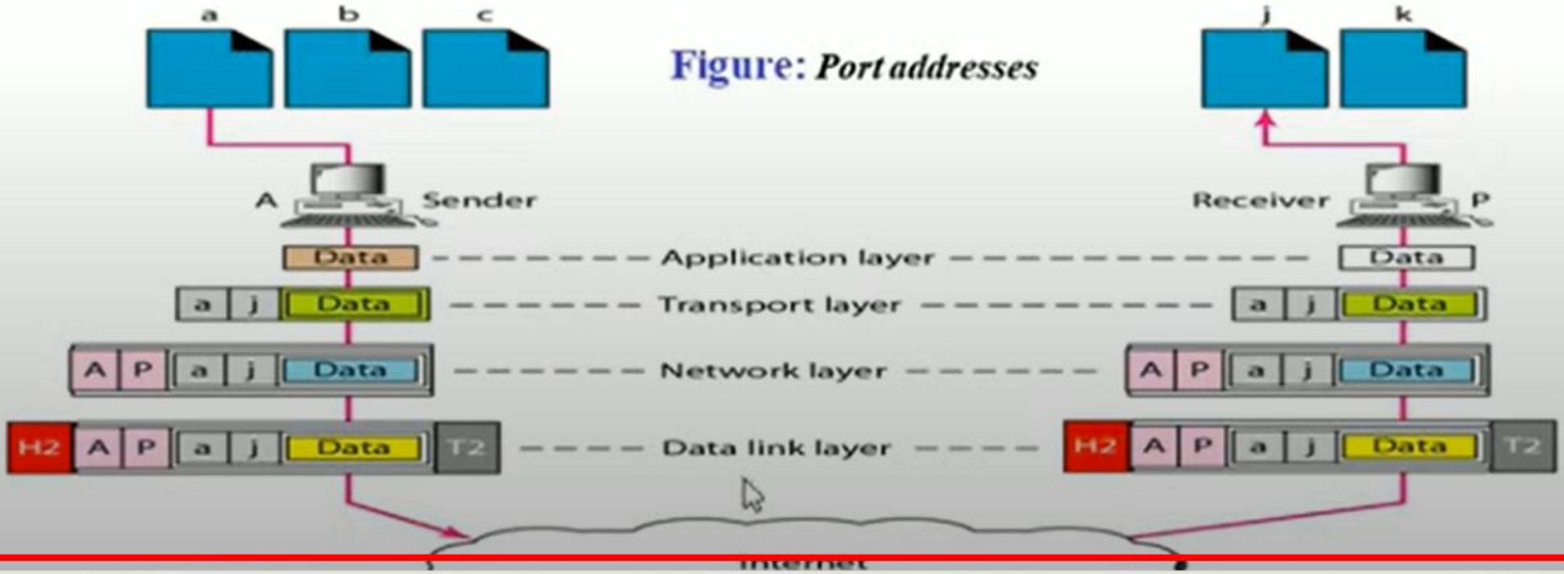


Figure: ***Port addresses***

4. Specific Addresses

- Specific address is associated with the Application layer.
- Some applications have user-friendly addresses that are designed for that specific address.
- Examples include the e-mail address (for example xyz456@yahoomail.com) and the URL - Universal Resource Locator (eg: www.google.com).

Error Detection and Correction in Data link Layer

Networks must be able to transfer data from one device to another with acceptable accuracy. Data can be corrupted during transmission. For reliable communication errors must be detected & corrected. Data-link layer uses error control techniques to ensure that frames (bit streams of data), are transmitted from the source to the destination with a certain extent of accuracy.

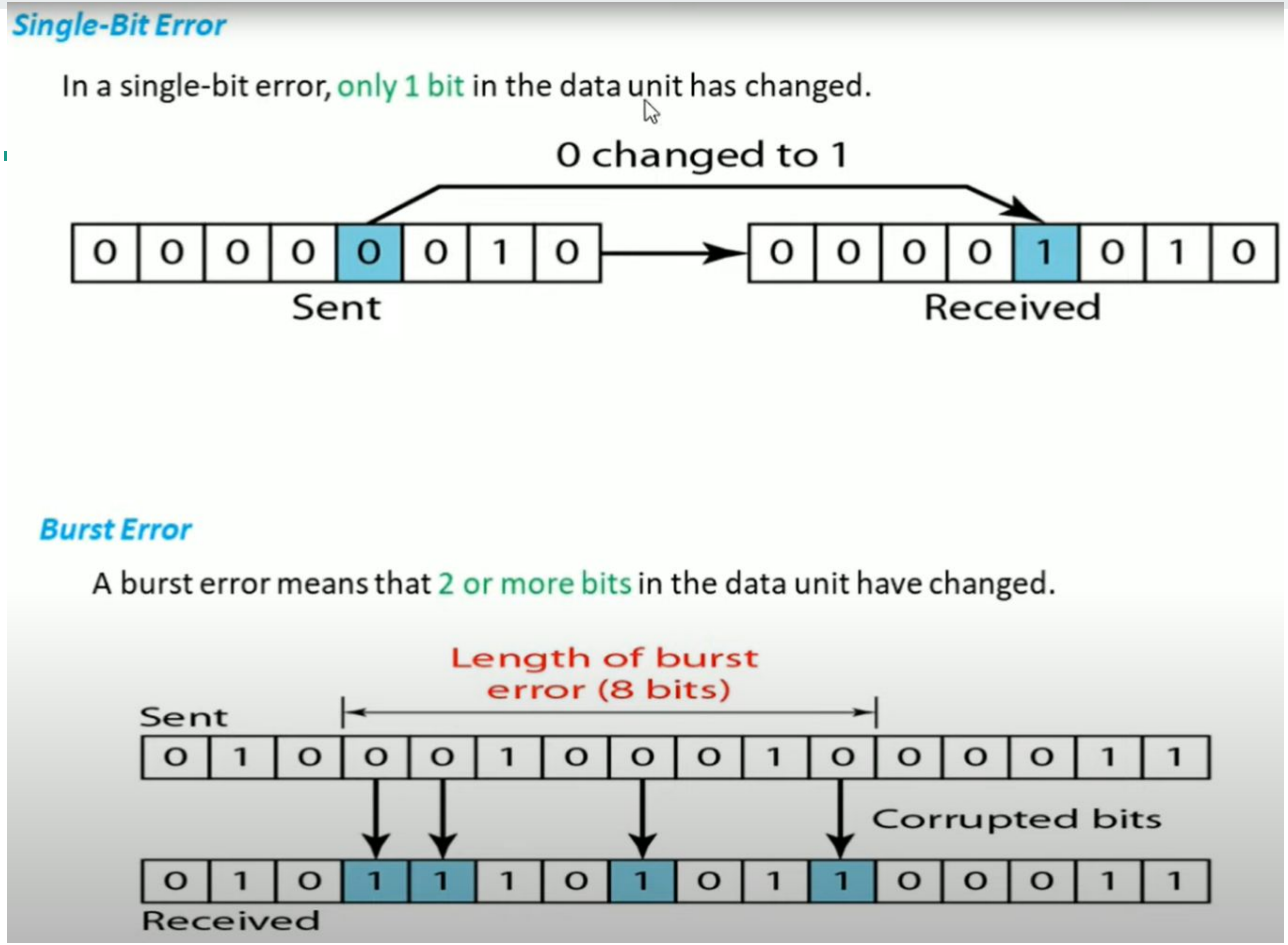
Errors

When bits are transmitted over the computer network, they get corrupted due to interference and network problems. Thus the data received at the destination will not be identical to the data transmitted by the sender. These corrupted bits received are invalid and are called errors. For example data sent → 11010010   and data received → 10010110

Types of Errors

Errors can be of two types, namely

- single bit errors,
- burst errors.



Single-Bit Error

In a single-bit error, only 1 bit in the data unit has changed.

Sent							
0	0	0	0	0	0	1	0

Received							
0	0	0	0	1	0	1	0

Burst Error

A burst error means that 2 or more bits in the data unit have changed.

