APPLICATION LAYER

Notes by : Samundar singh

Principles of Network Applications:

1. Network Application Architecture:

It is designed by the application developer and dictates how the application is structured over the various end systems. In choosing the application Architecture the application developer will choose any of the two Architecture:

a.) Client-server Architecture b.) P2P Architecture

Client-server Architecture:

In this architecture there is an always-on host, called the ***server***, which services requests from many other hosts, called ***clients*****.** In this architecture clients do not directly communicate with each other and the servers has fixed with a well known address known as IP address and the server is always ON. The client can always contact the server by sending a packet to the server's IP address. In a client server Architecture there is no single-server host since single-server host is incapable of keeping up with all the requests from clients so a **data center** having a large number of host is used.

P2P Architecture:

in peer-2-peer architecture there is minimal (or no) reliance on dedicated servers in data centers, the application direct communication between pairs of internal connected hosts, called *peers*. These peer helps to communicate between desktop without server hence they are called peer to peer Architecture. The most important feature of the P2P Architecture is their **self-scalability**

2. Process Communicating:

Here we learn how the process communicatee with each other in a network ?

Process on two different end system communicate with each other by exchanging the message between computer network

Overview of client and server process: A network application consists **of pairs of** processes that send messages to each other over a network. The process that initiates the communication is called **client** and the process that waits to be contacted to begin the session is called **server.**

Any message sent from one process to another must go through the underlying network. A process sends messages into, and receives messages from, the network through a **software interface** called a **socket.** so socket is a software interface between the application layer and transport layer within host that help the process to communicate. Also called **API** between application and network

3. Addressing Process:

How the data transfer between the process running on different Host?

The process running in on one host send data packets to a process running on another host via means of an Address. When a host(*sender*) have to send some data then it send data on an Address, the host on the another side (receiver) identify the receiving process by the address of the host (sender). The host is identified by its IP address

4. Transport Service Available to Application:

When a process send some data then the transport-layer protocol has the responsibility of getting the messages to the socket of the receiving process. There is more than one transport-layer protocol. So for communication we must have to make a choice between then according to our requirement.

The transport services provide these services to application:

a.) **Reliable data transfer:** The packet may get lost in a computer network due to some reasons like overflow of buffer. So it must ensure that the data deliver as whole or the process terminates and again reload.

b.) **Throughput:** it is a rate at which the sending process can deliver bits to the receiving process. Since other session can also share the same bandwidth and these sessions will be coming and going thus the throughput can fluctuate with time. the application could request a guaranteed throughput of *r* bits/sec, and the transport protocol would then ensure that the available throughput is always at least *r* bits/sec.

c.) **Timing :** the transport layer protocol can provide the timing guarantee. long delay in communication will result in starvation of data or may be some program have to be terminated due to long delay.

d.) **Security :** A transport protocol will encrypt all data transmitted by the sending process, and in the receiving host, the transport-layer protocol can decrypt the data before delivering the data to the receiving process.

Network Architecture Model:

1. OSI reference Model 2.TCP/IP Model

OSI reference Model:

The OSI model has seven layers:

Physical Layer: it will transmit raw bits over communication channel

Data link Layer: it will transform the raw transmitted into a line it break the input data into data frames and transmit the data frames sequentially

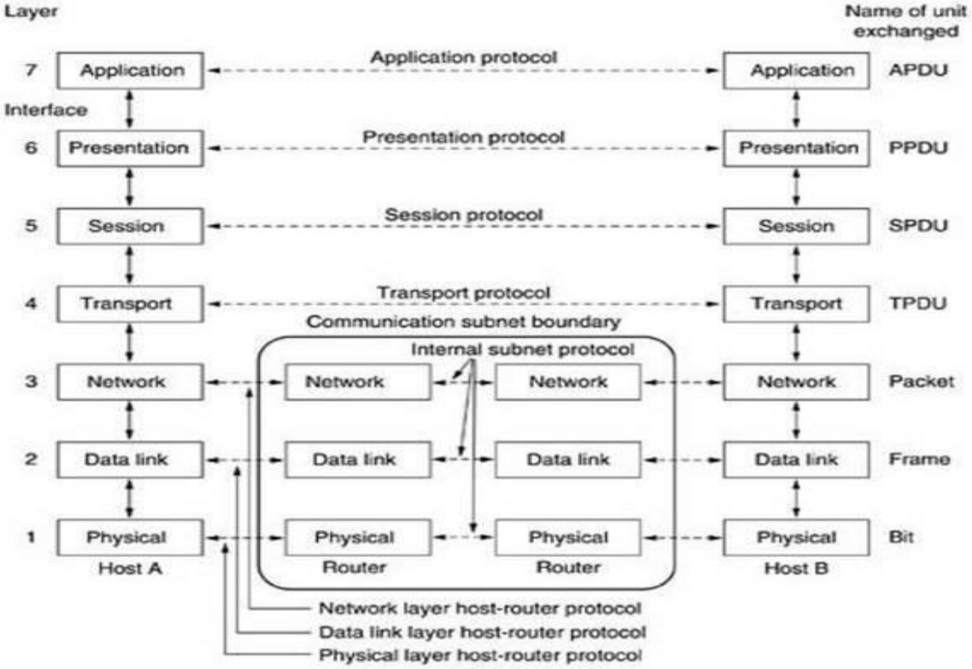
Network Layer: it will control the operation of the subnet and responsible for the routing of the packet from source to its destination

Transport Layer: the transport layer accept data from above ,split it up into smaller units and pass these to the network layer and also ensure that packets arrive correctly at other end. It also determine the services that is provided to the session layer

Session Layer: it offers various services like:
<u>dialog control</u>: keeping track of whose turn it is to transmit
<u>token Management</u>: preventing two parties from attempting the same critical operation at the same time
<u>Synchronization</u>: check pointing long transmissions to allow them to continue from where they were after a crash

Presentation Lyr: The presentation layer is concerned with the syntax and semantics n of the information transmitted

Application Layer: The application layer contains a variety of protocols that are commonly needed by users



TCP/IP Model:

The TCP/IP model has four layers:

Host to Network Layer: in this layer the host connect to the network using some protocol so that it can send some IP packet to it.

Internet Layer: the main job of this layer is to permit host to inject packets into any network and have travel independently to the destination. The internet layer defines a protocol known as internet **protocol (IP).**

Transport Layer: here two end-to end transport protocol is defined i.e. **TCP and UDP**

TCP Protocol: Transmission control protocol allows a byte stream originating on one machine to be delivered without error on any other machine. It fragments the incoming byte stream into discrete message and passes each message on the internet layer. The same process have done at the destination side where the received message is considered as the output stream.

UDP Protocol: User Datagram Protocol is an unreliable connectionless protocol for application here TCP sequencing or flow control have low priority. It is widely used for those condition or in application where delivery is more important to the accurate delivery. For example speech and video transmission.

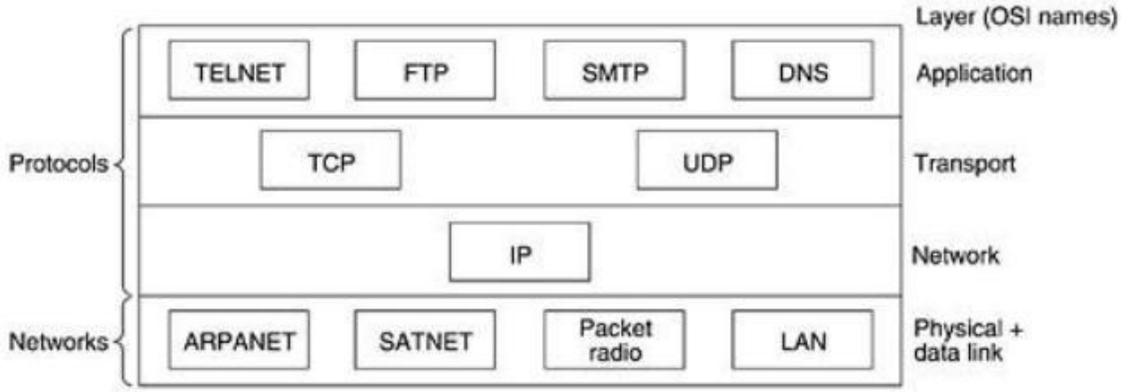
Application Layer: At the top of this layer there is application layer ,session and presentation layer is absent here. It contain all higher-level protocol that are:

TELENET Protocol: it is also known as the virtual terminal which allow user on one machine to log onto a distant machine and work there

FTP Protocol: File Transfer Protocol helps to transfer data from one place to another

SMTP Protocol: it also transfer file (generally used for Email).

HTTP Protocol: used in webpage view and streaming media as well.



Notes by : Samundar singh

TCP/IP Protocol Architecture:

1. DNS(domain name system):

The Domain Name System (DNS) is a hierarchical, distributed database system used to map between host names and IP addresses for various Internet applications. It basically translates the name to an address and vice-versa. Computer hosts connected to the network need a way to recognize each other. In addition to the binary IP address, each host may also register an ASCII string as its host name. which specify the country, street, city etc information Hosts under the DNS have a unique *domain name.* Each DNS *name server* of a *domain* maintains its own database of name-to-address mappings so that other systems (clients) across the Internet can query the name server through the DNS protocol messages.

There are several top-level domain like .com, .in, .gov, .org etc the domain name is divided in these categories for better scalability each domain represent a particular service or meaning.

Zone: it is the subset of the domain means a domain may contain various zone example

Name server: it is a host that contain a database built with zone data files and a resolution program to answer DNS queries.

Upon receiving a DNS query, which gives a domain name and requests its corresponding IP address, the name server looks up its database for the answer. The server replies to the client if there is a lookup match; otherwise, it performs further lookups in other name servers. The zone data file inside the name server have several records known as **RESOURCE RECORD.**

There are 6 types of Resource Record that describe the various aspect of domain name and they are:-

a.) **Start Of Authority(SOA):** it identifies the authority of that zone

b.) **Address(A):** this resource record is used for matching domain names to IP addresses as requested by the *forward query* Since multi-homed hosts have multiple network interface cards and therefore multiple IP addresses, it is allowed to have multiple A RRs pertaining to the same domain name.

Example:
```
linux.cs.nctu.edu.tw            86400 IN A 140.113.168.127
                                86400 IN A 140.113.207.127
```

It means queries for linux.cs.nctu.edu.tw will be returned with these two IP addresses.

c.) **Canonical Name(CNAME):** A CNAME creates an alias with a domain name that points to the canonical domain name of an IP address, which is especially useful for running multiple services from a single IP address.

Example:
```
www.cs.nctu.edu.tw. 86400 IN CNAME
cache.cs.nctu.edu.tw.
cache.cs.nctu.edu.tw. 86400 IN A 140.113.166.122
```

clearly we can seed that cache is added as alias to the same domain name and this time we are accessing the cache service of that domain

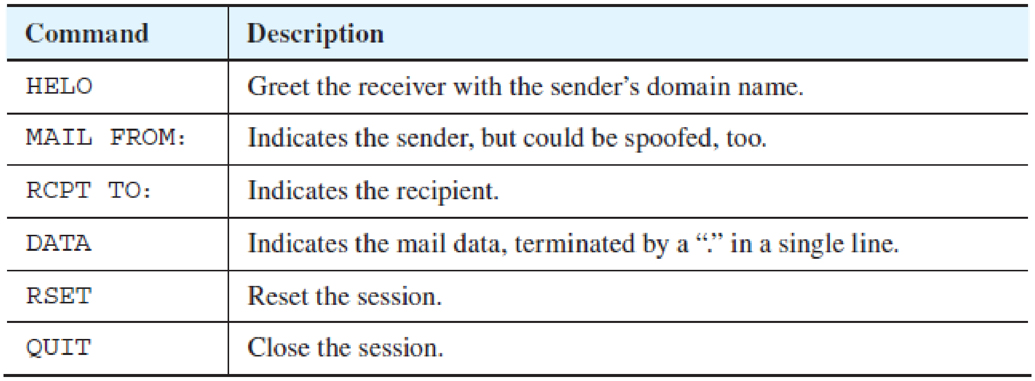
d.) **Pointer(PTR):** point to domain names from their corresponding IP addresses.it also called *reverse query*, querying with an IP address for the domain name.

Example:
```
10.23.113.140.in-addr.arpa.      86400      IN      PTR
laser0.cs.nctu.edu.tw.
```

INTERNET MAIL PROTOCOL:

1. Simple Mail Transfer Protocol (SMTP):

It is a standard host-to-host mail transport protocol traditionally operating over TCP on port 25. There is a server called SMTP server that accepts incoming messages and then delivers messages to appropriate recipients or to the next SMTP server. If a SMTP server is unable to deliver a message to a particular address and the errors are not due to permanent rejections, the message is put in a message queue for later delivery Retries of delivery continue until the delivery succeeds or the SMTP server gives up.



Command	Description
`HELO`	Greet the receiver with the sender's domain name.
`MAIL FROM:`	Indicates the sender, but could be spoofed, too.
`RCPT TO:`	Indicates the recipient.
`DATA`	Indicates the mail data, terminated by a "." in a single line.
`RSET`	Reset the session.
`QUIT`	Close the session.

2. Post Office Protocol (POP):

It is designed for user-to-mailbox access. A daemon that listens to port 110 and speaks POP3 is called a POP3 server which accept server accepts connections from clients and retrieves messages for them. It can only download and store the message.

States of POP :

Authorization: during authorization the identity of client is verified. The client must tell the POP3 server its username and

password.

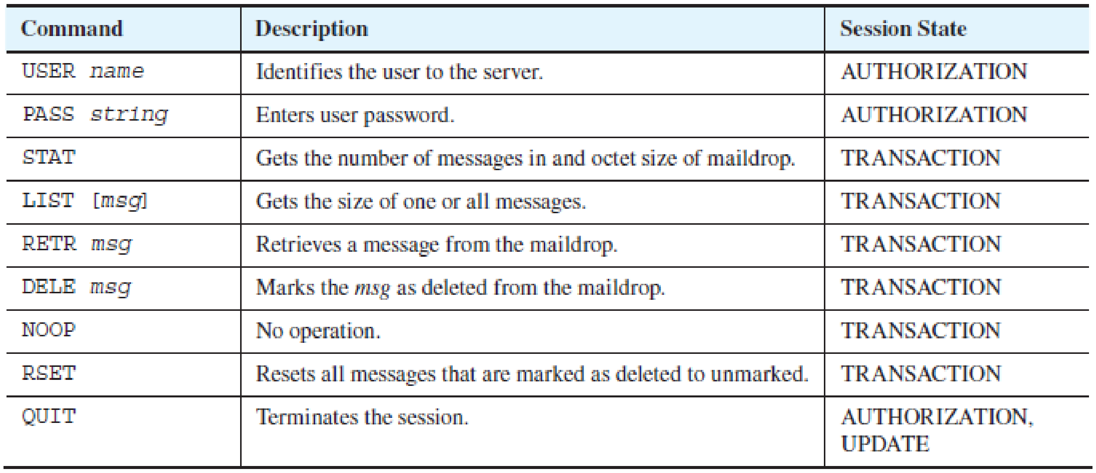
Transaction: After authorization the transaction state started, at this time the client can issue can issue commands to

the server and request the server to act on commands.

Update: When the client has issued the `QUIT` command, the session enters the `UPDATE` state In this state, the

POP3 server releases any resources allocated to the client during the `AUTHORIZATION` state, says

goodbye to the client, and finally closes the connection with the client.



Command	Description	Session State
`USER` *name*	Identifies the user to the server.	AUTHORIZATION
`PASS` *string*	Enters user password.	AUTHORIZATION
`STAT`	Gets the number of messages in and octet size of maildrop.	TRANSACTION
`LIST` [*msg*]	Gets the size of one or all messages.	TRANSACTION
`RETR` *msg*	Retrieves a message from the maildrop.	TRANSACTION
`DELE` *msg*	Marks the *msg* as deleted from the maildrop.	TRANSACTION
`NOOP`	No operation.	TRANSACTION
`RSET`	Resets all messages that are marked as deleted to unmarked.	TRANSACTION
`QUIT`	Terminates the session.	AUTHORIZATION, UPDATE

3. Internet Message Access Protocol (IMAP):

It is a replacement for the POP protocol. It comes from the need to use Web browsers anywhere to access e-mails on the server without actually downloading them. The IMAP not only download and store messages of user but also allow them to manipulate.

States of IMAP:

Non-authenticated: When a connection is established between the IMAP4 server and the client, the server enters the `non-authenticated` state. The client must supply authentication credentials before most commands can be permitted.

Authenticated: When a pre-authenticated connection starts, the server enters the `authenticated` state when acceptable authentication credentials have been provided or after an error in mailbox selection. In the `authenticated` state, the client must select a mailbox to access before commands that affect messages can be permitted.

Selected: When a mailbox has been successfully selected, the server enters the `selected` state. In this state, a mailbox has been selected to access.

