Data Communication II

Tilak De Silva

PREFACE

In year 1 you studied the basic principles and concepts of data communication. I mainly focused on TCP/IP in this book. I tried to present all basic concepts on TCP/IP in simple manner, by giving appropriate examples. We will mainly study the operations of TCP/IP.

Your comments, suggestions to improve this book are highly appreciated and please send them to my e-mail: tilak@slt.com.lk

Tilak De Silva
BSc Eng, C.Eng, FIESL, FIEE (UK), MBCS

Table of content

1 Introduction

1.1 What is Internet?

This is a group of connected networks all over the world. This was originated in 1967 by the US Department of Defense to interconnect their networks, which were in different locations. Later it was expanded to some universities of USA. Finally it was expanded to millions of networks dispersed all over the world.

1.2 Internet Administration

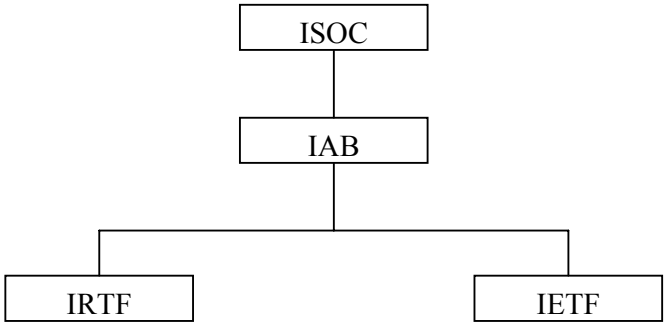


Fig. 1-1 Internet Organization

1.2.1 Internet Society

Internet Society (ISOC) is an international non-profit organization formed in 1992 to provide support for the Internet standards. It has many supporting administrative bodies such as IAB, IETF and IANA. ISOC also promotes research and other development activities related to the Internet.

1.2.2 Internet Architecture Board (IAB)

IAB is technical advisor to ISOC. It accomplishes this through IRTF and IETF. Another responsibility of IAB is the editorial management of RFCs.

The Internet standards are continuously developed. Specification of a standard begins as an Internet draft. Upon recommendation from the Internet authorities, a draft may be published as a **Request for Comment** (RFC). Each RFC is assigned a number and made available to all interested parties. After several processes an RFC will become a permanent standard of Internet.

1.2.3 Internet Engineering Task Force (IETF)

IETF is responsible for identifying operational problems and proposing solutions to these problems. It also develops and reviews specifications of Internet standards.

1.2.4 Internet Research Task Force (IRTF)

IRTF is responsible on long-term research topics related to Internet protocols, applications, architecture and technology.

1.3 Internet Protocols

TCP/IP is the protocol family used in Internet. This will be discussed in detail in other chapters. The TCP/IP protocol suite and its relationship to ISO-OSI Model is given in fig 1-2.

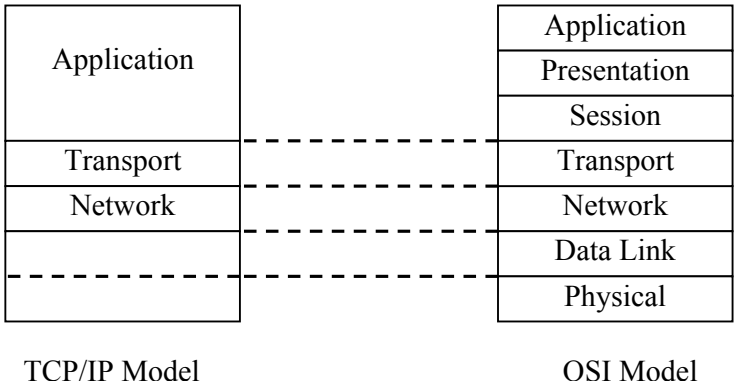


Fig. 1-2 TCP Model and OSI Model

TCP/IP does not define the physical layer and Data Link layer. It can work with any existing such layers that were defined by any other standard body such as IEEE and ITU-T.

The top most layer of TCP/IP is Application layer. It is equivalent to Session, Presentation and Application layers of OSI model.

1.3.1 Network Protocol

TCP/IP has one network protocol called Internet Protocol (IP)

1.3.2 Transport layer protocols

The Transport layer has two protocols. They are Transport Control Protocol (TCP) and User Datagram Protocol (UDP).

1.3.3 Application layer protocols

Application layer has many protocols and some of them are given in Table 1-1

Protocol	Use
HTTP	Web applications
TELNET	Remote log to a computer
FTP	Transfer long files
TFTP	Transfer short messages
SMTP	To send E-mail
SNMP	Remotely manage network devices

Table 1-1 TCP/IP Application Layer Protocols.

HTTP - Hypertext Transfer Protocol.
FTP - File Transfer Protocol
TFTP - Trivial File Transfer Protocol
SMTP - Simple Mail Transfer Protocol
SNMP - Simple Network Management Protocol

1.3.4 Relationship with different layer protocols

Each application should select either TCP or UDP as their Transport layer protocol. This is defined in the application and we cannot change it. All applications should select IP as the Network layer protocol.

The relationship among these three layers is given in Table 1-2.

Application Protocol	Transport Protocol	Network Protocol
HTTP	TCP	IP
TELNET	TCP	IP
FTP	TCP	IP
TFTP	UDP	IP
SMTP	TCP	IP
SNMP	TCP	IP

Table 1-2 Relationship of three layer protocols.