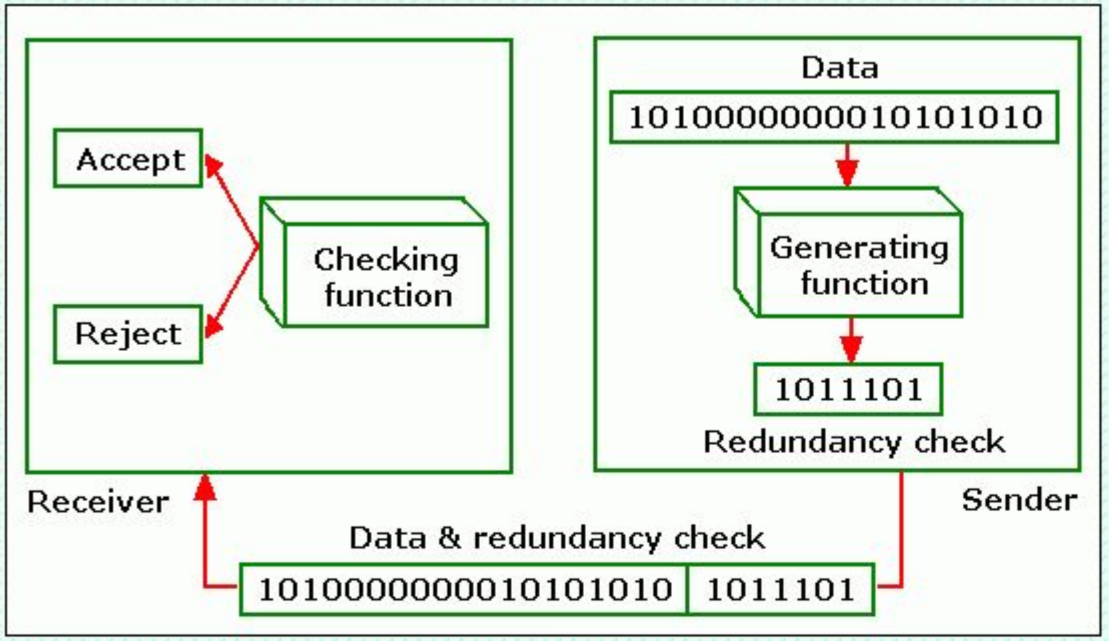
Sent	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1
Received	0	1	0	1	1	1	0	1	0	1	1	0	0	0	1	1

Burst error :

The term burst error means that two or more bits in the data unit have changed from 0 to 1 or vice-versa. Some bits in between may not have been corrupted. Burst errors are most common in data transmission. The length of Burst error is measured from the first corrupted bit to the last corrupted bit.

Redundancy

- The central concept in detecting and correcting errors is redundancy.
- For both error detection and error correction, the sender needs to send some additional bits along with the data bits. Adding extra bits to the data is called Redundancy.
- Therefore , Redundant bits are extra bits that are generated and added to the data bits. These extra bits help in error detection & correction during data transfer .
- When the destination receives the data, it performs necessary checks based upon the additional redundant bits. If the receiver finds that the data is free from errors, it removes the extra redundant bits before passing the message to the upper layers, else the data is rejected.



In the figure, the sender generates redundant bits, which are added to the data to be sent. Hence the original data with the extra bits are sent to the destination.
At the receiving end, error detection is done based on the redundant bits. Depending on the result of the checking function, the receiver will accept or reject the data.

Error Control : Error control can be done in two ways

- **Error detection** − Error detection involves checking whether any error has occurred or not. The number of error bits and the type of error does not matter.

- **Error correction** − Error correction is more complicated than error detection. Error correction involves finding the exact number of bits that has been corrupted and the location of the corrupted bits.

Error Detection Techniques

Error detection techniques are based on the use of redundancy bits. Some popular techniques for detecting errors in frames are:

The most popular Error Detecting Techniques are:

- **Single Parity Check or Simple Parity Check**
- **Two- dimensional Parity Check**
- **Checksum**
- **Cyclic Redundancy Check (CRC).**

1. Single Parity Check

Parity bits :

It is the bit added to the data to ensure that the total number of 1s in the data is even or odd. Parity bits are used for error detection. There are 2 types of parity bits -

Even Parity : If the no. of 1s in the given set of bits are even, then the parity bit is set as 0 else 1.

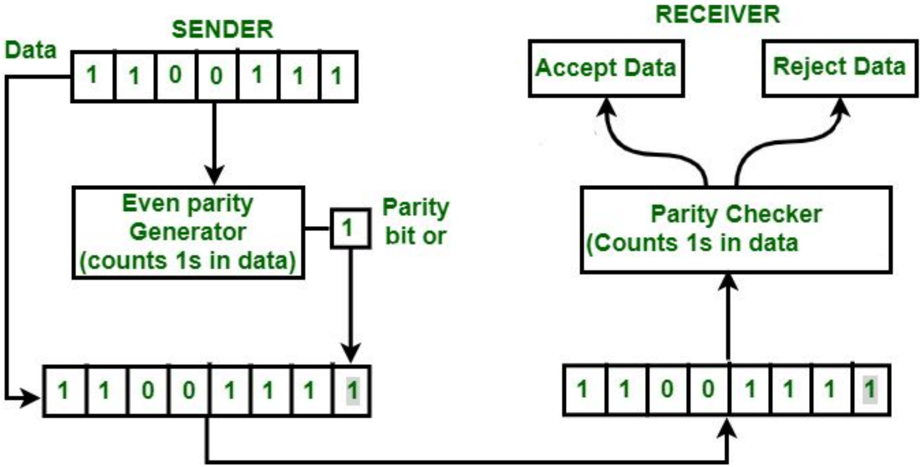
Odd Parity : If the no. of 1s in the given set of bits are odd, then the parity bit is set as 0 else 1.

- Single Parity checking is the simple mechanism to detect the errors.
- In this technique, a redundant bit is also known as a parity bit is added at the end of the data unit.
- The extra bit added to the original bits to make the number of 1s even in the case of even parity or odd in the case of odd parity.

Working of Simple/ Single Parity Check

- Simple parity check uses a parity generator and parity checker. The parity generator is used at the sender side and it adds parity bit with the data. The parity bits are transmitted along with the data unit.
- At the receiving end, a parity checker is used, which checks the parity bits. If the sent parity bits are equal to the received parity bits, then there is no error and receiving machine accepts the data.
- If the sent parity bits are not equal to the received, then it shows an error in the data and the data gets rejected.

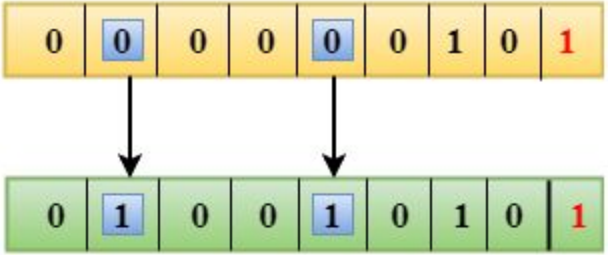
Single Parity Check



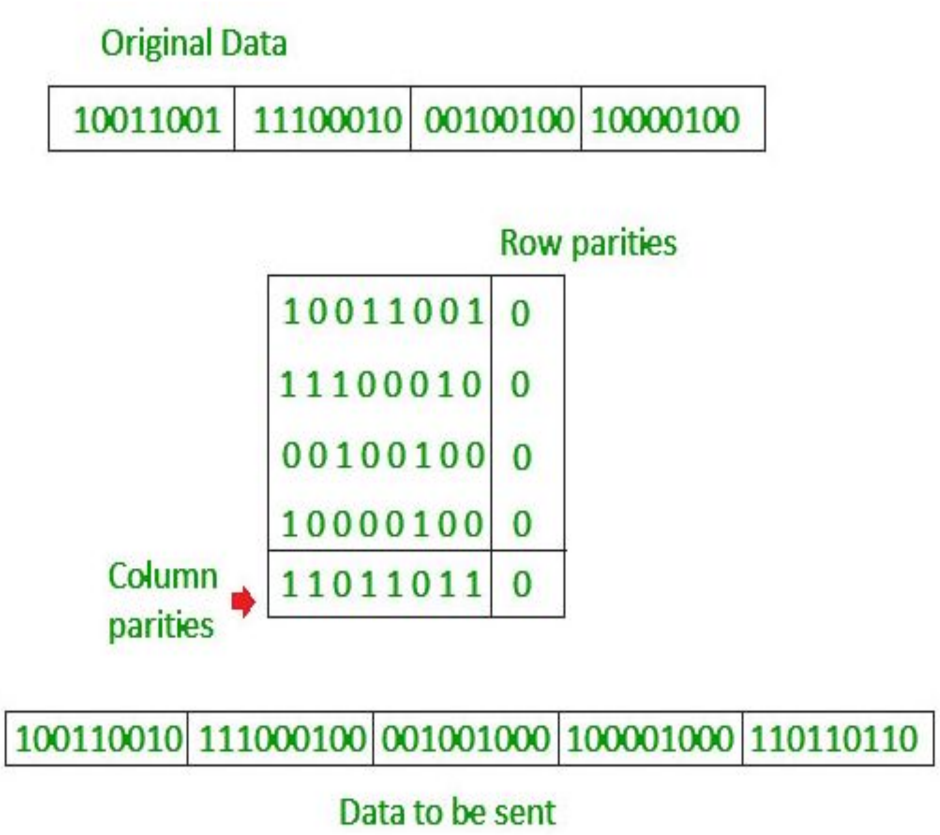
The above diagram shows the basic function of parity checking error detection.