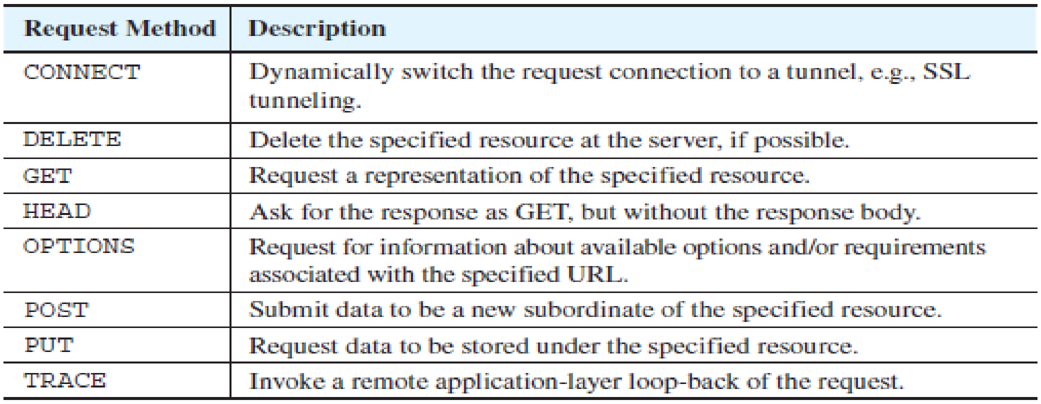
Logout : When the client asks to exit the server, the server enters the `logout` state. At this time, the server will close the connection

Hyper Text Transfer Protocol (HTTP):

The HTTP messages consist of requests and responses between clients and servers. The structure of request message are as follow:

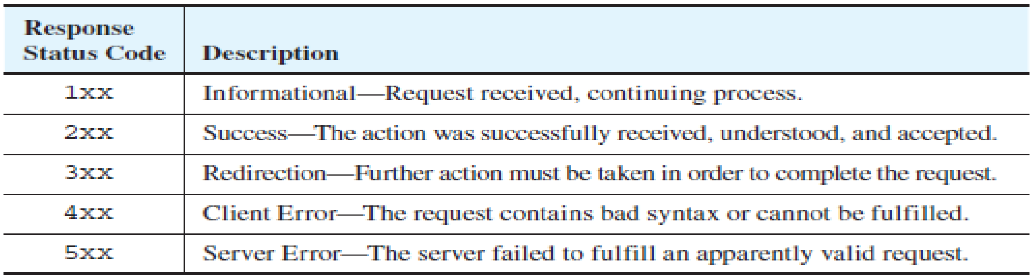
1. **Request line:** it includes the method to be applied to the resource, the *identifier* of the resource, and the protocol *version* in use
2. **Header:** which defines various *features* of the *data* that is requested or being provided
3. **Empty line:** it separate header from message body
4. **Optional Message Body:** it used when some info have to be sent

Request Method of *http* protocol:



Request Method	Description
CONNECT	Dynamically switch the request connection to a tunnel, e.g., SSL tunneling.
DELETE	Delete the specified resource at the server, if possible.
GET	Request a representation of the specified resource.
HEAD	Ask for the response as GET, but without the response body.
OPTIONS	Request for information about available options and/or requirements associated with the specified URL.
POST	Submit data to be a new subordinate of the specified resource.
PUT	Request data to be stored under the specified resource.
TRACE	Invoke a remote application-layer loop-back of the request.

Response code of *http* protocol:



Response Status Code	Description
1xx	Informational—Request received, continuing process.
2xx	Success—The action was successfully received, understood, and accepted.
3xx	Redirection—Further action must be taken in order to complete the request.
4xx	Client Error—The request contains bad syntax or cannot be fulfilled.
5xx	Server Error—The server failed to fulfill an apparently valid request.

How to convert a http from stateless to stateful:

HTTP is basically a *stateless* protocol, that is, the server does not have any state kept during transactions with clients A server fetches the page requested by the client and completes a transaction, so each transaction is independent of the other. We can make http server to act as stateful. For example:

To use the concept of *session*, in which all parameters pertaining to a potential session are kept in the server without client awareness. The server has limited memory space resulting in session states that soon expire. To fix this drawback, relatively small *cookies* are employed as an alternative, in which states are sent in HTTP headers to the clients
and then stored in the form of a cookie.

Persistent and Non-Persistent connection of *http*:

Persistent connection: when client and server interaction such that request and response sent over a separate TCP connection it means for each request-response pair a TCP connection is created

Non-persistent connection: in this all of the request-response pair is sent over a same TCP connection.

***http* uses both persistent and non-persistent connection and the persistent connection is set as default to http.**

Example to show persistent and non-persistent connection:

Let us consider that there is a page of base html file and 10 jpeg images and all the 11 object reside on same server and the ***url*** of that base html file is

⇨ ***http://www.santabanta.com/Rolls-Royce/home***

Step1: the http client process initiate a TCP connection to the server **www.santabanta.com** on port number 80.there will be a socket at the client and socket at the server associated to the TCP connection.

Step2: the http-client sends an http request message to server via its socket. the request message includes the path name /***Rolls-Royce/home***

Step3: The http server process receives the request message via its socket retrieve the requested object from its memory encapsulates the object in a http response message and send it to the client via a socket.

Step4: the http server process tells TCP to close the TCP connection

Step5: the http client receive the response message and TCP terminates the client extract file from response message, examines the html file and find reference to the **10** jpeg object.

Step6: the first four step repeated for each of the referenced jpeg object.

In above process the TCP connection transport exactly one request-response message thus in total **11** TCP connection is generated.

Now in persistent connection a connection is established and maintained for each requested object. A TCP buffer is created and kept on both the client and server side. These serve request from hundreds of different client simultaneously. In this process the server leaves the TCP connection open after sending a response, subsequent request and response between the same client and server can be sent over the same connection.

Introduction to socket programming:

Client Server Model:

- A network application consist of pair of program a client program and a server program residing in two different end-system.
- An application that initiates the communication is called a **client**. The client contacts a server, sends a request, and awaits a response
- An application that waits for incoming communication requests from clients is called a server. The server receives a client’s request, performs the necessary computation, and returns the result to the client

Socket:

- “Socket” is an interface between applications and the network services provided by OS. An application sends and receives data through a socket. This interface is an abstraction and known as socket abstraction.
- Developed by : Advanced Research Projects Agency (ARPA) in 1980.

Socket Descriptor:

- It is used for network I/O operation.
- Each active socket is identified by its socket descriptor.
- The Windows operating system keeps a separate table of socket descriptors (named socket descriptor table, or SDT) for each process.

Creating a socket:

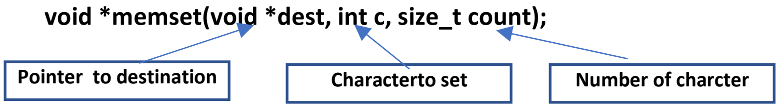
- The socket API contains a function socket() that can be called to create a socket. E.g.:
- When an application process calls socket(), the operating system allocates a new data structure to hold the information needed for communication, and fills in a new entry in the process’s socket descriptor table (**SDT**) with a pointer to the data structure.

```
#include <winsock2.h>
…
SOCKET s;
…
s = socket(AF_INET, SOCK_DGRAM, 0);
```

- The internal data structure for a socket contains many fields, but the system leaves most of them unfilled. The application must make additional procedure calls to fill in the socket data structure before the socket can be used.
- The socket is used for data communication between two processes (which may locate at different machines). So the socket data structure should at least contain the address information, e.g., **IP addresses, port numbers**, etc.

What is memset() and memcpy():

- memset(): set buffer to a specified character.



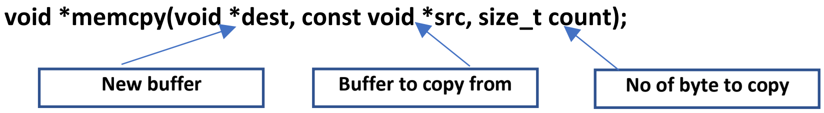
void *memset(void *dest, int c, size_t count);

Pointer to destination

Characterto set

Number of charcter

- memcpy(): copy characters between buffers



void *memcpy(void *dest, const void *src, size_t count);

New buffer

Buffer to copy from

No of byte to copy

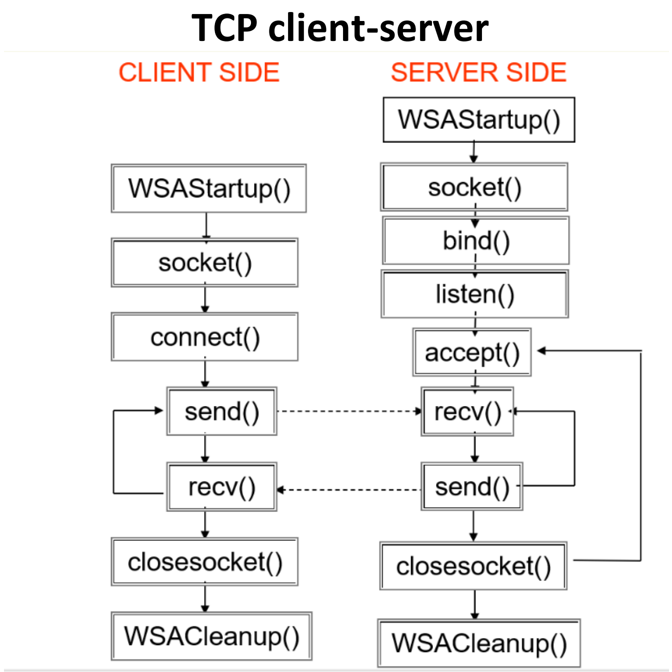
How a communication is identified:

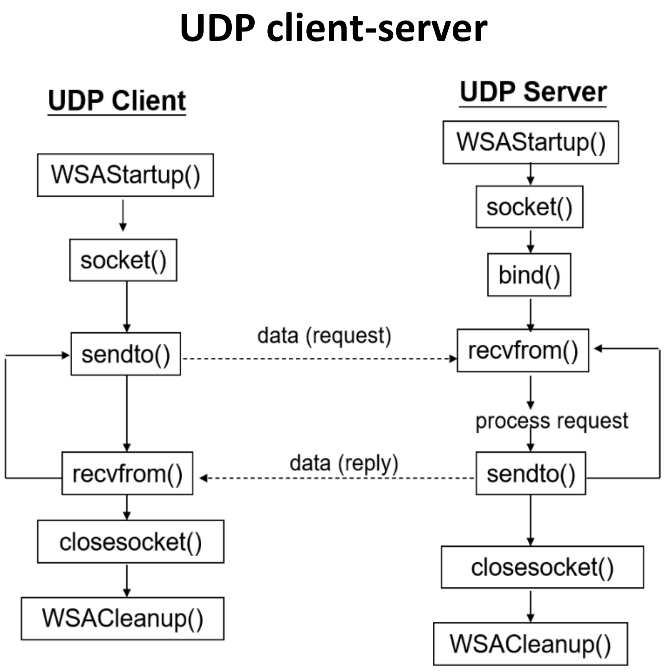
- A communication is identified bu means of source IP address ,source number, Destination IP Address ,destination port number.
- We have to specify the end point address that contain following steps :-
    - When a socket is created, it doesn't contain address information of either local machine or the remote machine. Before an application uses a socket for sending/receiving data, it must specify one or both of these addresses for the socket.
    - TCP/IP define a communication endpoint to consist of an IP address and a protocol port number.
    - Several data structures are involved to represent the endpoint address

      →general structure: ***struct sockaddr*** .

      →Specific Structure: ***struct sockaddr_in.***

TCP client-server vs UDP client-server:





Some import function which is common to both the TCP and UDP client-server:

1. **WSA Startup ():** it is called before using socket. It is needed because the operating system uses dynamically linked libraries (DLLs).
2. **WSA cleanip():** Once an application finishes using and closing sockets, it calls WSACleanup() to deallocate all data structures and socket bindings.
3. **Socket():** this function create a socket .

   Syntax: **SOCKET socket (int *af*, int *type*, int *protocol*);**
4. **Closesocket():** Once a client or server finishes using a socket, it calls closesocket() to deallocate it closesocket() immediately terminates the connection and deallocates the socket.
5.

PENDING

Notes by : Samundar singh

TRANSPORT LAYER

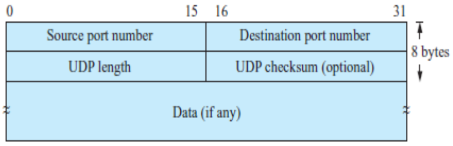
Notes by : Samundar singh

Unreliable connectionless Transport UDP(User Datagram Protocol):

It is is an *unreliable connectionless* transport protocol that does not provide reliability and rate control. It is a stateless protocol in that the sending or receiving of a segment is independent of that of any other segments.
It is generally used in streaming multimedia, network management, internet telephony etc. it has no congestion control.

UDP segment structure:

Header format:



The UDP header serves only two functions: addressing and error detection. It consists of four fields: source and destination port number, UDP length, and UDP checksum,

The communication between two application processes that reside different host in internet are done by binding the process with a local unique port number on local host it is ideal to bind frequently used server process to a fixed port number that are well known to the public.
The source/destination port numbers, concatenated with the source and destination IP addresses and protocol ID (indicating TCP or UDP) in the IP header, form a *socket pair* of *5-tuple* with a total length of **32 x 2 + 16 x 2 + 8 = 104** bits.
UDP allows applications on different hosts to send data segments directly to one another without having to establish a connection first. A UDP port accepts segments from a local application process, packs them into units called datagrams of no more than 64K bytes, and fills the 16-bit source and destination port numbers and other UDP header fields of the datagrams.

Error Control UDP Checksum:

UDP header also provides a 16-bit checksum field for checking on the integrity of each datagram The sender generates the checksum value and fills in the checksum field, which is to be verified by the receiver. To ensure that each received datagram is exactly the same as the one sent by the sender, the receiver recalculates the checksum with the received datagram and verifies if the result matches the value stored in the UDP checksum field. UDP receivers will *drop* the datagrams whose checksum field does not match the result they have calculated

UDP at sender side performs the 1c complement of the sum of all the 16-bit words in segment with any overflow encounter during the sum being wrapped around this result is put in the checksum field of UDP segment

TCP Connection :

Suppose a process running in one host wants to initiate a connection with another process in another host. the process that is initiating the connection is called the *client process*, while the other process is called the *server process* The client application process first informs the client transport layer that it wants to establish a connection to a process in the server. TCP in the client then proceeds to establish a TCP connection with TCP in the server Once a TCP connection is established, the two application processes can send data to each other

The client process passes a stream of data through the socket Once the data passes through the door, the data is in the hands of TCP running in the client. TCP directs this data to the connection's **send buffer**, From time to time, TCP will grab chunks of data from the send buffer and pass the data to the network layer.
The maximum amount of data that can be grabbed and placed in
a segment is limited by the **maximum segment size (MSS)**. The MSS is typically set by first determining the length of the largest link-layer frame that can be sent by the local sending host and then setting the MSS to ensure that a TCP segment(when encapsulated in an IP datagram) plus the TCP/IP header length (typically 40 bytes) will fit into a single link-layer frame.
TCP pairs each chunk of client data with a TCP header, thereby forming **TCP segments**. The segments are passed down to the network layer, where they are separately encapsulated within network-layer IP datagrams. The IP datagrams are then sent into the network. When TCP receives a segment at the other end, the segment's data is placed

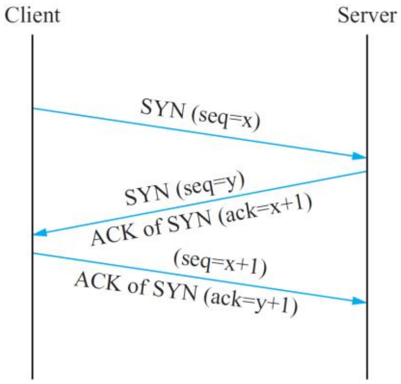
in the TCP connection’s receive buffer The application reads the stream of data from this buffer. Each side of the connection has its own send buffer and its own receive buffer

Connection Establishment(Three-Way handshake protocol):

Step 1 (SYN) : In the first step, client wants to establish a connection with server, so it sends a segment with SYN(Synchronize Sequence Number) which informs server that client is likely to start communication and with what sequence number it starts segments with

Step 2 (SYN + ACK): Server responds to the client request with SYN-ACK signal bits set. Acknowledgement(ACK) signifies the response of segment it received and SYN signifies with what sequence number it is likely to start the segments with

Step 3 (ACK) : In the final part client acknowledges the response of server and they both establish a reliable connection with which they will start the actual data transfer



Note: A TCP connection is full Duplex

Connection Termination:

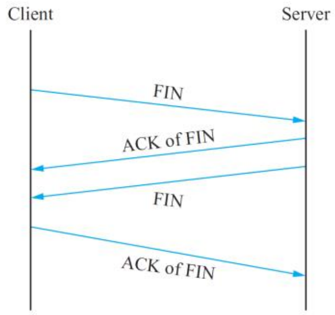
Step 1 (FIN From Client) – Suppose that the client application decides it wants to close the connection. (Note that the server could also choose to close the connection). This causes the client send a TCP segment with the **FIN** bit set to **1** to server and to enter the **FIN_WAIT_1** state. While in the **FIN_WAIT_1** state, the client waits for a TCP segment from the server with an acknowledgment (ACK).

Step 2 (ACK From Server) – When Server received FIN bit segment from Sender (Client), Server Immediately send acknowledgement (ACK) segment to the Sender (Client).

Step 3 (Client waiting) – While in the **FIN_WAIT_1** state, the client waits for a TCP segment from the server with an acknowledgment. When it receives this segment, the client enters the **FIN_WAIT_2** state. While in the **FIN_WAIT_2** state, the client waits for another segment from the server with the FIN bit set to 1.

Step 4 (FIN from Server) – Server sends FIN bit segment to the Sender(Client) after some time when Server send the ACK segment (because of some closing process in the Server).

Step 5 (ACK from Client) – When Client receive FIN bit segment from the Server, the client acknowledges the server’s segment and enters the **TIME_WAIT** state. The **TIME_WAIT** state lets the client resend the final acknowledgment in case the **ACK** is lost.The time spent by client in the **TIME_WAIT** state is depend on their implementation, but their typical values are 30 seconds, 1 minute, and 2 minutes. After the wait, the connection formally closes and all resources on the client side (including port numbers and buffer data) are released.