Drawbacks Of Single Parity Checking

- It can only detect single-bit errors which are very rare.
- Cannot detect errors if the no. of bits corrupted is even.



2. Two-Dimensional Parity Check



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

- In this technique, the whole data is divided into block of bits and these blocks are organized in the form of a table.
- Parity check bits are calculated (using Single Parity check) for each row, and column.
- Row parities and column parities are calculated as shown in the figure.
- Parity bits are appended at the end of the data block and these parity bits are sent along with data.
- At the receiving end, these are compared with the parity bits calculated on the received data.
- **Two- Dimensional Parity Checking increases the chance of detecting burst errors.**

Drawbacks Of 2D Parity Check

- If two bits in one data unit are corrupted and two bits in exactly the same column position in another data unit are also corrupted, then 2D Parity checker will not be able to detect the error.
- This technique cannot be used to detect the 4-bit errors or more in some cases.

3. Checksum

- ➢ This error detection scheme is used in many protocols.
- ➢ The sender generates the checksum and sends the original data along with the checksum.
- ➢ The receiver end also generates the checksum from the received data.
- ➢ If the generated sum at the receiver side is all zeros then only the data is accepted.

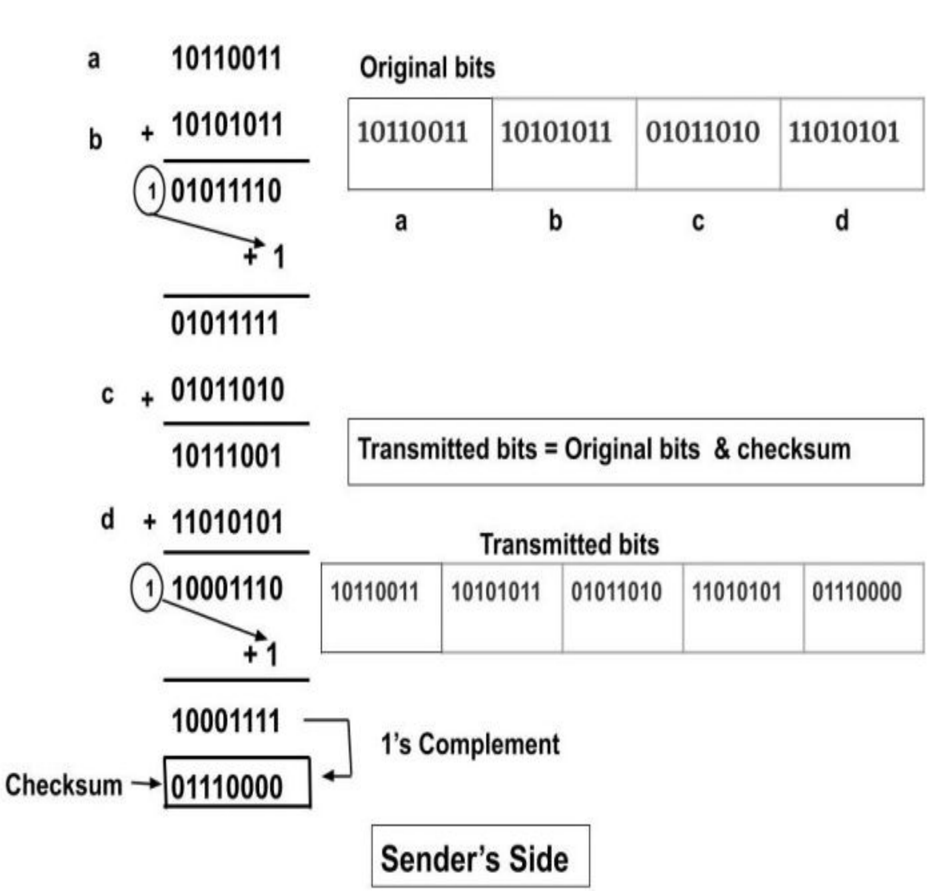
The sender follows these steps:

1. Data is divided into k blocks or sections of ‘n’ bits.
2. All the blocks are added using one's complement arithmetic.
3. The final sum is bitwise complemented (convert 0 to 1 and 1 to 0) to get the checksum.
4. The sender sends the original data along with the checksum.

The receiver follows the following steps:

1. Data is divided into k sections or blocks of given 'n' bits.
2. All the blocks are added using 1’s complement arithmetic.
3. The final sum is bitwise complemented (convert 0 to 1 and 1 to 0).
4. If the result is all zeros then it accepted else rejected.

Suppose we have to send 10110011 10101011 01011010 11010101. The whole 32 bit data is divided into a group of 8 bits i.e. 4 groups.



```
a     10110011
b   + 10101011
    ----------
  (1) 01011110
           + 1
    ----------
      01011111
c   + 01011010
    ----------
      10111001
d   + 11010101
    ----------
  (1) 10001110
           + 1
    ----------
      10001111   1's Complement
Checksum → 01110000
```

Original bits

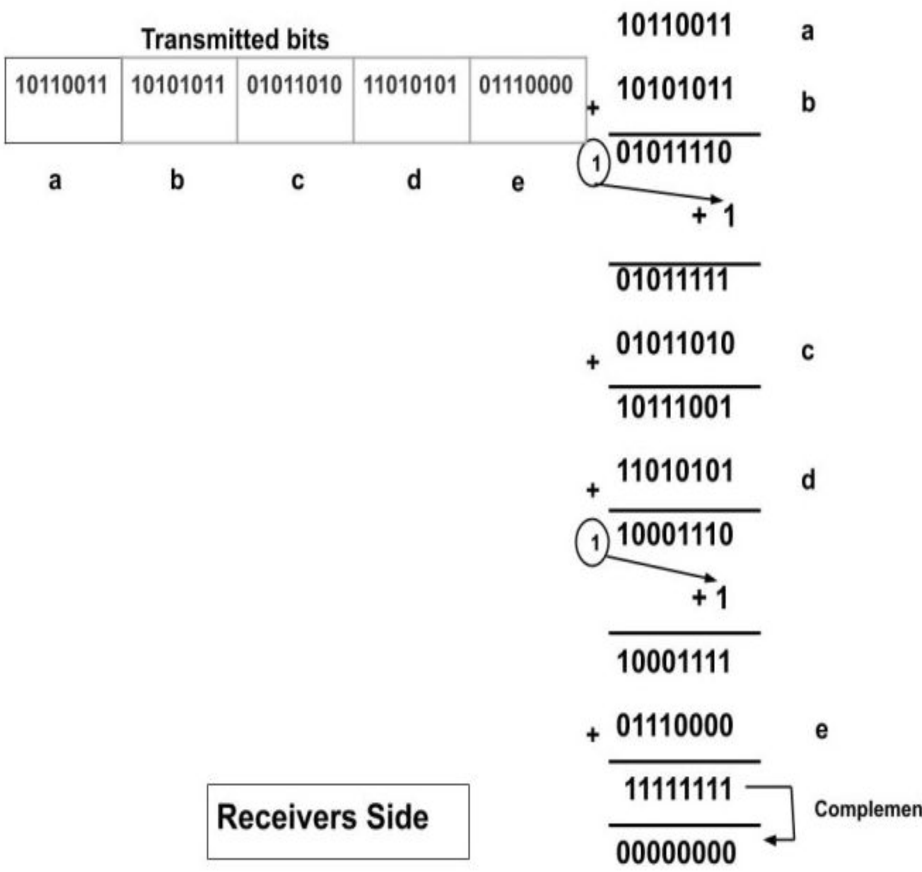
10110011	10101011	01011010	11010101
a	b	c	d

Transmitted bits = Original bits & checksum

Transmitted bits

10110011	10101011	01011010	11010101	01110000

Sender's Side



Transmitted bits

10110011	10101011	01011010	11010101	01110000
a	b	c	d	e

```
      10110011   a
    + 10101011   b
    ----------
  (1) 01011110
           + 1
    ----------
      01011111
    + 01011010   c
    ----------
      10111001
    + 11010101   d
    ----------
  (1) 10001110
           + 1
    ----------
      10001111
    + 01110000   e
    ----------
      11111111   Complemen
    ----------
      00000000
```

Receivers Side

Drawback of Checksum Error Detection Method

The main problem is that, if one or more bits of a segment is damaged and the corresponding bit / bits of opposite value in the second segment are also damaged, then the error goes undetected. This is because the sum of these columns remain unchanged.

Cyclic Redundancy Check (CRC)

- CRC is a powerful redundancy checking technique. It is a method of detecting accidental changes/ errors to the digital data , which is commonly used in digital networks & storage devices.

- **CRC is based on binary division**, where as checksum is based on binary addition.

- In CRC, a sequence of redundant bits are generated & appended to the end of data unit. It is also called the CRC remainder.

- CRC involves binary division of the data bit unit being sent, by a predetermined divisor agreed upon by the communicating system. The divisor is generated using polynomials. So CRC is also called **Polynomial code checksum.**

- The sender performs binary division of the data bits by the divisor. It then appends the remainder called **CRC bits** to the end of the data segment. This makes the resulting data unit exactly divisible by the divisor.

- At the destination, the receiver divides the incoming data units by the divisor. If there is no remainder, the data unit is correct & is accepted else it is corrupted and is rejected.

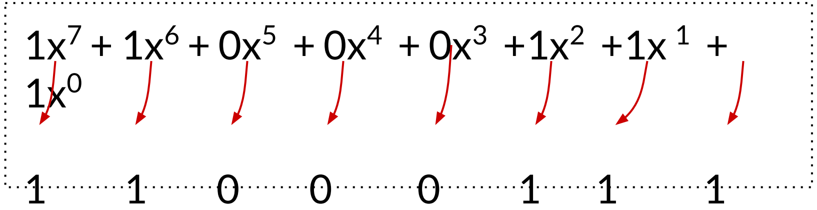
- **CRC is the most powerful error detection method used in the real life applications.**

CRC uses Generator Polynomial which is available on both sender and receiver side.

CRC Generator is an algebraic polynomial represented as a bit pattern. Bit pattern is obtained from CRC generator.

Consider the CRC generator $x^7 + x^6 + x^2 + x + 1$.

The corresponding binary pattern obtained is 11000111



$1x^7 + 1x^6 + 0x^5 + 0x^4 + 0x^3 + 1x^2 + 1x^1 + 1x^0$

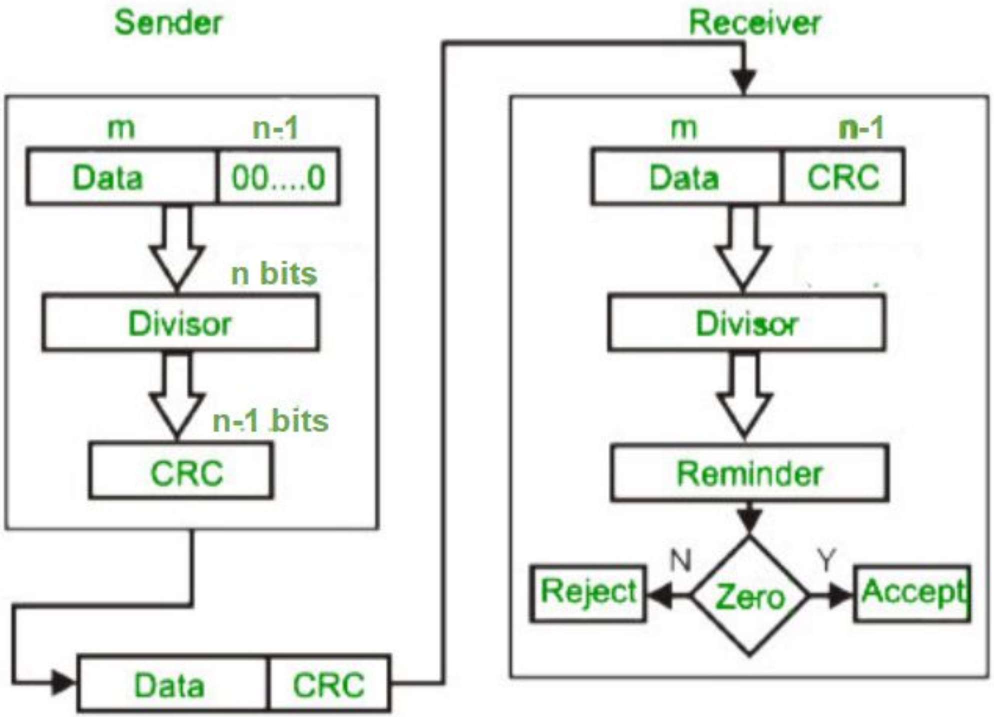
1 1 0 0 0 1 1 1

Another example : $x^3 + x + 1$, the binary pattern obtained is 1011

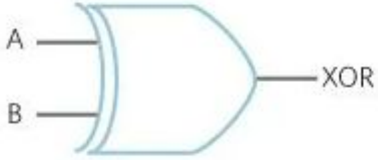
$x^2 + 1$, binary pattern obtained is 101

CRC Error Detection Steps :

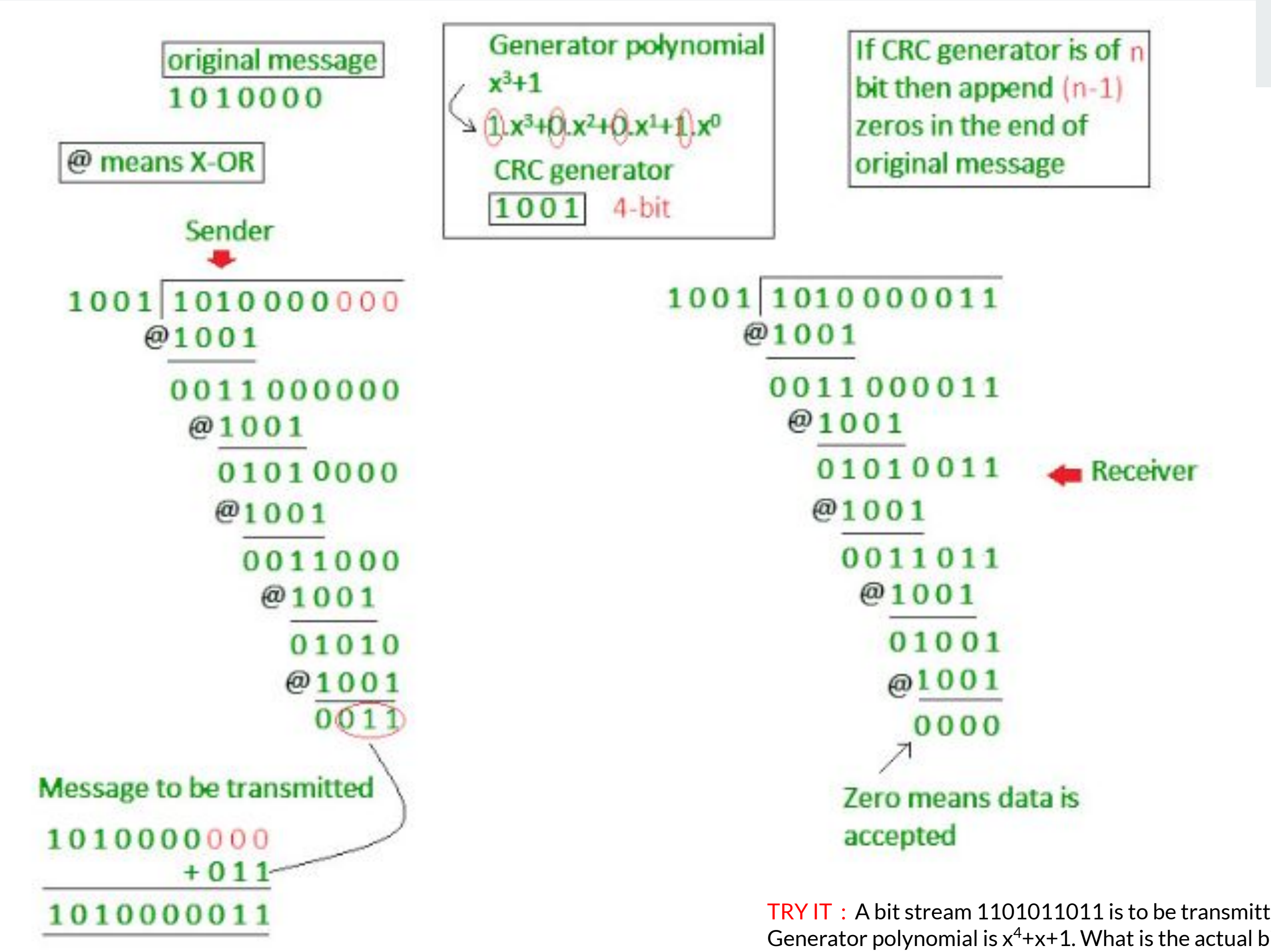
1. Calculation of CRC at Sender side
   1. Append n-1 zeroes to the end of the data unit to be transmitted, where n= no. of bits in the divisor [divisor is the bit pattern given by CRC generator]
   2. Using binary division, divide the above resultant string with the divisor.
   3. After division, the remainder obtained is called **CRC remainder.**
   4. Append CRC remainder to the end of the data unit. Actual data bits plus the remainder is called a **codeword.**
2. The newly formed codeword is transmitted to the receiver.
3. At the receiver side, the transmitted codeword is received. The code word is divided with the same divisor, ie the bit pattern given by CRC generator.
4. On division, if the remainder is zero, then there is no errors & the data is accepted. If it is non-zero, data bits are rejected