TCP State:

Client side

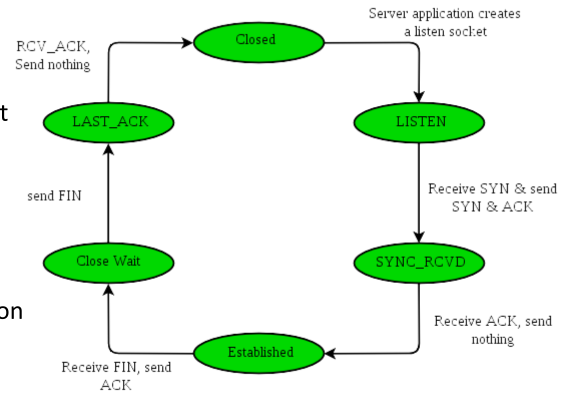


Fig : TCP states visited by a client TCP

LISTEN—waiting for a connection request from any remote TCP client.
SYN-RECEIVED—waiting for an acknowledgment of a connection request after having both received and sent a connection request.
ESTABLISHED—an open connection; data can be sent in both directions. The normal state for the data transfer phase of the connection
CLOSE-WAIT—waiting for a connection termination request from the local user.
LAST-ACK—waiting for an acknowledgment of the connection termination request previously sent to the remote TCP.
CLOSING—waiting for an acknowledgment of a connection termination Request from the remote TCP

Server side

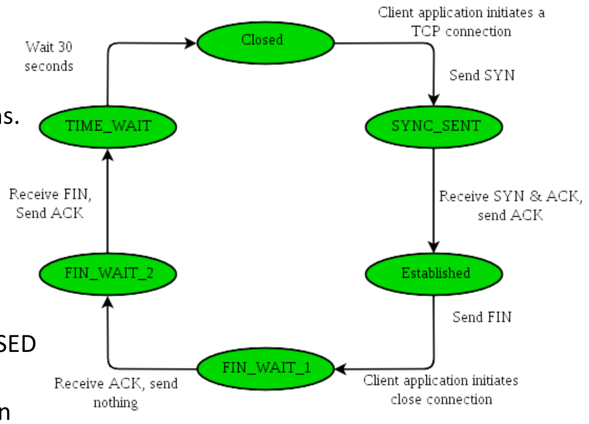

Closed
Client application initiates a TCP connection
Wait 30 seconds
Send SYN
TIME_WAIT
SYNC_SENT
Receive FIN, Send ACK
Receive SYN & ACK, send ACK
FIN_WAIT_2
Established
Send FIN
FIN_WAIT_1
Receive ACK, send nothing
Client application initiates close connection

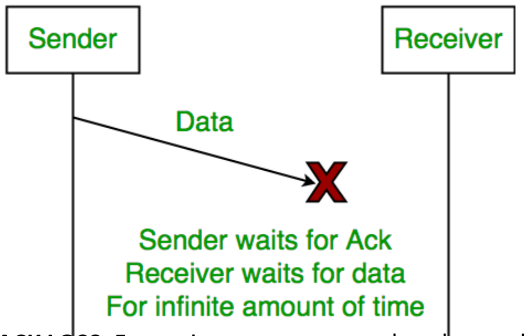
Fig : TCP states visited by a client TCP

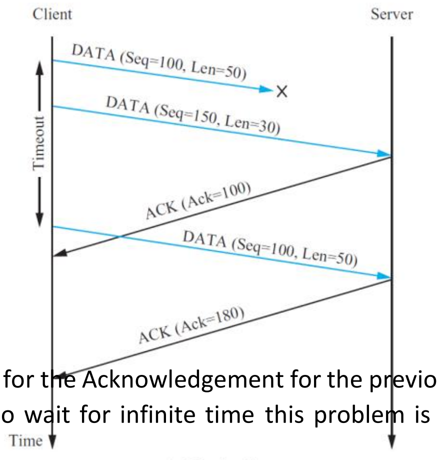
SYN-SENT—waiting for a matching connection request after having sent a connection request
ESTABLISHED—an open connection; data can be sent in both directions. The normal state for the data transfer phase of the connection.
FIN-WAIT-1—waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent.
FIN-WAIT-2—waiting for a connection termination request from the Remote TCP.
TIME_WAIT—waiting for enough time before transitioning to the CLOSED state to ensure the remote TCP receives its last ACK.
CLOSING—waiting for an acknowledgment of a connection termination request from the remote TCP.

Sliding Window :

Before starting this we have to know the basic abnormal case and transmission fact ;

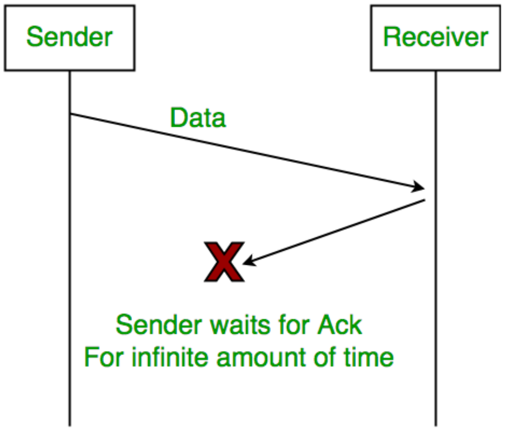
1.) **PACKET LOSS:** As we can see in the picture that the server does not receive any data due to data loss so there is no acknowledgement is generated and the sender have to wait for infinite time. This problem is solved by using a timeout period where at some time interval if the sender does not receive any ACK from the previous data then it again sent it back

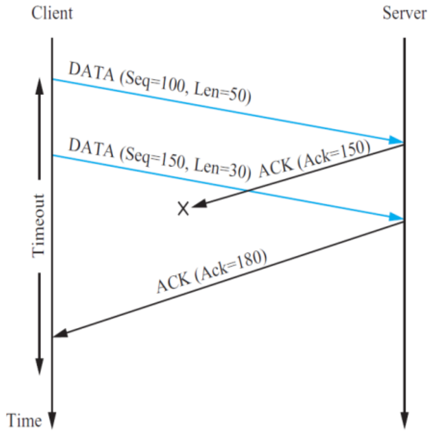




(a) Packet loss

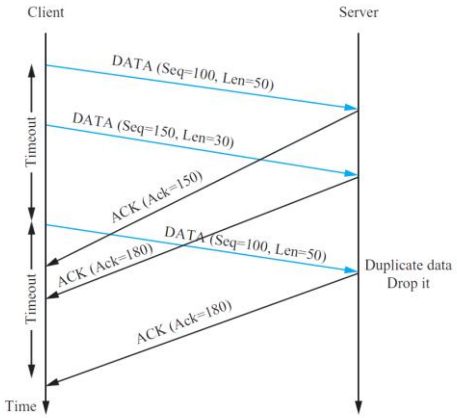
2.) **ACK LOSS:** From picture we can see that the sender wait for the Acknowledgement for the previous data to send further data but the ACK is lost so sender have to wait for infinite time this problem is solved by assigning Sequence Number when a Ack is not received within the time limit then a ACK with -ve number is sent.





(c) ACK loss

3.) **Delayed ACK or Data:** here due to delay of sending ACK or data the timeout will trigger the resending procedure due to which data is repeatedly sent again and again which result in loss of data as well as time



(b) Delay

This procedure is called **Stop and Wait ARQ** the problem is that it can sent one packet at a time sender always waits for acknowledgement even if it has next packet ready to send. so propagation delay is too high that why we are using the concept of sliding window.

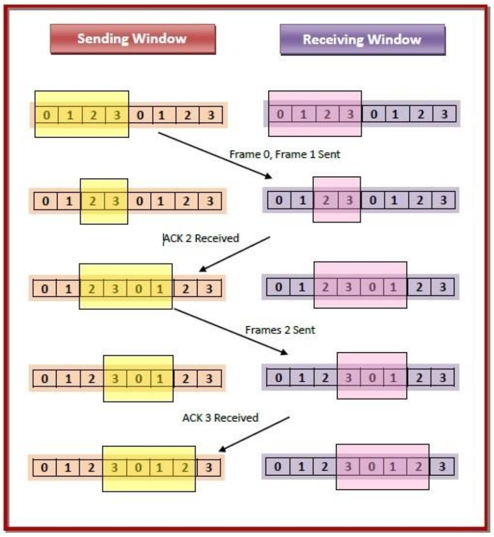
Sliding Window Protocol:

Sliding Window protocol handles this efficiency issue by sending more than one packet at a time with a larger sequence numbers. The sender and receiver are programmed to use a fixed window size, which is the maximum amount of data that can be sent before an acknowledge arrives.

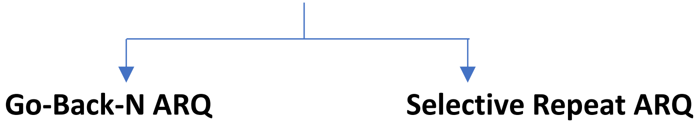
In these protocols, the sender has a buffer called the **sending window** and the receiver has buffer called the **receiving window.** The size of the sending window determines the sequence number of the outbound frames.it means The range of sequence numbers that can be assigned is 0 to 2^n-1 thus the size of the sending window is 2^n-1. And n order to accommodate a sending window size of 2^n-1, a n-bit sequence number is chosen.

The sequence numbers are numbered as modulo-n. For example, if the sending window size is 4, then the sequence numbers will be 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, and so on. The number of bits in the sequence number is 2 to generate the binary sequence 00, 01, 10, 11.

Suppose that we have sender window and receiver window each of size 4. So the sequence numbering of both the windows will be 0,1,2,3,0,1,2 and so on. The following diagram shows the positions of the windows after sending the frames and receiving acknowledgments.



Types of Sliding window protocol:



Go-Back-N ARQ

Selective Repeat ARQ

1. **Go-Back-N ARQ:** the sender sends the frames continuously without waiting for acknowledgement. That is why it is also called as continuous ARQ. the sender sends the frames continuously without waiting for acknowledgement. That is why it is also called as continuous ARQ. . As the receiver receives the frames, it keeps on sending ACKs or a NACK, in case a frame is incorrectly received. When the sender receives a NACK, it retransmits the frame in error plus all the succeeding frames

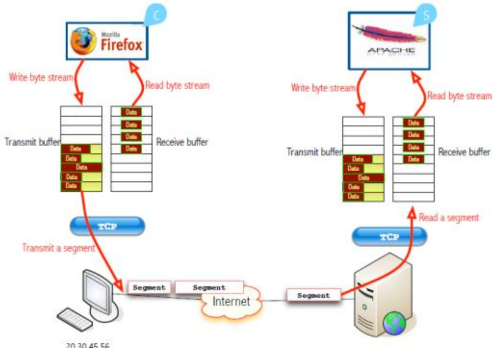
   If a frame is lost, the receiver sends NAK after receiving the next frameIn case there is long delay before sending the NAK, the sender will resend the lost frame after its timer times out. If the ACK frame sent by the receiver is lost, the sender resends the frames after its timer times out

   Here draw all diagram of abnormal cases as mention in previous page.

2. **Selective-Repeat ARQ** The selective-repetitive ARQ scheme retransmits only those for which NAKs are received or for which timer has expired:

Piggybacking: In practice, the link between receiver and transmitter is full duplex and usually both transmitter and receiver stations send data to each over. So, instead of sending separate acknowledgement packets, a portion (few bits) of the data frames can be used for acknowledgement. This phenomenon is known as piggybacking. The piggybacking helps in better channel utilization. Further, multi-frame acknowledgement can be done.

Triggering Transmission:



- TCP supports a byte stream abstraction
- Application programs write bytes into streams
- It is up to TCP to decide that it has enough bytes to send a segment

It has 3 mechanism to trigger the transmission of a segment

1.) from the sending process (Size of the largest segment TCP can send without causing local IP to fragment)
2.) Sending process has explicitly asked TCP to send it TCP supports **push operation**
3.) When the self-clocking timer fires (Reception of a brand **new ACK**) Resulting segment contains as many bytes as are currently buffered for transmission

Silly Window Syndrome(SWS):

It arises due to poor implementation of TCP. where sender window size to shrink to a silly value due to which window size shrinks to such an extent where the data being transmitted is smaller than TCP Header.

The major cause of this problem is:

1. **Sender window transmitting one byte of data repeatedly:**

   Suppose if only one byte of data is generated by an application due to some reason then poor implementation of TCP leads to transmit this small segment of data. Every time the application generates a byte of data, the window transmits it.

   For example if we have to send 5 byte of data then in such situation the sender will attach 1 byte of data to the TCP header (20byte) with ip header(20 byte) it means if we have to send 1 byte of data then total (**20+20**)byte +1 byte =41 byte of bandwidth is utilized so in order to send all 5 byte data we have consumed **205 bytes** o0f bandwidth which is not ideal it have to consumed (5+40) **45 byte** of bandwidth so there is unnecessary consumption of the bandwidth.

   Nagle's Solution:

   This algorithm says that we can create a buffer with a timeout the data can only transmit either the buffer is full or the timeout for buffering runs out and it can send the data to receiver thus there is uniformity in data transmission is maintained

2. **Receiver window accepting one byte of data repeatedly**

   Suppose that the receiver side window is full and only 1 byte of data space is available thus it informs the sender to send only one byte and this process goes on since every time it receives one-byte data the one byte of data processed and a ACK for 1 byte is sent

   Clark's Solution:

   Clark suggests that the receiver should not advertise the window size til the window is half empty

Adaptive Re-Transmission:

TCP uses several timers to ensure that excessive delays are not encountered during communications.

1. **Re-Transmission Timer:** To retransmit lost segments, TCP uses retransmission timeout (RTO). When TCP sends a segment the timer starts and stops when the acknowledgment is received. If the timer expires timeout occurs and the segment is retransmitted

2. **Persist-Timer:** To deal with a zero-window-size deadlock situation, TCP uses a persistence timer.When the sending TCP receives an acknowledgment with a window size of zero, it starts a persistence timer. When the persistence timer goes off, the sending TCP sends a special segment called a probe. This segment contains only 1 byte of new data. It has a sequence number, but its sequence number is never acknowledged; it is even ignored in calculating the sequence number for the rest of the data. The probe causes the receiving TCP to resend the acknowledgment which was lost.

3. **Keep Alive Timer –** A keepalive timer is used to prevent a long idle connection between two TCPs. If a client opens a TCP connection to a server transfers some data and becomes silent the client will crash. In this case, the connection remains open forever. So a keepalive timer is used. Each time the server hears from a client, it resets this timer. The time-out is usually 2 hours. If the server does not hear from the client after 2 hours, it sends a probe segment. If there is no response after 10 probes, each of which is 75 s apart, it assumes that the client is down and terminates the connection.

Congestion (extra knowledge):

It is a phenomenon in which when the message traffic is so heavy that it slows down network response time. For better understanding let us take an example of **Leaky Bucket Algorithm**

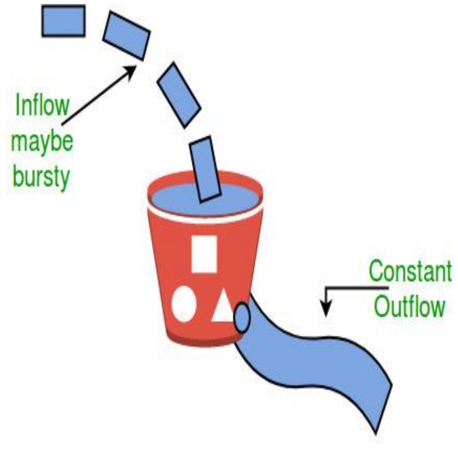
Leaky Bucket Algorithm:

Imagine a bucket with a small hole in the bottom. No matter at what rate water enters the bucket, the outflow is at constant rate. When the bucket is full with water additional water entering spills over the sides and is lost.

Similarly with the Networks:

1. When host wants to send packet, packet is thrown into the bucket.
2. The bucket leaks at a constant rate, meaning the network interface transmits packets at a constant rate.
3. Bursty traffic is converted to a uniform traffic by the leaky bucket.
4. In practice the bucket is a finite queue that outputs at a finite rate.

Solution of Leaky Bucket Algorithm:



Notes by : Samundar singh

Token bucket Algorithm:

The leaky bucket algorithm enforces output pattern at the average rate, no matter how bursty the traffic is. So in order to deal with the bursty traffic we need a flexible algorithm so that the data is not lost. One such algorithm is token bucket algorithm.

Steps of this algorithm can be described as follows:

1. In regular intervals tokens are thrown into the bucket. ƒ
2. The bucket has a maximum capacity. ƒ
3. If there is a ready packet, a token is removed from the bucket, and the packet is sent.
4. If there is no token in the bucket, the packet cannot be sent.

TCP Congestion Control:

TCP uses a congestion window and a congestion policy that avoid congestion. suppose a condition where network cannot deliver the data as fast as it is created by the sender, it must tell the sender to slow down

Congestion policy in TCP –

1. Slow Start

In this phase after every RTT the congestion window size increments exponentially.

```
Initially cwnd = 1

After 1 RTT, cwnd = 2^(1) = 2

2 RTT, cwnd = 2^(2) = 4

3 RTT, cwnd = 2^(3) = 8
```

In the first successful transmission and acknowledgement of a TCP segment increases the window to two segments. After successful transmission of these two segments and acknowledgements completes, the window is increased to four segments. Then eight segments, then sixteen segments and so on, At some point the congestion window may become too large for the network or network conditions may change such that packets may be dropped. Packets lost will trigger a timeout at the sender and then congestion avoidance came into picture.

2. Congestion Avoidance

In the Congestion Avoidance algorithm a retransmission timer expiring or the reception of duplicate ACKs can implicitly signal the sender that a network congestion situation is occurring. The sender immediately sets its transmission window to one half of the current window size.

If congestion was indicated by a timeout, the congestion window is reset to one segment, which automatically puts the sender into Slow Start mode. If congestion was indicated by duplicate ACKs, the Fast Retransmit and Fast Recovery algorithms are invoked

3. **Fast Retransmit**

When a duplicate ACK is received, the sender does not know if it is because a TCP segment was lost or simply that a segment was delayed and thus it received **out of order** at the receiver

No more than one or two duplicate ACKs should be received when simple out of order conditions exist. If more than two duplicate ACKs are received by the sender, it is a strong indication that at least one segment has been lost

When three or more duplicate ACKs are received, the sender does not even wait for a retransmission timer to expire before retransmitting the segment This process is called the Fast Retransmit algorithm

4. **Fast Recovery**

Since the Fast Retransmit algorithm is used when duplicate ACKs are being received, the TCP sender has implicit knowledge that there is data still flowing to the receive The reason is because duplicate ACKs can only be generated when a segment is received. This is a strong indication that serious network congestion may not exist and that the lost segment was a rare event. So instead of reducing the flow of data abruptly by going all the way into Slow Start, the sender only enters Congestion Avoidance mode. Rather than start at a window of one segment as in Slow Start mode, the sender resumes transmission with a larger window, incrementing as if in Congestion

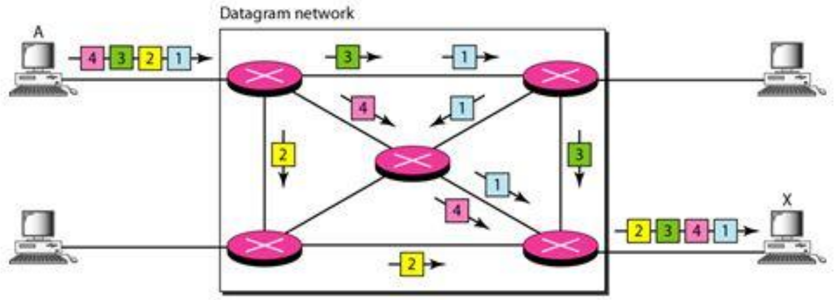
NETWORK LAYER

Notes by : Samundar singh

Packet Switching Technique:

DATAGRAM Network:

Datagram network provides network-layer **connectionless service.** In a datagram network, each packet is treated independently of all others. Even if a packet is part of a multi packet transmission, the network treats it as though it existed alone. Packets in this approach are referred to as datagrams.



n this example, all four packets (or datagrams) belong to the same message, but may travel different paths to reach their destination. This is so because the links may be involved in carrying packets from other sources and do not have the necessary bandwidth available to carry all the packets from A to X. This approach can cause the datagrams of a transmission to arrive at their destination out of order with different delays between the packets

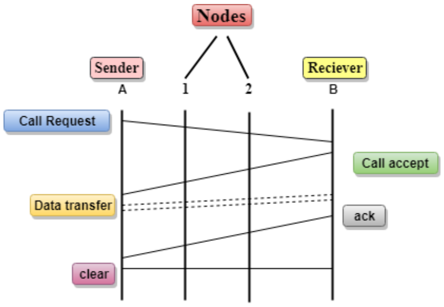
- **its header contains full information about the destination of the packet like source address, destination address and sequence number.**
- Packets will travel across the network, taking the shortest path as possible.
- All the packets are reassembled at the receiving end in correct order.
- If any packet is missing or corrupted, then the message will be sent to resend the message.
- If the correct order of the packets is reached, then the acknowledgment message will be sent.
- Intermediate nodes take the routing decisions to forward the packets.

Virtual Circuit Switching:

Virtual circuit switching is a packet switching methodology whereby a path is established between the source and the final destination through which all the packets will be routed during a call.

- Virtual Circuit Switching is also known as connection-oriented switching.
- In the case of Virtual circuit switching, a preplanned route is established before the messages are sent.
- Call request and call accept packets are used to establish the connection between sender and receiver.

⇨ In this case, the path is fixed for the duration of a logical connection.



In the above diagram, A and B are the sender and receiver respectively. 1 and 2 are the nodes. Call request and call accept packets are used to establish a connection between the sender and receiver. When a route is established, data will be transferred. After transmission of data, an acknowledgment signal is sent by the receiver that the message has been received. If the user wants to terminate the connection, a clear signal is sent for the termination.

Routing:

A Routing is a process of selecting path along which the data can be transferred from source to the destination. Routing is performed by a special device known as a router. It is performed in the **Network Layer.**

The routing algorithms are used for routing the packets. The routing algorithm is nothing but a software responsible for deciding the optimal path through which packet can be transmitted. The routing algorithm initializes and maintains the routing table for the process of path determination.

Routing can be classified into three categories:

1. Static Routing
2. Default Routing
3. Dynamic Routing

1. Static Routing:

It is a technique in which the administrator manually adds the routes in a routing table. A Router can send the packets for the destination along the route defined by the administrator. In this technique, routing decisions are not made based on the condition or topology of the networks

Advantages –

- No routing overhead for router CPU which means a cheaper router can be used to do routing.
- It adds security because only administrator can allow routing to particular networks only.
- No bandwidth usage between routers.

Disadvantage –

- For a large network, it is a hectic task for administrator to manually add each route for the network in the routing table on each router.
- The administrator should have good knowledge of the topology. If a new administrator comes, then he has to manually add each route so he should have very good knowledge of the routes of the topology.

2. Default Routing:

This is the method where the router is configured to send all packets towards a single router (next hop). It doesn't matter to which network the packet belongs, it is forwarded out to router which is configured for default routing. It is generally used with stub routers. A stub router is a router which has only one route to reach all other networks.

3. Dynamic Routing:

Dynamic routing makes automatic adjustment of the routes according to the current state of the route in the routing table. Dynamic protocols are used to discover the new routes to reach the destination. ìf any route goes down, then the automatic adjustment will be made to reach the destination.

A dynamic protocol have following features:

1. The routers should have the same dynamic protocol running in order to exchange routes.
2. When a router finds a change in the topology then router advertises it to all other routers.

Advantages –

- Easy to configure.
- More effective at selecting the best route to a destination remote network and also for discovering remote network.

Disadvantage –

- Consumes more bandwidth for communicating with other neighbors.
- Less secure than static routing.

IPv4:

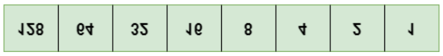
An IP stands for internet protocol. An IP address is assigned to each device connected to a network. Each device uses an IP address for communication. It also behaves as an identifier as this address is used to identify the device on a network.

IPv4 is a version 4 of IP. It is a current version and the most commonly used IP address. It is a 32-bit address written in four numbers separated by 'dot'

Example: **66.94.29.13**

each group of numbers separated by dots is called an Octet. Each number in an octet is in the range from 0-255

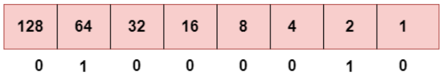
Representation of 8 Bit Octet



128	64	32	16	8	4	2	1

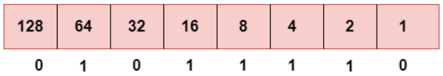
Now, we will see how to obtain the binary representation of the IP address, i.e., 66.94.29.13

Step 1: First, we find the binary number of 66.



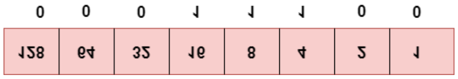
128	64	32	16	8	4	2	1
0	1	0	0	0	0	1	0

Step 2: Now, we calculate the binary number of 94.



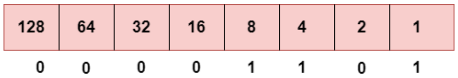
128	64	32	16	8	4	2	1
0	1	0	1	1	1	1	0

Step 3: The next number is 29.



0	0	0	1	1	1	0	0
128	64	32	16	8	4	2	1

Step 4: The last number is 13.



128	64	32	16	8	4	2	1
0	0	0	0	1	1	0	1

Thus we have the ip address in form of binary is:

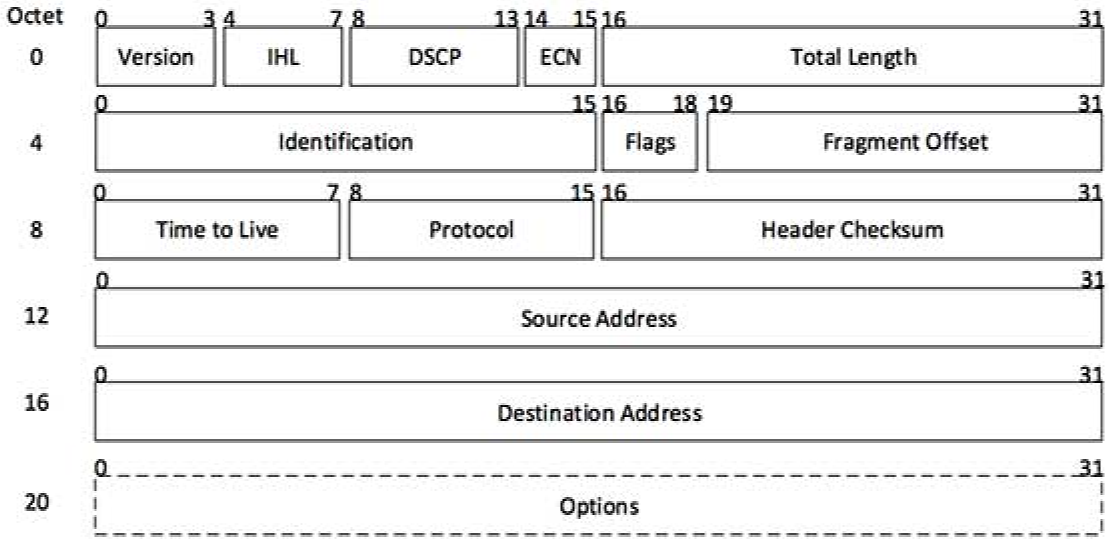
01000010. 01011110. 00011101. 00001101.

Drawbacks of IPv4

IPv4 produces 4 billion addresses, which are not enough for each device connected to the internet

Internet Routing is inefficient in IPv4.

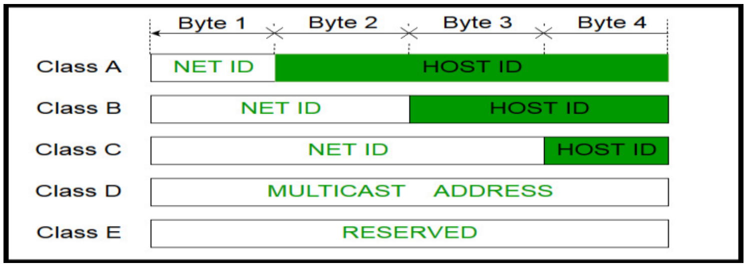
Header of IPv4



IP Address Classes:

The 32 bit IP address is divided into five sub-classes. These are:

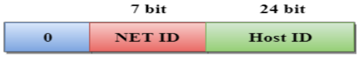
1. Class A
2. Class B
3. Class C
4. Class D
5. Class E



1. Class A

IP address belonging to class A are assigned to the networks that contain a large number of hosts.

- The network ID is 8 bits long.
- The host ID is 24 bits long.



In Class A, the first bit in higher order bits of the first octet is always set to 0 and the remaining 7 bits determine the network ID. The 24 bits determine the host ID in any network

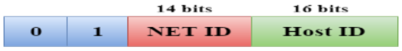
The total number of networks in Class A = 2^7 = 128 network address

The total number of hosts in Class A = 2^{24} - 2 = 16,777,214 host address

2. Class B

IP address belonging to class B are assigned to the networks that ranges from medium-sized to large-sized networks.

- The network ID is 16 bits long.
- The host ID is 16 bits long.



In Class B, the higher order bits of the first octet is always set to 10, and the remaining14 bits determine the network ID. The other 16 bits determine the Host ID.

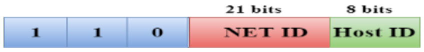
The total number of networks in Class B = 2^{14} = 16384 network address

The total number of hosts in Class B = 2^{16} - 2 = 65534 host address

3. Class C

IP address belonging to class C are assigned to small-sized networks.

- The network ID is 24 bits long.
- The host ID is 8 bits long.



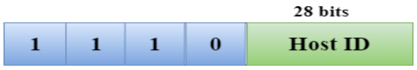
In Class C, the higher order bits of the first octet is always set to 110, and the remaining 21 bits determine the network ID. The 8 bits of the host ID determine the host in a network.

The total number of networks = 2^{21} = 2097152 network address

The total number of hosts = 2^8 - 2 = 254 host address

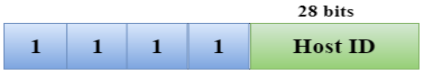
4. Class D

IP address belonging to class D are reserved for multi-casting. The higher order bits of the first octet of IP addresses belonging to class D are always set to 1110. The remaining bits are for the address that interested hosts recognize.



5. Class E

IP addresses belonging to class E are reserved for experimental and research purposes. This class doesn't have any sub-net mask. The higher order bits of first octet of class E are always set to 1111.



Classless Inter Domain Routing (CIDR):

It is also known as classless addressing In the Classful addressing the no of Hosts within a network always remains the same depending upon the class of the Network..

Class A network contains 2^{24} Hosts,
Class B network contains 2^{16} Hosts,
Class C network contains 2^{8} Hosts

Now, let's suppose an Organization requires 2^{14} hosts, then it must have to purchase a Class B network. In this case, (2^{16} -2^{14} = 49152 approx) Hosts will be wasted. This is the major drawback of Classful Addressing.
In order to reduce the wastage of IP addresses a new concept of **Classless Inter-Domain Routing** is introduced.

CIDR: It is as also a 32-bit address, which includes a special number which represents the number of bits that are present in the Block Id.

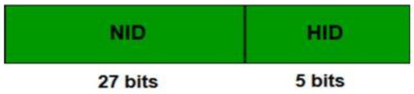
```
a . b . c . d / n
```

Where, n is number of bits that are present in Block Id / Network Id.

RULES for forming CIDR Block:

1. All IP addresses must be contiguous.
2. Block size must be the power of 2 (2^n).
   If the size of the block is the power of 2, then it will be easy to divide the Network. Finding out the Block Id is very easy if the block size is of the power of 2.

   Example: If the Block size is 2^5 then, Host Id will contain 5 bits and Network will contain 32 – 5 = 27 bits.



3. First IP address of the Block must be evenly divisible by the size of the block. in simple words, the least significant part should always start with zeroes in Host Id. Since all the least significant bits of Host Id is zero,
   then we can use it as Block Id part.

Q. Check whether 100.1.2.32 to 100.1.2.47 is a valid IP address block or not?

1. All the IP addresses are contiguous.
2. Total number of IP addresses in the Block = 16 = 2^4.

3. 1st IP address: 100.1.2.00100000
   Since, Host Id will contains last 4 bits and all the least significant 4 bits are zero. Hence, first IP address is evenly divisible by the size of the block.

All the three rules are followed by this Block. Hence, it is a valid IP address block

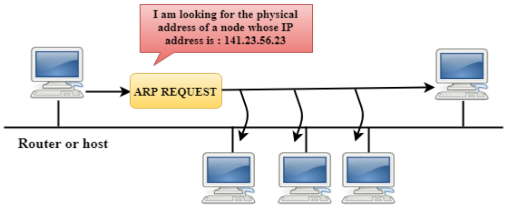
Address Resolution Protocol (ARP):

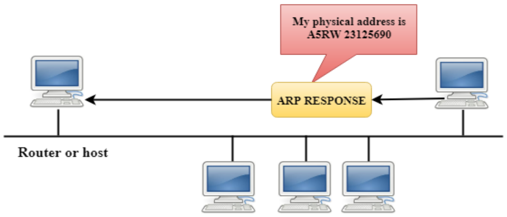
It is used to associate an IP address with the MAC address. Most of the computer programs/applications use **logical address (IP address)** to send/receive messages, however the actual communication happens over the **physical address (MAC address)** so ARP functionality is to translate IP address to physical address.

> **Note: MAC address: The MAC address is used to identify the actual device.**
> **IP address: It is an address used to locate a device on the network.**

Working of ARP:

If the host wants to know the physical address of another host on its network, then it sends an ARP query packet that includes the IP address and broadcast it over the network. Every host on the network receives and processes the ARP packet, but only the intended recipient recognizes the IP address and sends back the **physical** address.


I am looking for the physical address of a node whose IP address is : 141.23.56.23
ARP REQUEST
Router or host



Steps in ARP protocol:

Step1: The device will first look at its internet list, called the ARP cache to check whether an IP address contains a matching MAC address or not.

Step 2: If ARP cache is empty, then device broadcast the message to the entire network asking each device for a matching MAC address.

Step 3: The device that has the matching IP address will then respond back to the sender with its MAC address

Step 4: Once the MAC address is received by the device, then the communication can take place between two devices.

Step 5 If the device receives the MAC address, then the MAC address gets stored in the ARP cache**:**

Dynamic Host Configuration Protocol (DHCP):

It is a network protocol that allows a server to automatically assign an IP address from a specified range of numbers (a scope) to a computer or device when it is connected to a given network. So, it is a protocol for assigning dynamic IP addresses to devices on a network.

- ⇨ DHCP manages all the nodes or devices added or dropped from the network
- ⇨ DHCP maintains the unique IP address of the host using a DHCP server.