$$X = A \oplus B$$



A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0



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

```
1001 | 1010000000
      @1001
       0011000000
        @1001
         0101000
         @1001
          0011000
           @1001
            01010
            @1001
             0011
```

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

Zero means data is accepted

Message to be transmitted

```
1010000000
     +011
----------
1010000011
```

TRY IT : A bit stream 1101011011 is to be transmitted using the standard CRC method. Generator polynomial is x^4+x+1. What is the actual bit string transmitted?

CRC Advantages

- CRC is the most powerful error detection method used in real life applications
- It can detect all single bit errors, double bit errors and most burst errors.
- CRC methods are popular because they are simple to implement in binary hardware, easy to analyze mathematically and good at detecting common errors caused   by noise in transmission channels.
- Because of its simplicity, it can fit to any type of Operating systems including the latest ones.

CRC Disadvantages

- CRC is an easy to crack mechanism. Therefore not suitable for high security purpose.

Error Correction

Error-correcting codes (ECC) are a sequence of numbers generated by specific algorithms for detecting and removing errors in data that has been transmitted over noisy channels. Depending on the algorithm, Error correcting codes determine the exact number of bits that has been corrupted and the location of the corrupted bits

Error correction can be handled in 2 ways

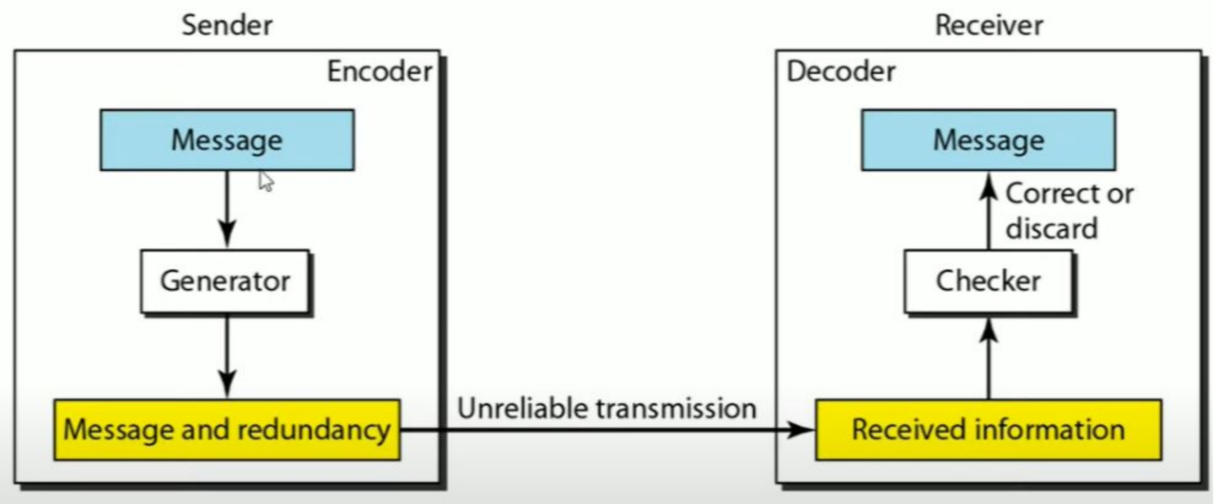
1. Correction by Retransmission / Backward Error Correction Technique : Once an error is found, the receiver can request the sender to retransmit the entire dataunit. This technique is simple & inexpensive in the case of wired transmission like fiber optics; but expensive in the case of wireless transmission.
2. Forward Error Correction Technique : In this case the receiver uses the error correcting code that automatically corrects the errors. In this error correction method, the receiver tries to guess the message using the redundant bit. This saves the band width required for retransmission. But if there are too many errors, frames need to be retransmitted.

Error correction codes can be broadly categorized into 2 types

i. Block Codes
ii. Convolution Codes

Coding

- The main concept in the detection & correction of data transmission errors is redundancy.
- **Redundancy is achieved through various coding schemes.**
- The sender adds redundant bits through a process using algorithms, that creates a relationship between the redundant bits and the data bits. The receiver checks the relationships between the two set of bits to detect errors



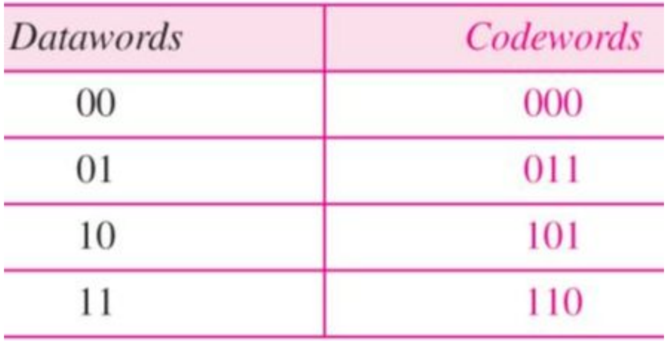
Coding scheme can be broadly be divided into two broad categories

- Block Coding
- Convolution coding

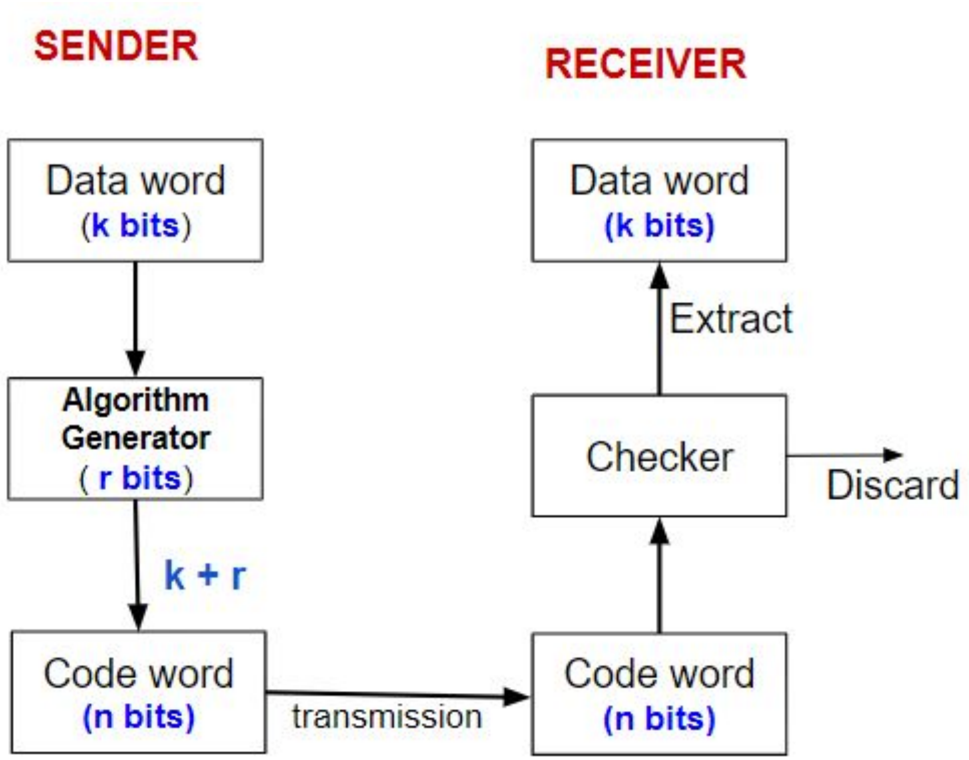
Block coding

- In block coding, we divide our message into blocks, each of **k** bits, called **data words**.
- We add **r** redundant bits to each block to make the length **n = k + r**. The resulting n-bit blocks are called **code words**.
- ie We have a set of data words, each of size k, and a set of code words, each of size of n. With k bits, we can create a combination of 2^k data words, with n bits; we can create a combination of 2^n code words. Since n > k, the number of possible code words is larger than the number of possible data words.This means that we have 2^n-2^k code words that are not used. These code words are invalid or illegal.
- If the receiver receives an invalid codeword, this indicates that the data got corrupted during transmission.