- It sends a request to the DHCP server whenever a client/node/device, which is configured to work with DHCP, connects to a network. The server acknowledges by providing an IP address to the client/node/device.

Working of DHCP:

DHCP runs at the application layer of the TCP/IP protocol. DHCP is based on client-server protocol in which servers manage a pool of unique IP addresses, as well as information about client configuration parameters, and assign addresses out of those address pools.on basis of which it assign dynamic IP address to the Client.

Steps in DHCP Protocol:

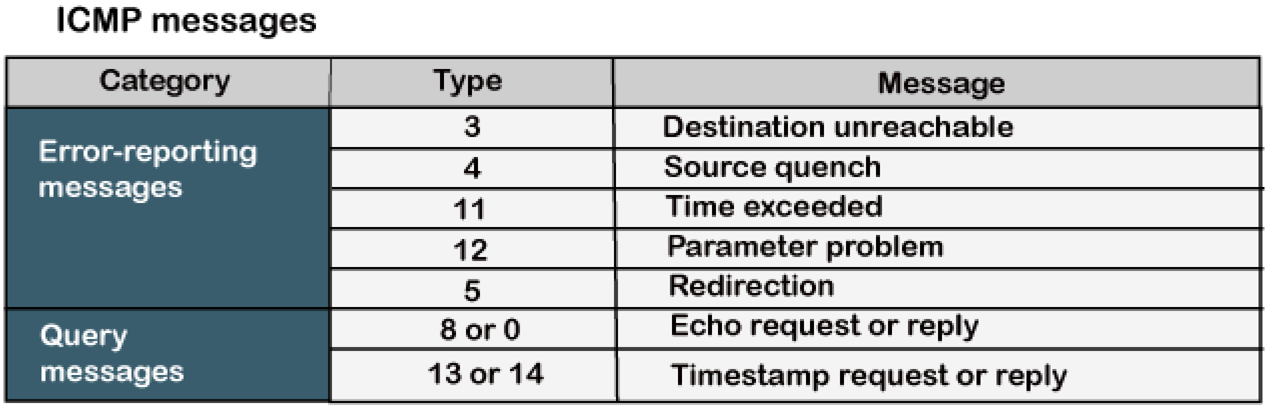
- First of all, a client (network device) must be connected to the internet.
- DHCP clients request an IP address. Typically, client broadcasts a query for this information.
- DHCP server responds to the client request by providing IP server address and other configuration information. This configuration information also includes time period, called a lease, for which the allocation is valid.
- When refreshing an assignment, a DHCP clients request the same parameters, but the DHCP server may assign a new IP address. This is based on the policies set by the administrator.

Internet Control Message Protocol (ICMP):

It is a network layer protocol. It is used for error handling in the network layer, and it is primarily used on network devices such as routers

For example, some sender wants to send the message to some destination, but the router couldn't send the message to the destination. In this case, the router sends the message to the sender that I could not send the message to that destination.

The ICMP messages are usually divided into two categories:

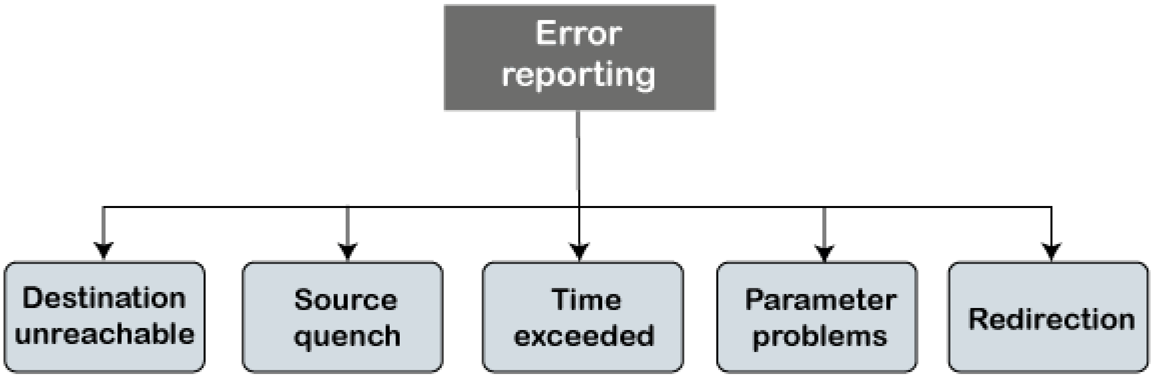


ICMP messages

Category	Type	Message
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply

> ***Note: The ICMP protocol always reports the error messages to the original source. For example, when the sender sends the message, if any error occurs in the message then the router reports to the sender rather than the receiver as the sender is sending the message.***

Types of Error Reporting Messages:



a. **Destination unreachable**
The destination unreachable error occurs when the packet does not reach the destination. Suppose the sender sends the message, but the message does not reach the destination, then the intermediate router reports to the sender that the destination is unreachable.

b. **Source quench**
when receiving host detects that rate of sending packets (traffic rate) to it is too fast it sends the source quench message to the source to slow down so that no packet can be lost. A souce quench message informs the sender that the datagram has been discarded due to congestion occurs in the network layer So, the sender must either stop or slow down the sending of datagrams until the congestion is reduced. The router sends one source-quench message for each datagram that is discarded due to the congestion in the network layer.

c. **Time exceeded**
Sometimes the situation arises when there are many routers that exist between the sender and the receiver. When the sender sends the packet, then it moves in a routing loop. The time exceeded is based on the time-to-live value. When the packet traverses through the router, then each router decreases the value of TTL by one. Whenever a router decreases a datagram with a **time-to-live value to zero**, then the router discards a datagram and sends the time exceeded message to the original source

d. **Parameter problems**
Whenever packets come to the router then calculated header checksum should be equal to recieved header checksum then only packet is accepted by the router. If there is mismatch packet will be dropped by the router.

ICMP will take the source IP from the discarded packet and informs to source by sending parameter problem message.

e. **Redirection Problems**

Redirect requests data packets be sent on an alternate route. The message informs to a host to update its routing information (to send packets on an alternate route).

Routing

Distance Vector Routing(DVR) :

distance Vector Routing protocol is a ‘dynamic routing’ protocol. With this protocol, every router in the network creates a routing table which helps them in determining the **shortest path** through the network. All the routers in the network are aware of every other router in the network and they keep on updating their routing table **periodically**. This protocol uses the principle of **Bellman-Ford’s** algorithm.

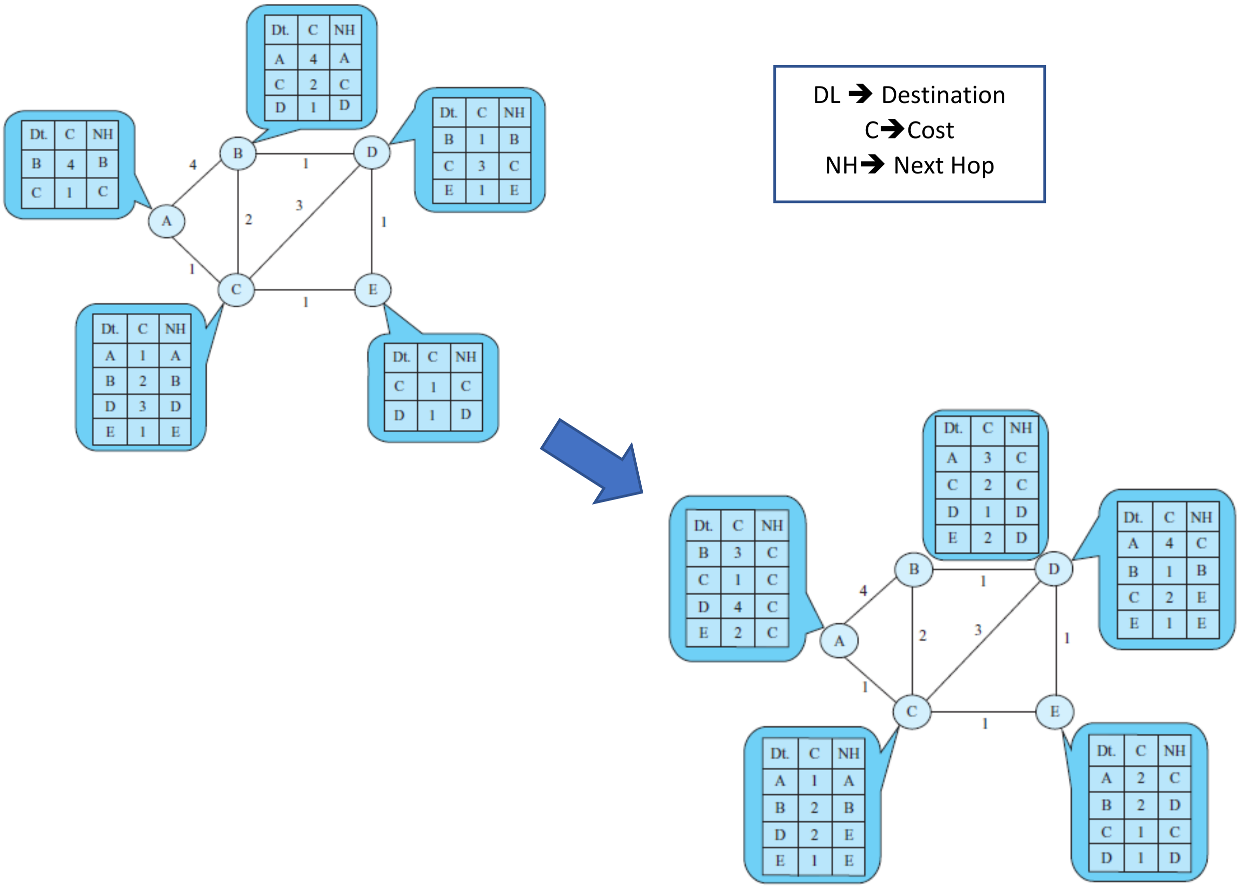
the distance vector algorithm is an *asynchronous* , *distributed* algorithm that uses *local* information. It uses only information *exchanged* from the directly connected *neighbors* .

Bellman Ford Basics – Each router maintains a Distance Vector table containing the distance between itself and ALL possible destination nodes. Distances, based on a chosen metric, are computed using information from the neighbors’ distance vectors

.

Working of DVR algorithm:

- ⇨ A router transmits its distance vector to each of its neighbour in a routing packet.
- ⇨ Each router receives and saves the most recently received distance vector from each of its neighbour
- ⇨ A router recalculates its distance vector when:
    - It receives a distance vector from a neighbour containing different information than before
    - It discovers that a link to a neighbour has gone down.
    -



Dt.	C	NH
B	4	B
C	1	C

Dt.	C	NH
A	4	A
C	2	C
D	1	D

Dt.	C	NH
B	1	B
C	3	C
E	1	E

Dt.	C	NH
A	1	A
B	2	B
D	3	D
E	1	E

Dt.	C	NH
C	1	C
D	1	D

DL ➔ Destination
C➔Cost
NH➔ Next Hop

Dt.	C	NH
B	3	C
C	1	C
D	4	C
E	2	C

Dt.	C	NH
A	3	C
C	2	C
D	1	D
E	2	D

Dt.	C	NH
A	4	C
B	1	B
C	2	E
E	1	E

Dt.	C	NH
A	1	A
B	2	B
D	2	E
E	1	E

Dt.	C	NH
A	2	C
B	2	D
C	1	C
D	1	D

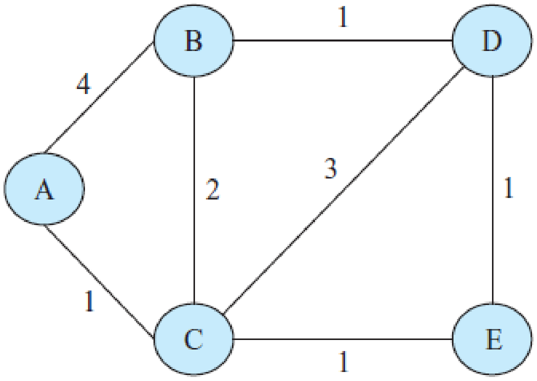
Link State Routing(LSR):

While distance vector routers use a distributed algorithm to compute their routing tables, link-state routing uses link-state routers to exchange messages that allow each router to learn the entire network topology.

To find shortest path, each node need to run the famous **Dijkstra algorithm**. The Dijkstra's algorithm is an iterative, and it has the property that after kth iteration of the algorithm, the least cost paths are well known for k destination nodes.

- **C(v):** It defines the cost of the path from source code to destination v that has the least cost currently.
- **p(v):** It defines the previous node (neighbor of v) along with current least cost path from source to v.

Example:



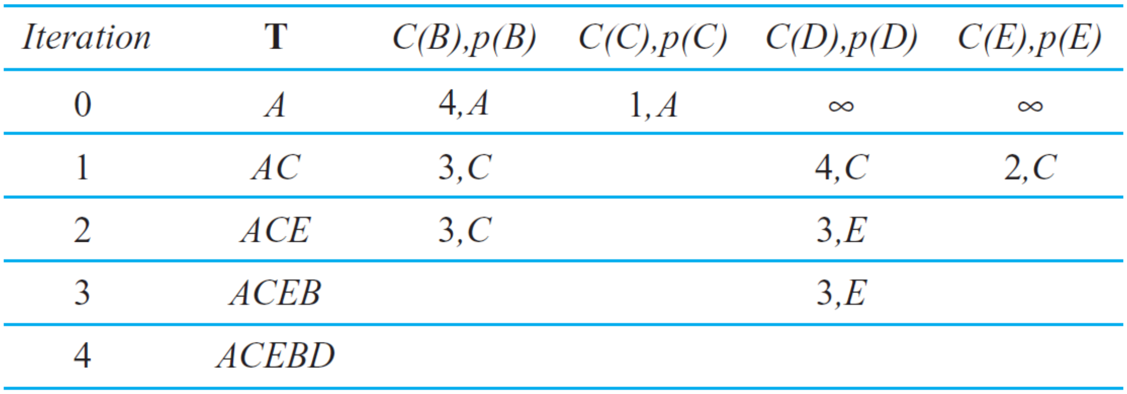
Step 1: The first step is an initialization step. The currently known least cost path from A to its directly attached neighbours, B, C are 4, 1 respectively. The cost from A to B is set to 4, from and from A to C is set to 1. The cost from A to D and E are set to infinity as they are not directly linked to A.

Step 2: we observe that vertex C had the least cost path in step 1 therefore it is added to T.we need to determine least cost path through C

Step 3 : From C to B ,D, E the cost of paths are 3, 4, 2 respectively hence E has the least cost path therefore it is added in T.

Step 4: From E to B and D the cost of paths are 3, 3 (A➔C➔B) and (A➔C➔E➔D) since there is a Tie between them we choose any node randomly Thus B is added to T

Step 5: Now only D is left thus it is added in the T.



Iteration	T	*C(B),p(B)*	*C(C),p(C)*	*C(D),p(D)*	*C(E),p(E)*
0	*A*	4,*A*	1,*A*	∞	∞
1	*AC*	3,*C*		4,*C*	2,*C*
2	*ACE*	3,*C*		3,*E*	
3	*ACEB*			3,*E*	
4	*ACEBD*				

Border Gateway Protocol(BGP):

A BGP is used to Exchange routing information for the internet and is the protocol used between ISP which are different ases

The protocol can connect together any internetwork of autonomous system using an arbitrary topology. The only requirement is that each AS have at least one router that is able to run BGP and that is router connect to at least one other AS's BGP router.

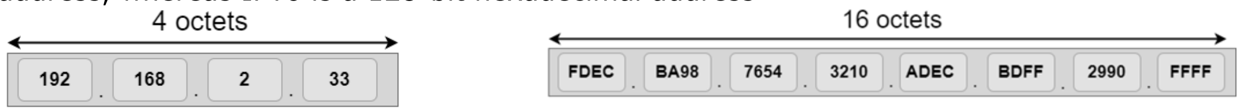
The main role of BGP is to provide communication between two autonomous systems. BGP supports Next-Hop Paradigm and support CIDR.