Suppose message to be sent :
00011011 , k=2, r=1 ∴ n=3



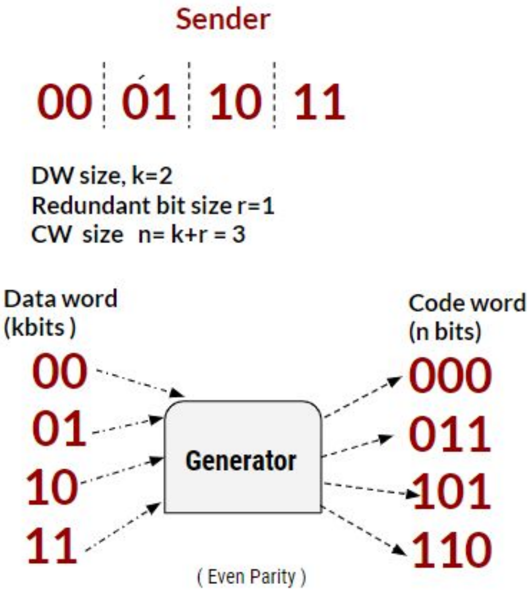
Datawords	*Codewords*
00	000
01	011
10	101
11	110



The sender creates codewords out of data words using a generator application that applies an algorithm. These codewords are sent to the receiver through the transmission media. If the received codeword is same as one of the valid codewords, it is accepted. The corresponding data word is extracted for use. Due to distortion in transmission media, the received codeword may change. If the received codewords does not match with the valid codewords, it is discarded.

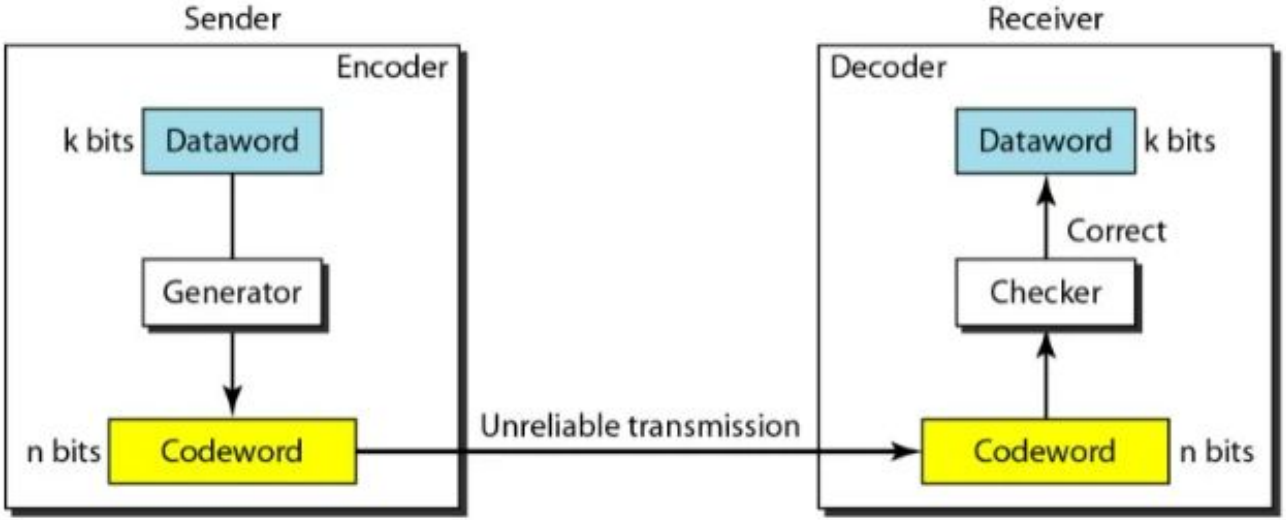
Eg: Suppose the message to be transmitted from the sender to the receiver is 00011011

- In block coding, first the message is divided into blocks of k bits
- Imagine if k=2, then 4 blocks or 4 datawords(DW) are created.
- These DWs are to be transmitted to the destination.
- To detect errors, redundant bits are added by generator
- Let the generated redundant bit size be 1, r=1, ∴ CW size, n= k+ r = 3
- CWs are created by adding redundant bit to the DWs,
- These CWs are send one after the other thru the transmission media.
- The received CWs are forwarded to the checker.
- The checker compares these CWs with valid CWs [here 8 combinations are possible; 4 shown in figure are valid, rest are invalid]
- If it is a valid CW, the redundant bit is removed and DW is extracted for use.
- If the received CW does not match with any of the valid CWs, the received CW is corrupted, which is either discarded or corrected. It depends on the application.



DW size, k=2
Redundant bit size r=1
CW size n= k+r = 3

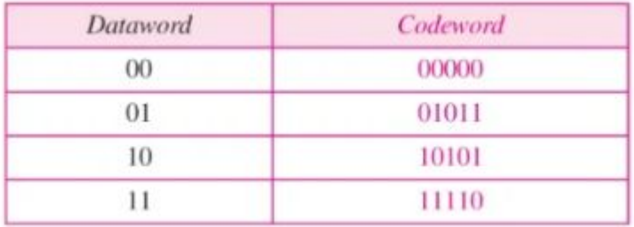
Error Correction in Block Coding



- **Example:**
  Assume that k = 2 and r = 3
  n = 5

Dataword	*Codeword*
00	00000
01	01011
10	10101
11	11110

Error Correction: Example



Dataword	Codeword
00	00000
01	01011
10	10101
11	11110

- Assume the dataword is 01. The sender creates the codeword 01011. The codeword is corrupted during transmission, and 01001 is received. First, the receiver finds that the received codeword is not in the table. This means an error has occurred. The receiver, assuming that there is only 1 bit corrupted, uses the following strategy to guess the correct dataword

1. *Comparing the received codeword with the first codeword in the table (01001 versus 00000), the receiver decides that the first codeword is not the one that was sent because there are two different bits. (the same for third or fourth one in the table)*
2. *The original codeword must be the second one in the table because this is the only one that differs from the received codeword by 1 bit.*

Drawbacks of Block coding

If the received CW gets corrupted in 2 bits[eg: 011 is received as 000], the checker compares to find whether it is valid or not. Since 000 is a valid CW, the checker treats it as valid, even though error has occurred in 2 bits. This is an undetectable error in this coding scheme.

Hence this coding mechanism is capable of identifying only single-bit errors.

Linear Block Codes

- **A linear block code is a code in which the eXclusive OR(XOR) of two valid CWs creates another valid CW.**
- The code in the table below is a linear block code, because the result of XORing any codeword with any other CW gives a valid CW

Eg:

Data word	Code word
00	000
01	011
10	101
11	110

0 0 0 0 1 1 ⊕ 0 1 1	0 1 1 1 0 1 ⊕ 1 1 0
1 1 0 0 1 1 ⊕ 1 0 1	1 0 1 1 1 0 ⊕ 0 1 1

Linear Property : **If Ci and Cj are two CWs, then XORing them, Cp = Ci ⊕ Cj, Cp should also be a valid CW.**

- **Almost all Block Codes used today belong to a subset of block codes called Linear Block Codes.**
- The use of non-linear block codes for error detection & correction is not wide spread because of its difficulty in its implementation.

Hamming Code

Hamming code is a Linear block code that is capable of **detecting up to two simultaneous bit errors and correcting single-bit errors**. It was developed by R.W. Hamming for error correction.

- In this coding method, the source encodes the message by inserting redundant bits within the message.
- These redundant bits are extra bits that are generated and inserted at specific positions in the message itself to enable error detection and correction.
- When the destination receives this message, it performs recalculations to detect errors and find the bit position that has error.

Sender Side : Encoding a message in Hamming Code

The procedure used by the sender includes the following steps-

Step 1 – Calculation of the number of redundant bits.
Step 2 – Positioning the redundant bits.
Step 3 – Calculating the values of each redundant bit.