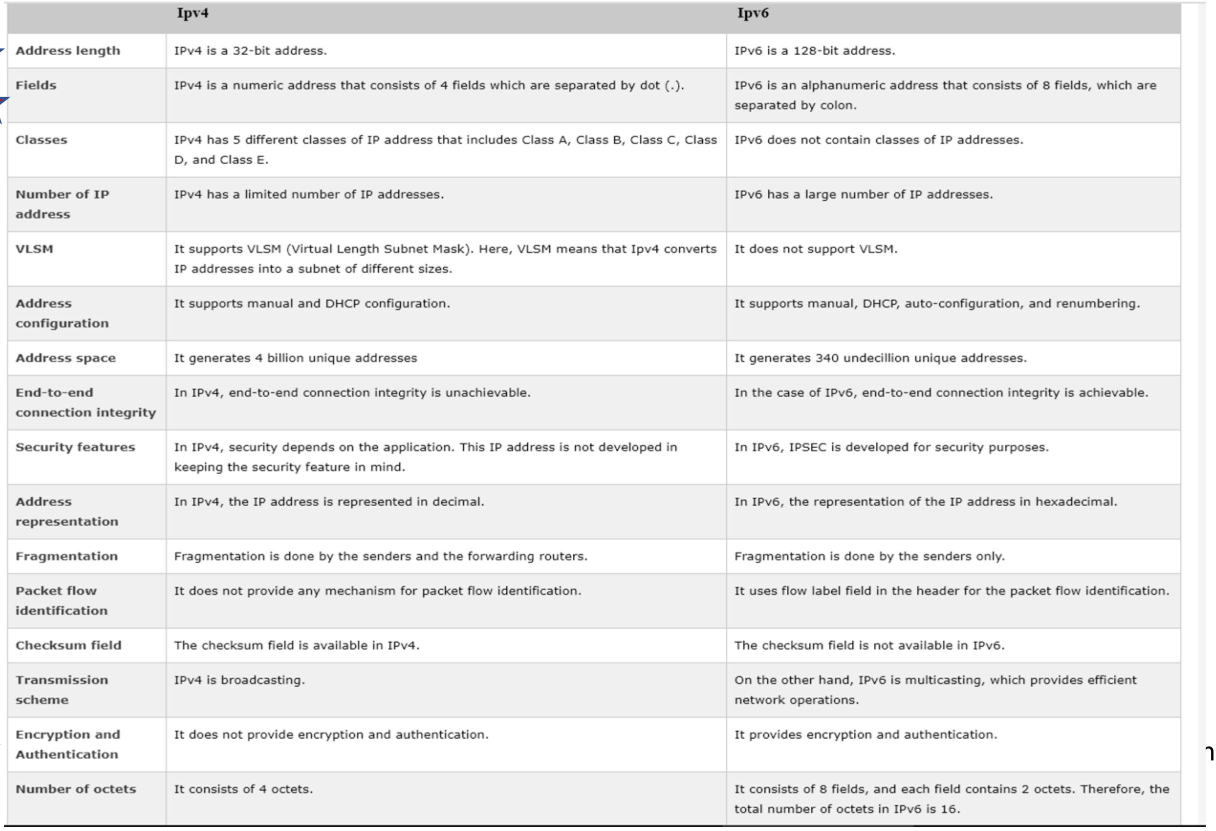
Function of BGP:

- ⇨ The first function consist of initial peer acquisition and authentication. both the peers established a TCP connection and perform message exchange that guarantees both sides have agreed to communicate.
- ⇨ The second function mainly focus on sending of negative or positive reach-ability information.
- ⇨ The third function verifies that the peers and the network connection between them are functioning correctly.

IPv6:

IPv6 is the next generation of IP addresses.it is the upgraded version of IPv4. The main difference between IPv4 and IPv6 is the address size of IP addresses. The IPv4 is a 32-bit address, whereas IPv6 is a 128-bit hexadecimal address





	Ipv4	Ipv6
Address length	IPv4 is a 32-bit address.	IPv6 is a 128-bit address.
Fields	IPv4 is a numeric address that consists of 4 fields which are separated by dot (.).	IPv6 is an alphanumeric address that consists of 8 fields, which are separated by colon.
Classes	IPv4 has 5 different classes of IP address that includes Class A, Class B, Class C, Class D, and Class E.	IPv6 does not contain classes of IP addresses.
Number of IP address	IPv4 has a limited number of IP addresses.	IPv6 has a large number of IP addresses.
VLSM	It supports VLSM (Virtual Length Subnet Mask). Here, VLSM means that Ipv4 converts IP addresses into a subnet of different sizes.	It does not support VLSM.
Address configuration	It supports manual and DHCP configuration.	It supports manual, DHCP, auto-configuration, and renumbering.
Address space	It generates 4 billion unique addresses	It generates 340 undecillion unique addresses.
End-to-end connection integrity	In IPv4, end-to-end connection integrity is unachievable.	In the case of IPv6, end-to-end connection integrity is achievable.
Security features	In IPv4, security depends on the application. This IP address is not developed in keeping the security feature in mind.	In IPv6, IPSEC is developed for security purposes.
Address representation	In IPv4, the IP address is represented in decimal.	In IPv6, the representation of the IP address in hexadecimal.
Fragmentation	Fragmentation is done by the senders and the forwarding routers.	Fragmentation is done by the senders only.
Packet flow identification	It does not provide any mechanism for packet flow identification.	It uses flow label field in the header for the packet flow identification.
Checksum field	The checksum field is available in IPv4.	The checksum field is not available in IPv6.
Transmission scheme	IPv4 is broadcasting.	On the other hand, IPv6 is multicasting, which provides efficient network operations.
Encryption and Authentication	It does not provide encryption and authentication.	It provides encryption and authentication.
Number of octets	It consists of 4 octets.	It consists of 8 fields, and each field contains 2 octets. Therefore, the total number of octets in IPv6 is 16.

DATA LINK LAYER

Notes by : Samundar singh

Data Link Layer:

Data link layer works between two hosts which are directly connected in some sense. This direct connection could be point to point or broadcast.

Data link layer is responsible for converting data stream to signals bit by bit and to send that over the underlying hardware. At the receiving end, Data link layer picks up data from hardware which are in the form of electrical signals, assembles them in a recognizable frame format, and hands over to upper layer.

The main responsibility of the Data Link Layer is to transfer the datagram across an individual link

Function of Data Link Layer:

Framing

Data-link layer takes packets from Network Layer and encapsulates them into Frames.Then, it sends each frame bit-by-bit on the hardware. At receiver' end, data link layer picks up signals from hardware and assembles them into frames.

Addressing

Data-link layer provides layer-2 hardware addressing mechanism. Hardware address is assumed to be unique on the link. It is encoded into hardware at the time of manufacturing.

Synchronization

When data frames are sent on the link, both machines must be synchronized in order to transfer to take place.

Error Control

Sometimes signals may have encountered problem in transition and the bits are flipped.These errors are detected and attempted to recover actual data bits. It also provides error reporting mechanism to the sender.

Flow Control

Stations on same link may have different speed or capacity. Data-link layer ensures flow control that enables both machine to exchange data on same speed.

Multi-Access

When host on the shared link tries to transfer the data, it has a high probability of collision. Data-link layer provides mechanism such as CSMA/CD to equip capability of accessing a shared media among multiple Systems.

Framing in Data Link Layer:

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.

Types of 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 –

1. **Length field –** We can introduce a length field in the frame to indicate the length of the frame. Used in **Ethernet(802.3)**. The problem with this is that sometimes the length field might get corrupted.
2. **End Delimeter (ED) –** We can introduce an ED(pattern) to indicate the end of the frame. Used in **Token Ring**. The problem with this is that ED can occur in the data. 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.

Point-to-Point Protocol(PPP)

Point - to - Point Protocol (PPP) is a communication protocol of the data link layer that is used to transmit multiprotocol data between two directly connected (point-to-point) computers.

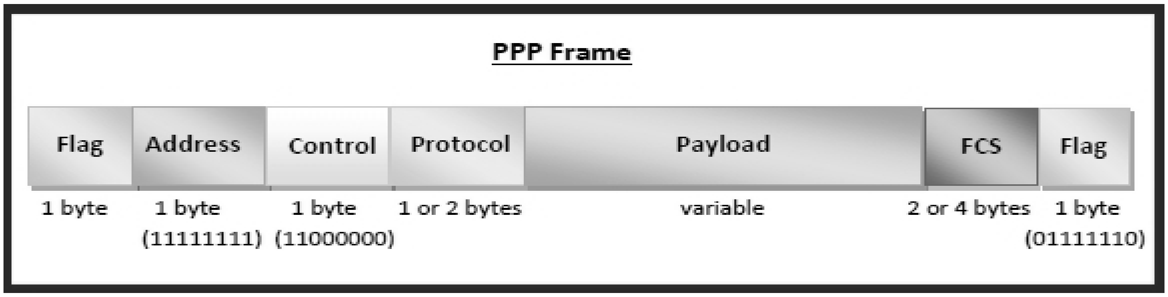
Uses:

It is widely used in broadband communications having heavy loads and high speed. For example, an internet operates on heavy load and high speed.

It is used to transmit the multiprotocol data between the two connected (point-to-point) computers. It is mainly used in point-to-point devices, for example :routers

Services provided by PPP:

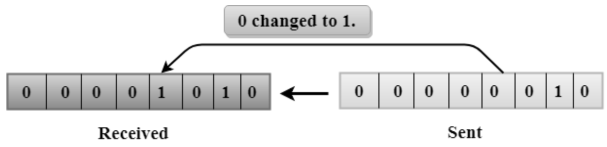
- It defines the format of frames through which the transmission occurs.
- It defines data exchange process, i.e., how data will be exchanged, the rate of the exchange.
- Stating authentication rules of the communicating devices.
- The main feature of the PPP protocol is the encapsulation.It defines how network layer data and information in the payload are encapsulated in the data link frame



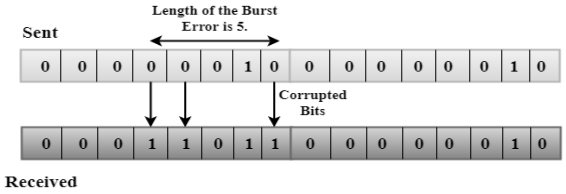
PPP connections deliver or transmit packets in sequence and provide full-duplex simultaneous bi-directional operation. PPP usually encapsulates any of the network layer packets in its frame that makes it possible for PPP layer three protocol to become independent and even capable of carrying multiple-layer three packets through a single link or connection.

Error in Networks:

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



Single-Bit Error



Burst Error

HDLC Error Detection Technique:

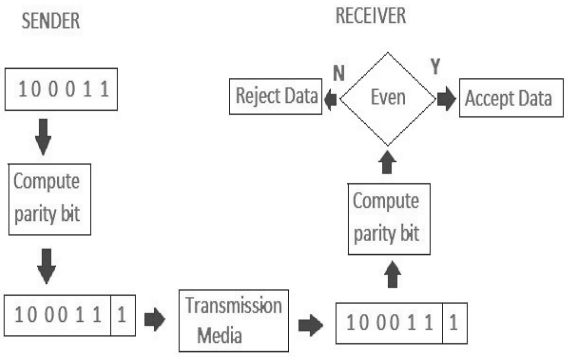
Types:

Simple Parity check **2-d Parity Check** **Checksum** **Cyclic Redundancy Check**

1. Simple parity check:

In this technique, A parity bit is appended at the end of the data unit so that the number of 1s becomes even. Therefore, the total number of transmitted bits would be 9 bits.

- 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



Drawbacks:

1. It can only detect single-bit errors which are very rare.

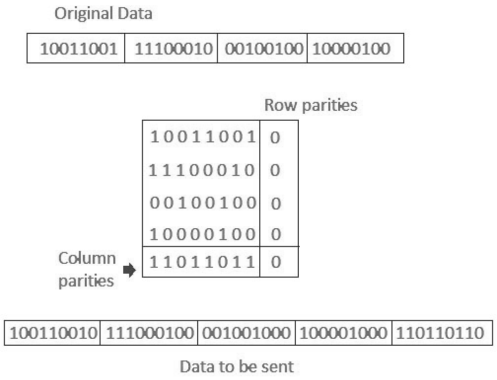
2. If two bits are interchanged, then it cannot detect the errors.

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.

Drawbacks:

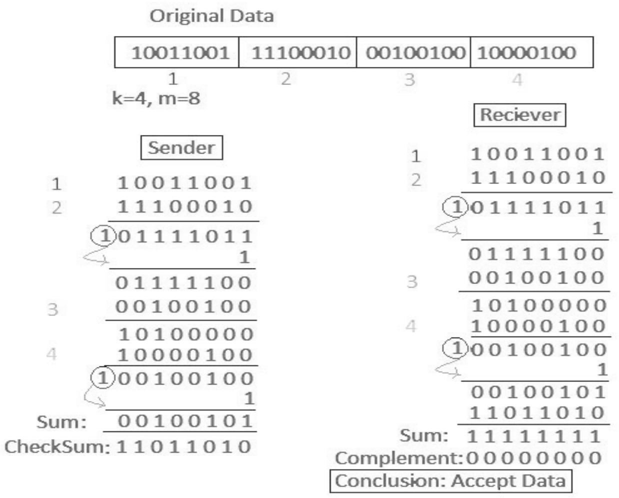
If two bits in one data unit are corrupted and two bits exactly the same position in another data unit are also corrupted, then 2D Parity checker will not be able to detect the error.



Notes by : Samundar singh

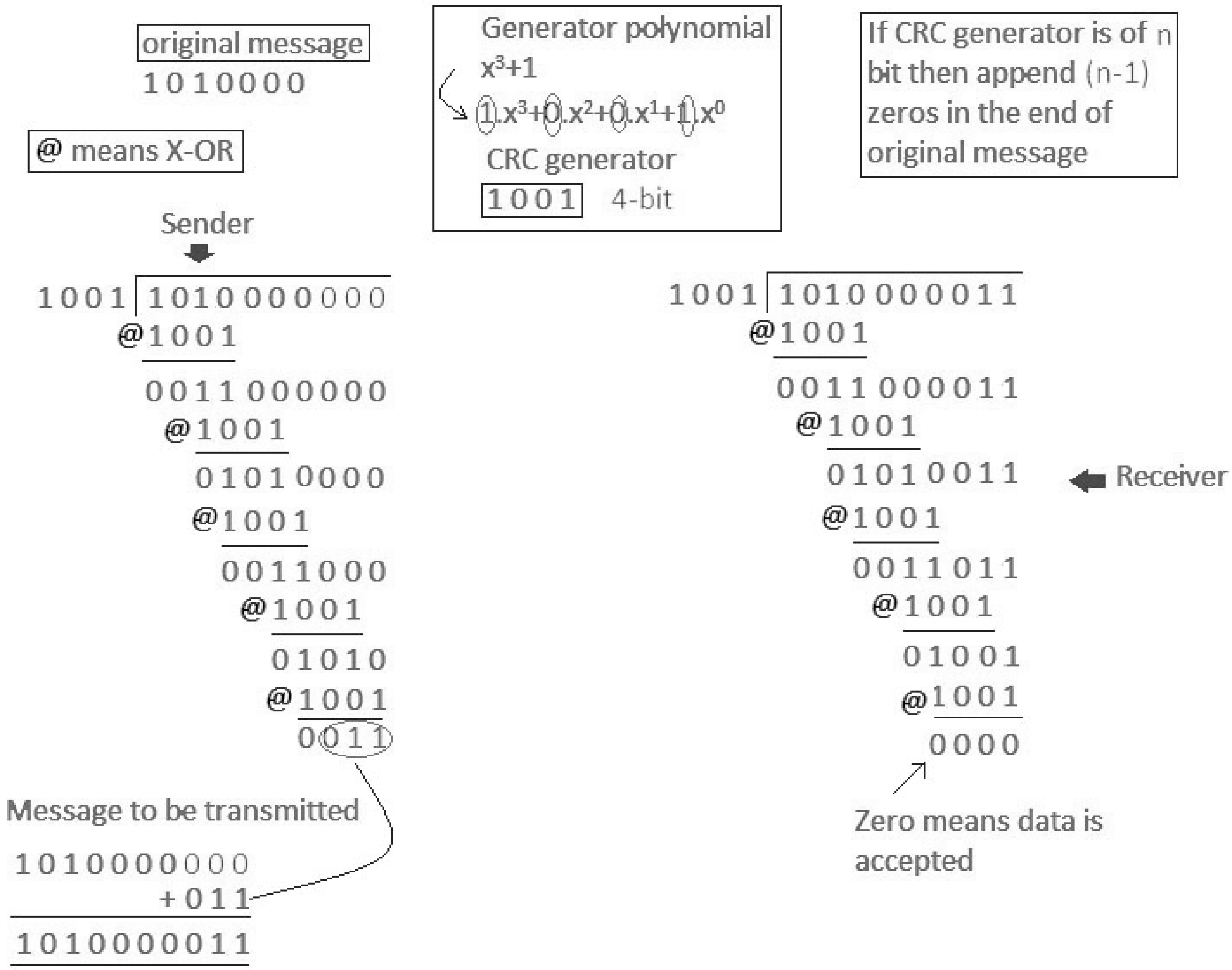
3. CheckSum:

- 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):

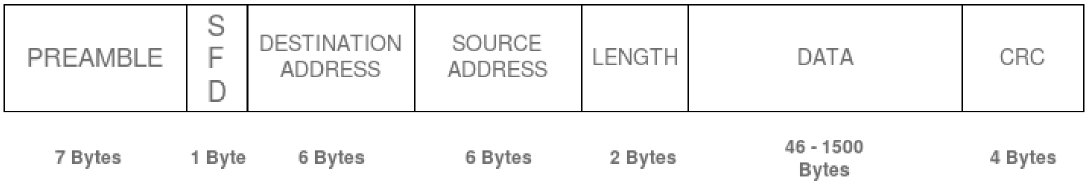
- CRC is based on binary division. n CRC.
- A sequence of redundant bits, called cyclic redundancy check bits, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number.
- At the destination, the incoming data unit is divided 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.



Ethernet(IEEE 802.3)

Ethernet is a set of technologies and protocols that are used primarily in LANs. It was first standardized in 1980s by IEEE 802.3 standard. IEEE 802.3 defines the physical layer and the medium access control (MAC) sub-layer of the data link layer for wired Ethernet networks.

Frame format of IEEE 802.3:



PREAMBLE	SFD	DESTINATION ADDRESS	SOURCE ADDRESS	LENGTH	DATA	CRC
7 Bytes	1 Byte	6 Bytes	6 Bytes	2 Bytes	46 - 1500 Bytes	4 Bytes

IEEE 802.3 ETHERNET Frame Format

- **Preamble**: It is the starting field that provides alert and timing pulse for transmission. Ethernet frame starts with 7-Bytes Preamble. PRE (Preamble) indicates the receiver that frame is coming and allow the receiver to lock onto the data stream before the actual frame begins.

- **Start of frame delimiter (SFD) :** It is a 1 byte field in a IEEE 802.3 frame that contains an alternating pattern of ones and zeros ending with two ones(always set to 10101011). The SFD warns station or stations that this is the last chance for synchronization.