Step 1 – Calculation of the number of redundant bits.

Let the no. of data bits=**m** and & no. of redundancy bits = **r**.

Therefore total no. of bits to be sent = **m + r**

The value of **r** must satisfy the following relation : $\mathbf{2^r >= m+r+1}$

Suppose the data to be sent is **1001,** then the number of data bits , m =4,

And the number of redundant bits can be calculated using

$\mathbf{2^r >= m+r+1}$

When r=1 , 2^1 >=4+1+1 - False

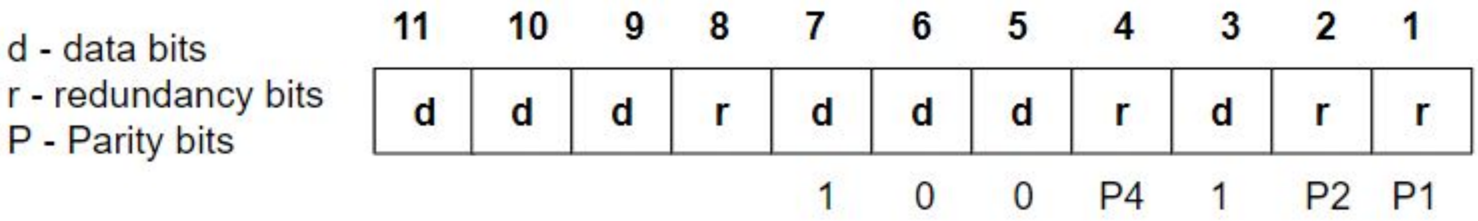
When r=2, 2^2 >=4+2+1 - False

When r=3, 2^3 >=4+3+1 - **True**

No. of redundant bits, r = 3 bits

Step 2 – Positioning the redundant bits.

The r redundant bits are placed at bit positions which are powers of 2, ie 1,2,4,8,16,...(2^0, 2^1, 2^2, 2^3,...)

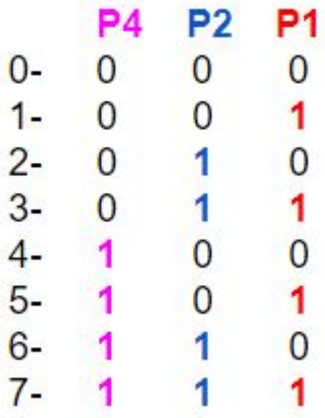


d - data bits
r - redundancy bits
P - Parity bits

11	10	9	8	7	6	5	4	3	2	1
d	d	d	r	d	d	d	r	d	r	r
				1	0	0	P4	1	P2	P1

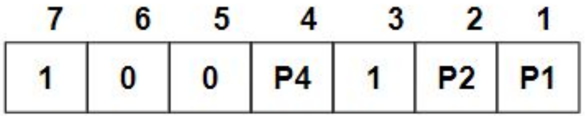
Therefore, these redundant bits/Parity bits here the 3 bits should be positioned at P4, P2, P1

Step 3 – Calculating Parity bits.

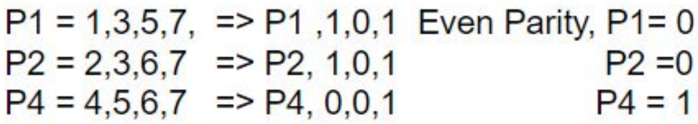


	P4	P2	P1
0-	0	0	0
1-	0	0	1
2-	0	1	0
3-	0	1	1
4-	1	0	0
5-	1	0	1
6-	1	1	0
7-	1	1	1

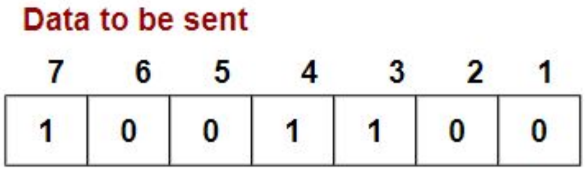
- Parity bits are calculated based upon the data bits & redundant bits.
- Parity bit **P1** is calculated using parity check at all the bit positions whose binary representations include a 1 in the Least Significant Bit position. **P1** - 1,3,5,7,9, 11,,,...
- Parity bit **P2** is calculated using parity check at all the bit positions whose binary representations include a 1 in the second position from the Least Significant Bit position. **P2** - 2, 3, 6, 7, 10,11,,,...
- Parity bit **P4** is calculated using parity check at all the bit positions whose binary representations include a 1 in the third position from the Least Significant Bit position. **P4** - 4, 5, 6, 7, 12, 13, 14, 15,,,... and so on..
- Let the data be sent with **Even Parity**



7	6	5	4	3	2	1
1	0	0	P4	1	P2	P1



P1 = 1,3,5,7, => P1 ,1,0,1 Even Parity, P1= 0
P2 = 2,3,6,7 => P2, 1,0,1 P2 =0
P4 = 4,5,6,7 => P4, 0,0,1 P4 = 1



Data to be sent

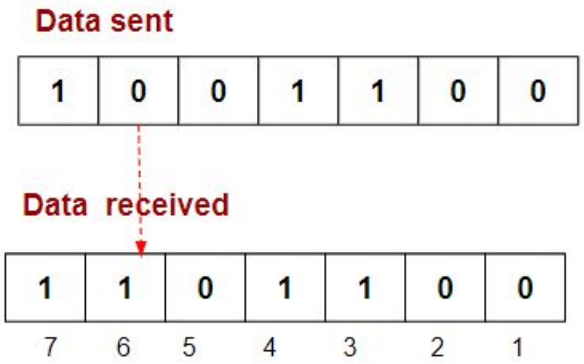
7	6	5	4	3	2	1
1	0	0	1	1	0	0

Receiver Side : Decoding a message in Hamming Code

Once the receiver gets an incoming message, it performs recalculations to detect errors and correct them. The steps for recalculation are –

- **Step 1 – Calculation of the number of redundant bits** : Using the same formula as in encoding, the number of redundant bits are determined.
- **Step 2 – Positioning the redundant bits** : The *r* redundant bits placed at bit positions of powers of 2, i.e. 1, 2, 4, 8, 16 etc..
- **Step 3 – Parity checking :** Parity bits are calculated based upon the data bits and the redundant bits using the same rule as during generation
- **Step 4 – Error detection and correction** :The decimal equivalent of the parity bits binary values is calculated. If it is 0, there is no error. Otherwise, the decimal value gives the bit position which has error.

Suppose the Received data has a single bit error.



Data sent

1	0	0	1	1	0	0

Data received

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

P1 = 1,3,5,7, => 0 ,1,0,1 Even Parity, P1= 0
P2 = 2,3,6,7 => 0,1,1,1 P2 =1
P4 = 4,5,6,7 => 1,0,1,1 P4 = 1

If all parity bits are Zero, there is no error.
Here P2 and P4 are not zeroes

P4	P2	P1
1	1	0

Decimal Equivalent = 6
Therefore 6th bit is error, and its value should be flipped.

Hamming Distance

Hamming distance is a method for comparing two binary data sequences. The Hamming distance between two codewords is simply the number of bit positions in which they differ.

The hamming distance between two binary sequences v1 and v2 is denoted as d(v1,v2). **The hamming distance d(v1,v2) between two n-bit binary sequences v1 and v2 is the number of bits in which v1 and v2 disagree.**

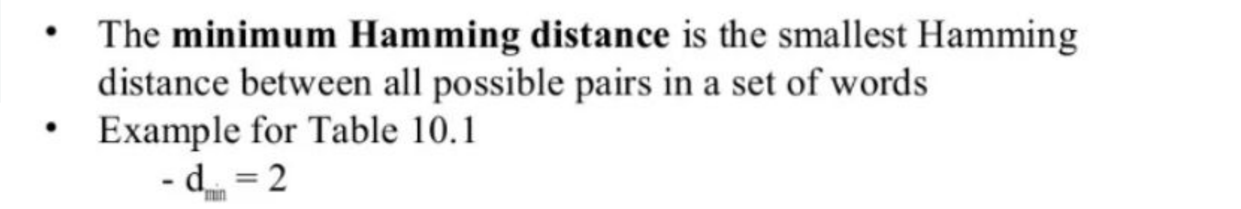
Eg: If v1=11001 and v2=11110, then d(v1,v2)= 3

Hamming distance can easily be found if we apply the XOR operation ⊕ on the two strings and count the no. of 1s in the result. Note that the hamming distance is a value >=0

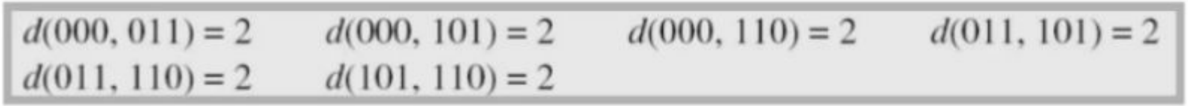
eg: 1. d(000,011)= $\begin{array}{l} 000 \oplus \\ \underline{011} \\ \mathbf{011}, \end{array}$ there are two 1s, ie there are 2 errors

2. d(10101, 11110) = $\begin{array}{l} 10101 \oplus \\ \underline{11110} \\ \mathbf{01011} \end{array}$ there are three 1s, ie there are 3 errors

The hamming distance between the received codeword and the sent codeword is the no. of bits that are corrupted during transmission.