- **Destination Address :** It is a 6 byte field containing physical address(MAC) of destination stations.

- **Source Address:** It is a 6 byte field containing the physical address(MAC) of the sending station

- **Length:** Length is a 2-Byte field, which indicates the length of entire Ethernet frame.

- **Data:** This is the place where actual data is inserted. both IP header and data will be inserted here. The maximum data present may be as long as 1500 Bytes.

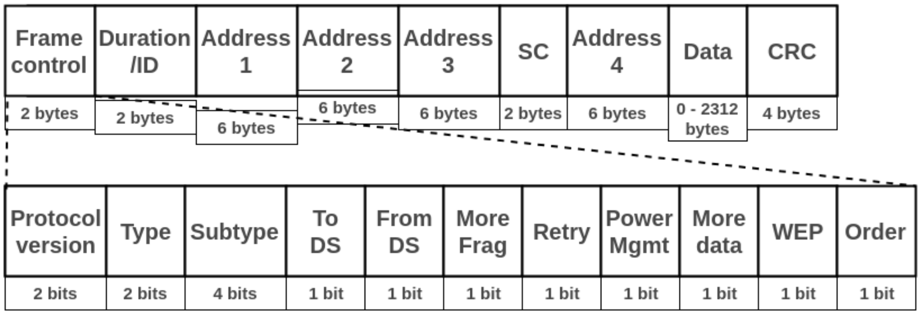
- **Cyclic Redundancy Check (CRC):** CRC is 4 Byte field. This field contains a 32-bits hash code of data, which is generated over the Destination Address, Source Address, Length, and Data field. If the checksum computed by destination is not the same as sent checksum value, data received is corrupted.

Wi-Fi(wireless-fidelity)

it allows an electronic device to transfer data or connect to the internet using ISM radio bands. It is an underlying technology of wireless local area network (WLAN). Wi-Fi allows computers and other devices to communicate over a wireless network. its technology for wireless local area networking with devices based on IEEE 802.11 standards.

Wi-Fi compatible devices can connect to the internet via WLAN network and a wireless **access point** abbreviated as AP. Every WLAN has an access point which is responsible for receiving and transmitting data from/to use rs.

Architecture of IEEE 802.11:



Frame control	Duration /ID	Address 1	Address 2	Address 3	SC	Address 4	Data	CRC
2 bytes	2 bytes	6 bytes	6 bytes	6 bytes	2 bytes	6 bytes	0 - 2312 bytes	4 bytes

Protocol version	Type	Subtype	To DS	From DS	More Frag	Retry	Power Mgmt	More data	WEP	Order
2 bits	2 bits	4 bits	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit

IEEE 802.11 MAC Frame Structure

1. Frame Control(FC) –
It is 2 bytes long field which defines type of frame and some control information. Various fields present in FC are:

Protocol version: t is a 2 bit long field which indicates the current protocol version which is fixed to be 0 for now.
Type: It is a 2 bit long field which determines the function of frame i.e management(00), control(01) or data(10).
Subtype : It is a 4 bit long field which indicates sub-type of the frame like 0000 for association request, 1000 for beacon.
To DS: It is a 1 bit long field which when set indicates that destination frame is for DS(distribution system).
From DS; It is a 1 bit long field which when set indicates frame coming from DS
More Frag: It is 1 bit long field which when set to 1 means frame is followed by other fragments.
Retry: It is 1 bit long field, if the current frame is a retransmission of an earlier frame, this bit is set to 1.
WEP: It is 1 bit long field which indicates that the standard security mechanism of 802.11 is applied.

2. Duration ID –
It is 4 bytes long field which contains the value indicating the period of time in which the medium is occupied(in µs).

3. Address 1 to 4 –
These are 6 bytes long fields which contain standard IEEE 802 MAC addresses (48 bit each). The meaning of each address depends on the DS bits in the frame control field.

4. SC (Sequence control) –
It is 16 bits long field which consists of 2 sub-fields, i.e., Sequence number (12 bits) and Fragment number (4 bits). Since acknowledgement mechanism frames may be duplicated hence, a sequence number is used to filter duplicate frames.

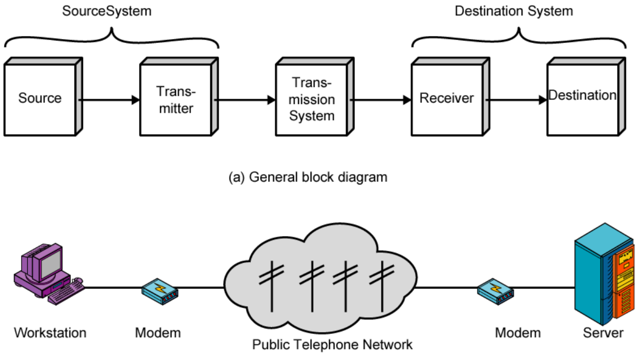
5. Data –
It is a variable length field which contain information specific to individual frames which is transferred transparently from a sender to the receiver(s).

6. CRC (Cyclic redundancy check) –
It is 4 bytes long field which contains a 32 bit CRC error detection sequence to ensure error free frame.

Fundamental of data communication

Notes by : Samundar singh

A communication Model:



(a) General block diagram

(b) Example

The fundamental purpose of a communications system is the exchange of data between two parties.

The key elements of this model are:

- Source - generates data to be transmitted
- Transmitter - converts data into transmittable signals
- Transmission System - carries data from source to destination
- Receiver - converts received signal into data
- Destination - takes incoming data

Communication Task:

The key tasks that must be performed in a data communications system:

- **transmission system utilization** - need to make efficient use of transmission facilities typically shared among a number of communicating devices a device must **interface** with the transmission system once an interface is established, **signal generation** is required for communication there must be **synchronization** between transmitter and receiver, to determine when a signal begins to arrive and when it ends there is a variety of requirements for communication between two parties that might be collected under the term **exchange management**

- **Error detection and correction** are required in circumstances where errors cannot be tolerated

- **Flow control** is required to assure that the source does not overwhelm the destination by sending data faster than they can be processed and absorbed

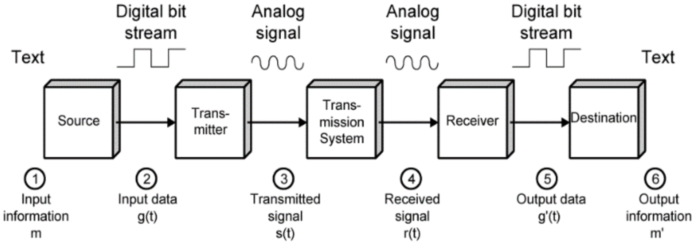
- **addressing** and **routing**, so a source system can indicate the identity of the intended destination, and can choose a specific route through this network

- **Recovery** allows an interrupted transaction to resume activity at the point of interruption or to condition prior to the beginning of the exchange

- **Message formatting** has to do with an agreement between two parties as to the form of the data to be exchanged or transmitted Frequently need to provide some measure of **security** in a data communications system

- **Network management** capabilities are needed to configure the system, monitor its status, react to failures and overloads, and plan intelligently for future growth See have gone from the simple idea of data communication between source and destination to a rather formidable list of data communications tasks.

Data Communication Model:



In order to understand data communication we have a example , Assume a PC user wants to send an email message m to another user.

- user keys in message m comprising bits g buffered in source PC memory
- input data is transferred to I/O device (transmitter) as sequence of bits g(t) using voltage shifts
- transmitter converts these into a signal s(t) suitable for transmission media being used
- whilst transiting media signal may be impaired so received signal r(t) may differ from s(t)
- receiver decodes signal recovering g'(t) as estimate of original g(t)
- which is buffered in destination PC memory as bits g' being the received message m'

Networks:

The **Network** allows computers to **connect and communicate** with different computers via any medium. LAN, MAN and WAN are the three major types of the network designed to operate over the area they cover.

Local Area Network (LAN):

LAN is used to connect two devices or computer for data sharing purpose. The group of computers and devices are connected together by a switch. LANs cover smaller geographical area (Size is limited to a few kilometres) and are privately owned. A Communication medium used for LAN has twisted pair cables and coaxial cables. It covers a short distance, and so the error and noise are minimized.

Metropolitan Area Network (MAN):

MAN or Metropolitan area Network covers a larger area than that of a LAN and smaller area as compared to WAN. It connects two or more computers that are apart but resides in the same or different cities,it may be served as a Internet Service Provider. Devices used for transmission of data through MAN are: Modem and Wire/Cable.

Wide Area Network (WAN):

WAN is a computer network that extends over a large geographical area. A WAN could be a connection of LAN connecting to other LAN's via telephone lines and radio waves. Devices used for transmission of data through WAN are: Optic wires, Microwaves and Satellites.

Example of a Switched WAN is the asynchronous transfer mode (ATM) network and Point-to-Point WAN is dial-up line that connects a home computer to the Internet.

Circuit switching:

In circuit switching network resources (bandwidth) is divided into pieces Telephone system network is the one of example of Circuit switching.

Methods:

1. Frequency Division Multiplexing (*Divides into multiple bands):* It is a technique by which the total bandwidth is divided into a series of non-overlapping frequency sub-bands,where each sub-band carry different signal.

Time Division Multiplexing (*Divides into frames):* it is a method of transmitting and receiving independent signals over a common signal path by means of synchronized switches at each end of the transmission line. It is used for long-distance communication

FOR GATE Purpose:

Circuit switching formula :

Transmission rate = Link Rate or Bit rate / no. of slots = R/h bps

Transmission time = size of file / transmission rate = x / (R/h) = (x*h)/R second

Total time to send packet to destination = Transmission time + circuit setup time

Example 1 : How long it takes to send a file of 'x bits' from host A to host B over a circuit switched network that uses TDM with 'h slots' and have a bit rate of 'R Mbps', circuit establish time is k seconds.Find total time?
Explanation :
Transmission rate = Link Rate or Bit rate / no. of slots = R/h bps
Transmission time = size of file/ transmission rate = x / (R/h) = (x*h)/R
Total time = transmission time + circuit setup time = (x*h)/R secs + k secs

Packet switching:

Packet switching is a method of transferring the data to a network in form of packets the data is broken into small pieces of variable length, called **Packet**. At the destination, all these small-parts (packets) has to be reassembled, belonging to the same file.

Packet Switching uses **Store and Forward** technique while switching the packets; while forwarding the packet each hop first store that packet then forward.

Methods:

1. Connection-oriented Packet Switching (Virtual Circuit): Before starting the transmission, it establishes a logical path or virtual connection using signalling protocol, between sender and receiver and all packets belongs to this flow will follow this predefined route.

2. Connectionless Packet Switching (Datagram) : Unlike Connection-oriented packet switching, In Connectionless Packet Switching each packet contains all necessary addressing information such as source address, destination address and port numbers etc In Datagram Packet Switching, each packet is treated independently. Packets belonging to one flow may take different routes because routing decisions are made dynamically, so the packets arrived at destination might be out of order

Frame relay:

Frame relay is a packet-switching telecommunication service designed for cost-efficient data transmission for intermittent traffic between local area networks (LANs) and between endpoints in wide area networks (WANs). Txhe original packet-switching networks were designed with a ~~data rate to the end user of about 64 kbps, frame relay networks are designed to operate efficiently at user data rates of up to 2 Mbps~~. The key to achieving these high data rates is to strip out most of the overhead involved with error control.

ATM(Asynchronous Transfer Mode):

ATM stands for Asynchronous Transfer Mode. It is a switching technique that uses time division multiplexing (TDM) for data communications. ATM networks are connection oriented networks for cell relay that supports voice, video and data communications. It encodes data into small fixed - size cells so that they are suitable for TDM and transmits them over a physical medium.

Cell Format – each cell is 53 bytes long which consists of 5 bytes header and 48 bytes payload.

Data Transmission:

Transmission media

a transmission medium is a physical path between the transmitter and the receiver i.e it is the channel through which data is sent from one place to another.

Types transmission media:

1.Guided media: It is also referred to as Wired transmission media. Signals being transmitted by using physical links. It provide secure and high speed data transmission and used for shorter distance.

There are 3 major types of Guided Media:

a. **Twisted Pair Cable –** consists of 2 separately insulated conductor wires wound about each other.
b. **Coaxial Cable –** t has an outer plastic covering containing 2 parallel conductors each having a separate insulated protection cover
c. **Optical Fibre Cable –** It uses the concept of reflection of light through a core made up of glass or plastic. The core is surrounded by a less dense glass or plastic covering called the cladding.

2.Un-Guided media: t is also referred to as Wireless or Unbounded transmission media.it is less secure and used for long distance.

There are 3 major types of Unguided Medi

a. **Radiowaves**
b. **Microwaves**
c. **Infrared**

Transmission Modes:

Types transmission media:

1. **Simplex Mode:** In Simplex mode, the communication is unidirectional, Only one of the two devices on a link can transmit, the other can only receive. The simplex mode can use the entire capacity of the channel to send data in one direction. **Example: Keyboard**

2. **Half Duplex Mode:** In half-duplex mode, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa

Example: Walkie- talkie

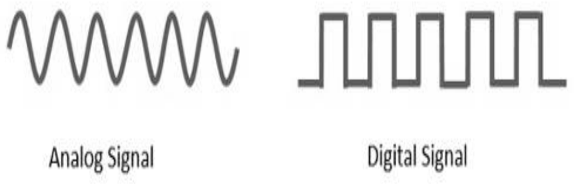
3. **Full Duplex Mode:** In full-duplex mode, both stations can transmit and receive simultaneously. In full duplex mode, signals going in one direction share the capacity of the link with signals going in other direction this sharing can occur in two ways:

- Either the link must contain two physically separate transmission paths, one for sending and other for receiving.
- Or the capacity is divided between signals travelling in both directions

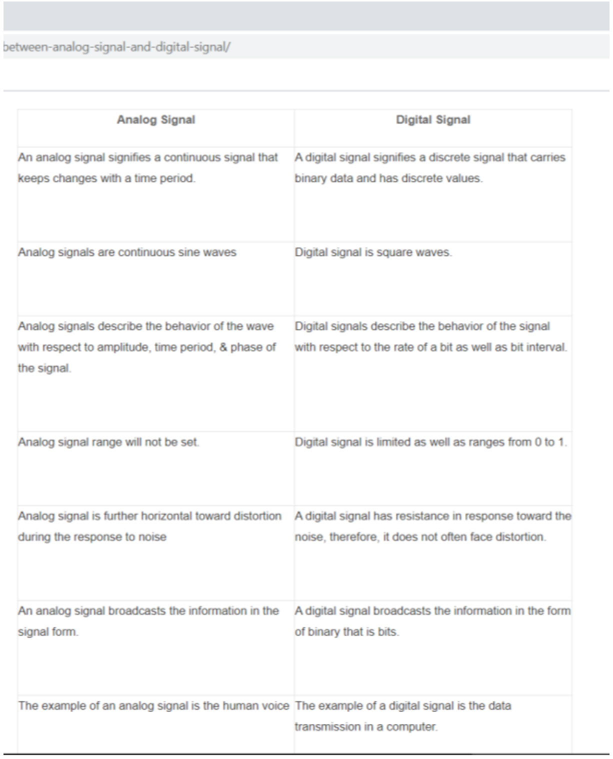
Example: Telephone Network

Signals:

There are 2 types of signal in networks that is analog and digital signal for better understanding let us compare these 2 signals :



Representation of Signals



Analog Signal	Digital Signal
An analog signal signifies a continuous signal that keeps changes with a time period.	A digital signal signifies a discrete signal that carries binary data and has discrete values.
Analog signals are continuous sine waves	Digital signal is square waves.
Analog signals describe the behavior of the wave with respect to amplitude, time period, & phase of the signal.	Digital signals describe the behavior of the signal with respect to the rate of a bit as well as bit interval.
Analog signal range will not be set.	Digital signal is limited as well as ranges from 0 to 1.
Analog signal is further horizontal toward distortion during the response to noise	A digital signal has resistance in response toward the noise, therefore, it does not often face distortion.
An analog signal broadcasts the information in the signal form.	A digital signal broadcasts the information in the form of binary that is bits.
The example of an analog signal is the human voice	The example of a digital signal is the data transmission in a computer.

Periodic signal:

Channel Capacity:

The maximum rate at which data can be transmitted over a given communication channel, under given conditions, is referred to as the **channel capacity.** Parts of Channel Capacity.

- **Data rate**, in bits per second (bps), at which data can be communicated

- **Bandwidth,** as constrained by the transmitter and the nature of the transmission medium, expressed in cycles per second, or Hertz

- **Noise,** average level of noise over the communications path

- **Error rate,** at which errors occur, where an error is the reception of a 1 when a 0 was transmitted or the reception of a 0 when a 1 was transmitted

pending

Transmission Impairments:

In communication system, analog signals travel through transmission media due to which there is deterioration in the quality of the signal due to which received signal is not same as the signal that was send.

Cause of impairments:

➔**Attenuation: –** It means loss of energy. The strength of signal decreases with increasing distance which causes loss of energy. Attenuation is measured in **decibels(dB)**

Attenuation introduces three considerations for the transmission:

1. A received signal must have sufficient strength so that the electronic circuitry in the receiver can detect the signal

2. The signal must maintain a level sufficiently higher than noise to be received without error.

3. Attenuation varies with frequency

➔**Distortion –** It means change in the shape of signal. This is generally seen in composite signals with different frequencies. Each frequency component has its own propagation speed travelling through a medium. Every component arrive at different time which leads to delay distortion.

➔ **Noise:** The random or unwanted signal that mixes up with the original signal is called noise. here are several types of noise like:

- **Thermal noise:** it is movement of electrons in wire which creates an extra signal. It is present in all electronic devices and transmission media and is a function of temperature

- **Crosstalk noise:** is when one wire affects the other wire. It can occur by electrical coupling between nearby twisted pairs or, rarely, coax cable lines carrying multiple signals. It can also occur when microwave antennas pick up unwanted signals

- **Impulse noise:** it is a signal with high energy that comes from lightning or power lines , consisting of irregular pulses or noise spikes of short duration and of relatively high amplitude.

Some Terms:

Unipolar - All signal elements have the same sign

Polar - One logic state represented by positive voltage the other by negative voltage

Data rate - Rate of data (R) transmission in bits per second

Duration or length of a bit - Time taken for transmitter to emit the bit (1/R)

Modulation rate -Rate at which the signal level changes, measured in baud = signal elements per second. Depends on type of digital encoding used.

Mark and Space - Binary 1 and Binary 0 respectively

Signal Encoding Technique:

Both analog and digital information can be encoded as either analog or digital signals. So there are 4 possible combination of data encoding i.e:

1. Digital data – to – Digital Signal:
2. Digital data – to –Analog Signal:
3. Analog data – to – Digital Signal:
4. Analog data – to – Analog Signal:

Process:

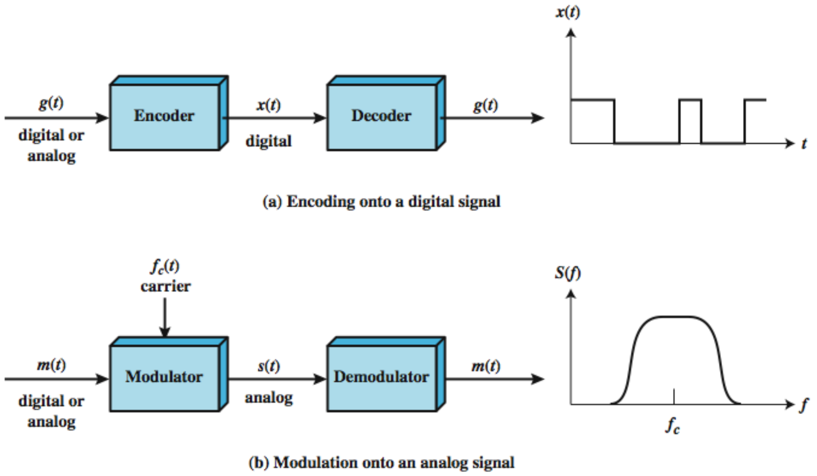
1. For **digital signaling**, a data source $g(t)$, which may be either digital or analog, is encoded into a digital signal $x(t)$.
2. The basis for **analog signaling** is a continuous constant-frequency f_c signal known as the **carrier signal**. Data may be transmitted using a carrier signal by modulation.

Modulation : process of encoding source data onto the carrier signal

3. All modulation techniques involve operation on one or more of the three fundamental frequency parameter:
   - → **Amplitude**
   - → **Frequency**
   - → **Phase**

Encoding:(In the sense of Digital data to Digital Signal)

- A digital signal is a sequence of discrete, discontinuous voltage pulses, Each pulse is a signal element
- Binary data are transmitted by encoding each data bit into signal elements.
- There is a one-to-one correspondence between bits and signal elements.
- More complex encoding schemes are used to improve performance, by altering the spectrum of the signal and providing synchronization capability



(a) Encoding onto a digital signal

(b) Modulation onto an analog signal

Figure 5.1 Encoding and Modulation Techniques

Interpreting signal:

It involve following process :

- First, the receiver must know the timing of each bit, knowing with some accuracy when a bit begins and ends
- Second, the receiver must determine whether the signal level for each bit position is high (0) or low These tasks can be performed by sampling each bit position in the middle of the interval and comparing the value to a threshold.
- Due to noise and other impairment there will be some error occur
- Encoding schema is used improve performance it is simply the mapping from data bits to signal elements.

Encoding Schema:

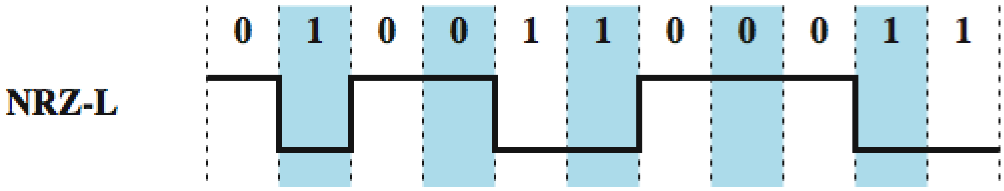
Some important terms:

- Signal Spectrum - Lack of high frequencies reduces required bandwidth, lack of dc component allows ac coupling via transformer, providing isolation, should concentrate power in the middle of the bandwidth
- Clocking - need for synchronizing transmitter and receiver either with an external clock or with a sync mechanism based on signal
- Error detection - useful if can be built in to signal encoding
- Signal interference and noise immunity - some codes are better than others
- Cost and complexity - Higher signal rate (& thus data rate) lead to higher costs, some codes require signal rate greater than data rate

The various Encoding Schemas are as follow:

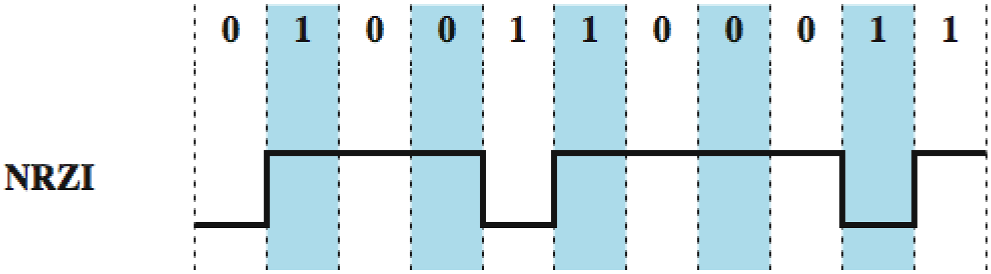
1. **Nonreturn to Zero-Level (**NRZ-L)

- It transmit digital signals to two different voltage levels for the two binary digits 0 and 1
- voltage level is constant during a bit interval and there is no transition (see no fluctuation in white and blue spaces in the bit interval)
- absence of voltage used to represent binary 0,
- Presence of voltage used to represent binary 1.
- **NRZ-L is typically the code used to generate or interpret digital** data by terminals and other devices.



2. **Nonreturn to Zero-Inverted (**NRZ-I):

- NRZI is an example of **differential encoding**. In differential encoding, the information to be transmitted is represented in terms of the changes between successive signal elements rather than the signal elements themselves.
- NRZI maintains a constant voltage pulse for the duration of a bit time.
- The encoding of the current bit is determined as follows:
    - if the current bit is a binary 0, then the current bit is encoded with the same signal as the preceding bit
    - if the current bit is a binary 1, then the current bit is encoded with a different signal than the preceding bit.



“”Yahan par clerly dikh raha hia jaise hi bit change toh wo tab tak contant bana rehta hai jab tak dubara usko same bit na mil jaye””.

Pros:

- NRZ codes are the easiest to engineer and, in addition, make efficient use of bandwidth
- Most of the energy in NRZ and NRZI signals is between dc and half the bit rate. Means energy efficient. Hai.

Cons:

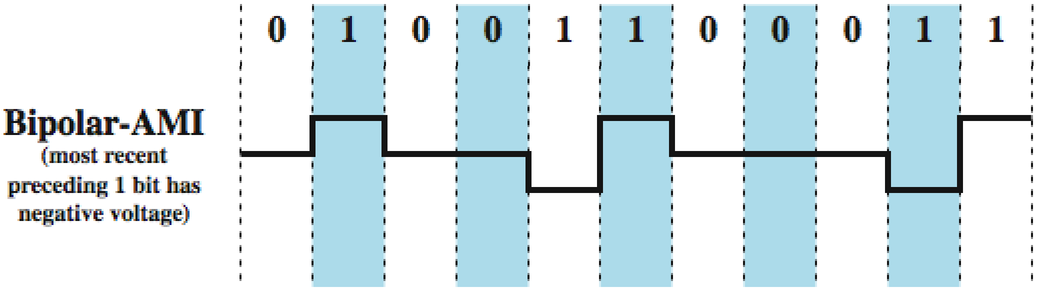
- The main limitations of NRZ signals are the presence of a dc component and the lack of synchronization capability

3.**Multilevel Binary Bipolar-AMI**

- These codes use more than two signal levels
- In the **bipolar-AMI** scheme, a binary 0 is represented by no line signal,
- Binary 1 is represented by a positive or negative pulse. The binary 1 pulses must alternate in polarity.

Pros:

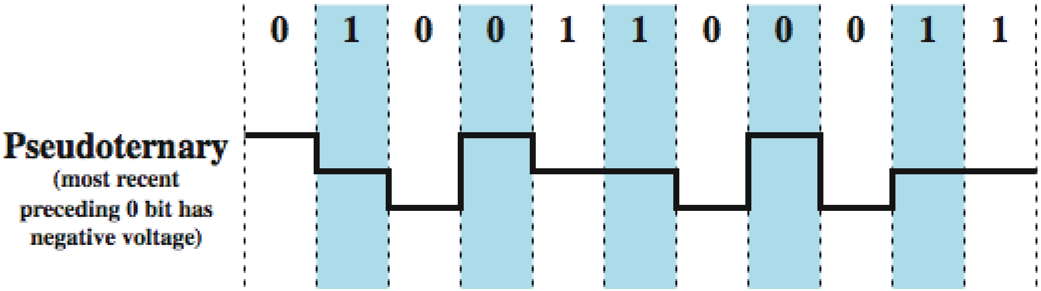
- There will be no loss of synchronization if a long string of 1s occurs
- Since 1 Signals alternate in voltage from positive to negative, there is no net dc component
- Error simply detected by means of pulse alteration property.



Notes by : Samundar singh

4.**Multilevel Binary Pseudoternary:**

- In absence of a line signal a binary 1 is represented.
- It represent 0 when there is alternating of positive and negative pulse.

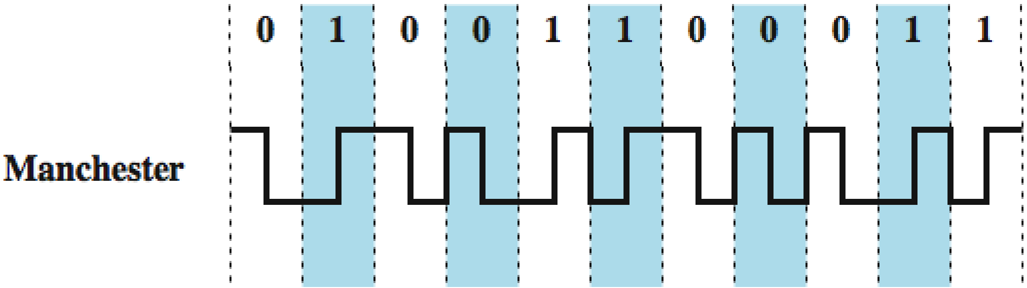


Issue:

A long string of 0s in the case of AMI or 1s in the case of pseudoternary still presents a problem

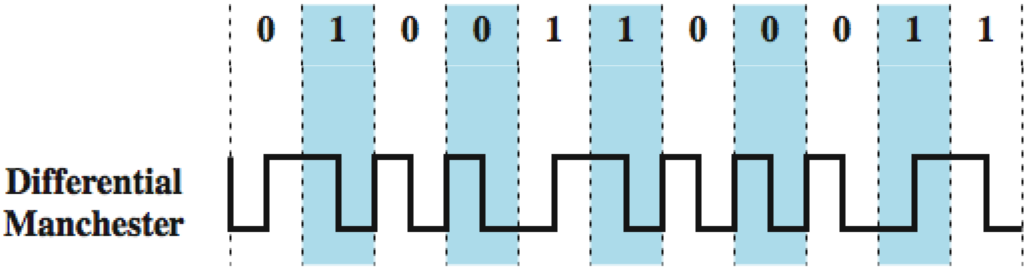
4.**Manchester Encoding(Biphase):**

- There is a transition at the middle of each bit period The midbit transition serves as a clocking mechanism and also as data
- A low-to-high transition represents a “1”
- A high-to-low transition represents a “”0



4.**Diffrential Manchester Encoding(Biphase):**

- The midbit transition is used only to provide clocking only.
- presence of a transition at the beginning of a bit period is represented by “ 0”,
- Absence of a transition at the beginning of a bit period is represented by “ 1”,



Biphase Pros and Cons:

Pros:

- **Synchronization:** Because there is a predictable transition during each bit time, the receiver can synchronize on that transition, known as self-clocking codes.
- **No dc component:** Biphase codes have no dc component
- **Error detection:** The absence of an expected transition can be used to detect errors. Noise on the line would have to invert both the signal before and after the expected transition to cause an undetected error.

Cons:

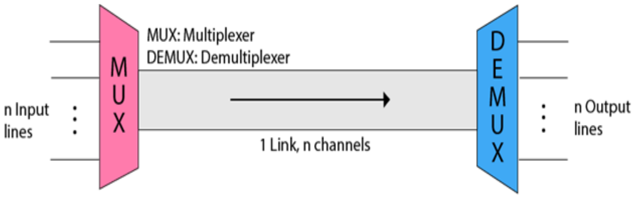
- The maximum modulation rate is twice that for NRZ;
- It require more bandwidth.

Bandwidth Utilization: Multiplexing and Spreading

- Bandwidth utilization helps to use the bandwidth in efficient manner,
- Efficiency can be achieved by multiplexing, privacy and anti-jamming

Multiplexing:

- Multiplexing is the set of techniques that allows the simultaneous transmission of multiple signals across a single data link
- It combines and send the multiple data streams over a single medium. The process of combining the data streams is known as multiplexing and hardware used for multiplexing is known as a multiplexer.



Working:

- The 'n' input lines are transmitted through a multiplexer and multiplexer combines the signals to form a composite signal.
- The composite signal is passed through a Demultiplexer and demultiplexer separates a signal to component signals and transfers them to their respective destinations.

Types of Multiplexing:

1.frequency division Multiplexing

2.wavelength division Multiplexing

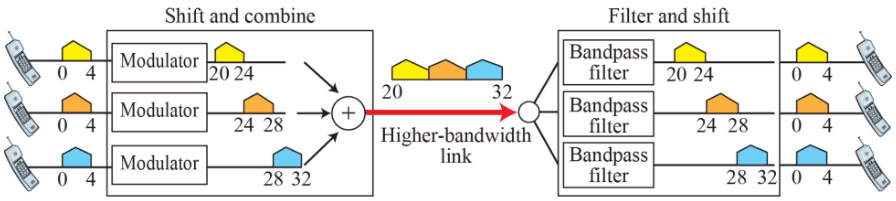
3.Time -division Multiplexing

1.Frequency division Multiplexing (FDM):

This technique combines analog signals, It can be applied when the bandwidth of a link (in hertz) is greater than the combined bandwidths of the signals to be transmitted

Q. Assume that a voice channel occupies a bandwidth of 4 kHz. We need to combine three voice channels into a link with a bandwidth of 12 kHz, from 20 to 32 kHz. Show the configuration, using the frequency domain. Assume there are no guard bands

Ans We shift (modulate) each of the three voice channels to a different bandwidth. We use the 20- to 24-kHz bandwidth for the first channel, the 24- to 28-kHz bandwidth for the second channel, and the 28- to 32-kHz bandwidth for the third on
Then we combine them:



Q. Five channels, each with a 100-kHz bandwidth, are to be multiplexed together. What is the minimum bandwidth of the link if there is a need for a guard band of 10 kHz between the channels to prevent interference?

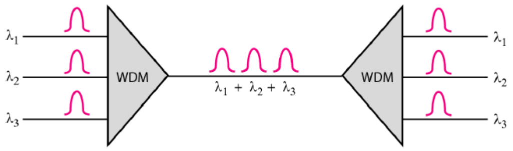
ANS: For five channels, we need at least four guard bands. This means that the required bandwidth is at least 5 × 100 + 4 × 10 = 540 kHz.

Q. The Advanced Mobile Phone System (AMPS) uses two bands. The first band of 824 to 849 MHz is used for sending, and 869 to 894 MHz is used for receiving. Each user has a bandwidth of 30 kHz in each direction. How many people can use their cellular phones simultaneously?

ANS: Each band is 25 MHz. If we divide 25 MHz by 30 kHz, we get 833.33. In reality, the band is divided into 832 channels. Of these, 42 channels are used for control,which means only 790 channels are available for cellular phone users.

2. **Wavelength division Multiplexing (WDM):**

Wavelength division multiplexing (WDM) is a technique of multiplexing multiple optical carrier signals through a single optical fiber channel by varying the wavelengths of laser lights.



Similar to prism as the white light scatter in prism in seven colour because of the different wavelength.

3. **Time- division Multiplexing (TDM):**

- It is a technique of combining several low rate channels into one high rate channel.
- It is a digital technique in which data is transmitted one-by one in form of frames.
- It is used when data transmission rate of media is greater than that of the source, and each signal is allotted a definite amount of time

It is also subdivided into two parts:

a. Synchronous TDM **b. Asynchronous TDM**

a.) Synchronous TDM : In synchronous TDM, every device which is present in this has given the same time slot to transmit data. It does not consider whether the device contains data or not. Time slots are fixed and pre-defined.

b.) Asynchronous TDM : In this, multiplexer does not allocates same time slots to each device. It also does not consider whether the device contains data or not**.** Time-slots are not pre-defined.