- The **minimum Hamming distance** is the smallest Hamming distance between all possible pairs in a set of words
- Example for Table 10.1
  - $d_{min} = 2$



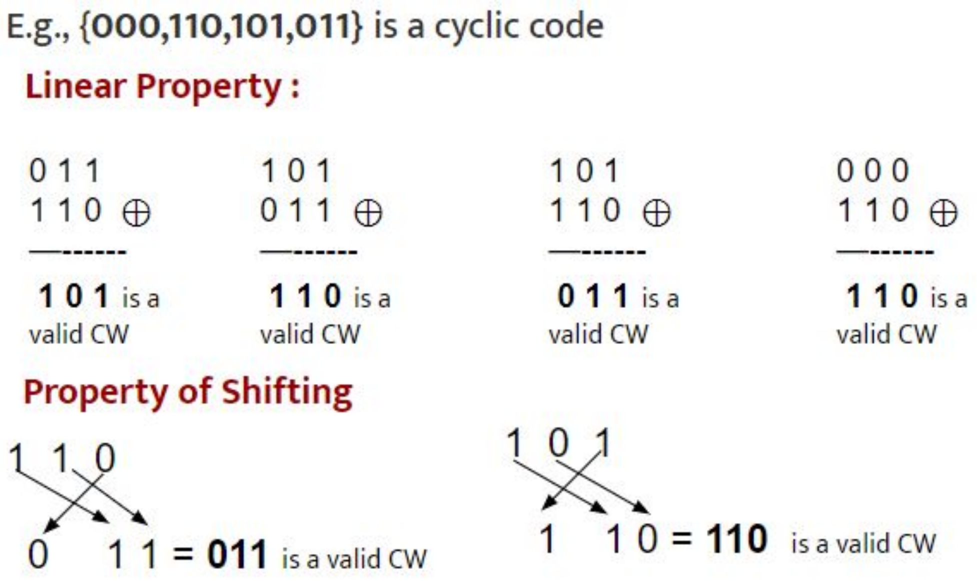
$d(000, 011) = 2$	$d(000, 101) = 2$	$d(000, 110) = 2$	$d(011, 101) = 2$
$d(011, 110) = 2$	$d(101, 110) = 2$		

- **Hamming distance can be used to detect how many errors can be detected and how many errors can be corrected.**
- The maximum number of errors that can be detected is, $t=d_{min}-1$
- If a code satisfies $d_{min}>=(2t+1)$, then the code can correct up to t bit errors.

Eg: Consider the codewords
00000
00111
11001
11110

On calculating the hamming distance between all pairs of combinations, we get
d(00000,00111)=3, d(00000,11001)=3, d(00000,11110)=4,
d(00111,11001)=4, d(00111,11110)=3,d(11001,11110)=3
Here $d_{min}=3$,
No. of errors that can be detected, $t= d_{min}-1 =3-1 = 2$
To find No. of errors that can be corrected using formula, $d_{min}>=(2t+1)$
3>=2t+1, t=1 satisfies ∴ No. of errors that can be corrected=1

Cyclic Codes



E.g., {000,110,101,011} is a cyclic code

Linear Property :

0 1 1	1 0 1	1 0 1	0 0 0
1 1 0 ⊕	0 1 1 ⊕	1 1 0 ⊕	1 1 0 ⊕
——------	——------	——------	——------
1 0 1 is a valid CW	**1 1 0** is a valid CW	**0 1 1** is a valid CW	**1 1 0** is a valid CW

Property of Shifting

1 1 0

0 1 1 = **011** is a valid CW

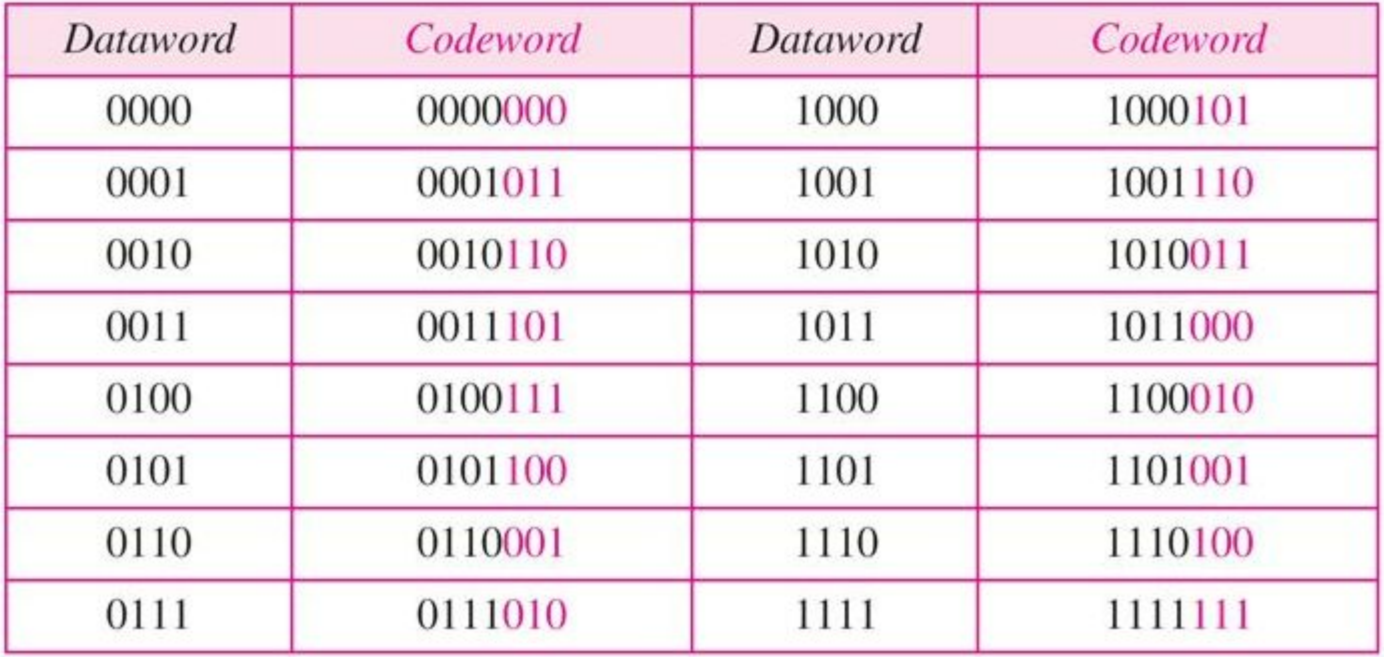
1 0 1

1 1 0 = **110** is a valid CW

Advantages of Cyclic Codes: : The cyclic codes have a very good performance in detecting single-bit errors, double errors, an odd number of errors, and burst errors. They can easily be implemented in hardware and software. They are especially fast when implemented in hardware. This has made cyclic codes a good candidate for many networks.

- Cyclic codes are special linear block codes with one extra property.
- In a cyclic code, if a code word is cyclically shifted (rotated), the result is another code word.
- Property of Shifting : After shifting left or right by any number of bits, the resultant should be a valid codeword.
- Linear Property : eXclusive OR (Modulo 2 Addition) of 2 valid codewords creates another valid codeword.
- CRC Cyclic Redundancy Check- One of the categories of cyclic codes called the cyclic redundancy check (CRC) is used in networks such as LANs and WANs.

The following table shows an example of a CRC code which shows both the linear and cyclic properties of this code.



Dataword	*Codeword*	*Dataword*	*Codeword*
0000	0000000	1000	1000101
0001	0001011	1001	1001110
0010	0010110	1010	1010011
0011	0011101	1011	1011000
0100	0100111	1100	1100010
0101	0101100	1101	1101001
0110	0110001	1110	1110100
0111	0111010	1111	1111111