It can be noticed that, although we can use the term TCP/IP, there are UDP/IP applications as well.

1.3.5 Internet Services

By using different application protocols we can provide many services via Internet. The most popular services are web service (HTTP) and e-mail service (SMTP).

Private Network and Public Network.

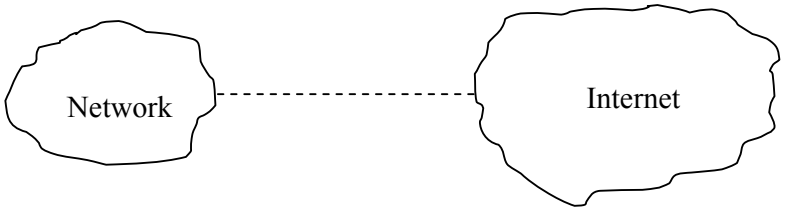


Fig. 1-3 Private and Public Networks

The Internet is a public network. If a network is not directly connected to Internet, it is called a private network. If it is directly connected to Internet, it will become a part of Internet.

In public networks Internet standards have to be followed where as in private networks it is not mandatory. However TCP/IP protocol suit can be used in private networks also.

1.3.6 Intranet and Extranet

A private network maintained by a company or particular organization is called Intranet. For instance Sri Lanka Telecom uses their own Intranet for exchanging internal information, which is restricted to the public.

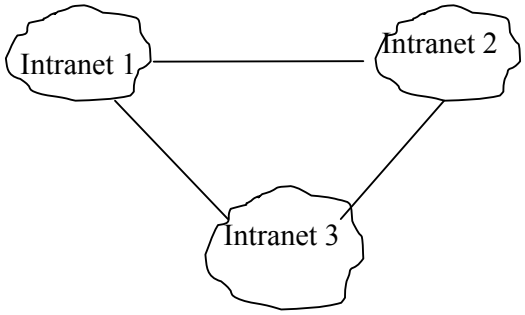


Fig. 1-4 Extranet

An interconnected Intranet is called an Extranet.

2 Overview of TCP/IP

2.1 TCP/IP Operation

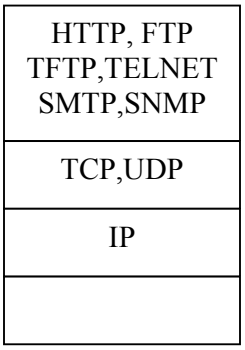
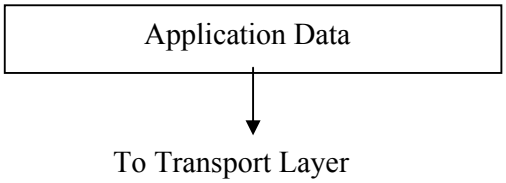


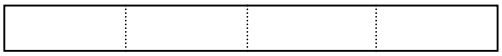
Fig. 2-1 TCP/IP Protocol Suite

The TCP/IP protocol suite is shown in the fig 2-1.
The operation of TCP/IP model is as follows.

Application layer sends the application data to Transport layer.



At the Transport layer the application data is divided into small parts. This process is called “segmentation”.



Each segments is combined with a TCP header or a UDP header. The selection of TCP or UDP depends on the application.

The Transport layer can receive more than one application data at the same time.

Each application is given a port number for its identification. Also each segment is given a sequence number in the case of TCP.

Eg: Application 1 - Port number 80, sequence numbers 1000, 1001, …1258

Application 2 - Port number 21, sequence numbers 1518, 1519, …9887

This information is included in the TCP header or UDP header. The standard size of TCP header is 20 bytes and UDP header is 8 bytes.

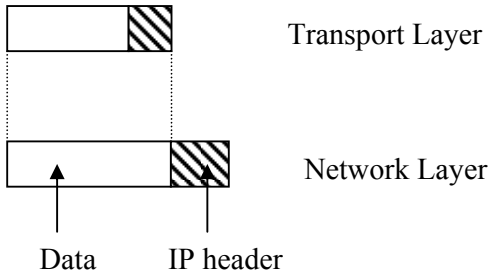


Fig. 2-2 Formation of IP Packet

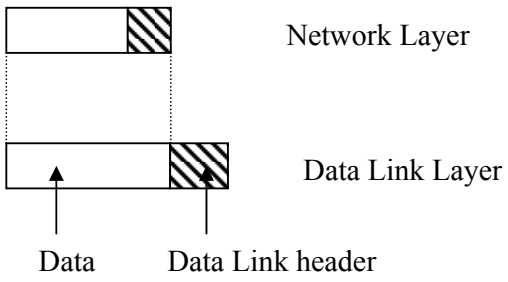
The Transport layer sends the application data and TCP or UDP header to Network layer. The fig 2-2 shows this operation. It can be observed that the application data and TCP or UDP header will consider as data for Network layer. The **standard IP header size is 20 bytes**.

Eg. Segment data -1000 bytes
TCP header -20 bytes
IP data -1000 + 20 = 1020 bytes
IP header -20 bytes
IP packet -1020 + 20 = 1040 bytes

The IP data and IP header together is called an IP Packet. However the maximum total length of the IP Packet is 65535 bytes. Therefore the maximum segment size should be limited to 65535 – 20 = 65515 bytes.

The IP header consists of many information. One of the most important header fields is destination IP address.

The IP packet is sent to the data link layer. The whole IP packet is considered as data for the data link layer frame.



The above explanation is for the transmit process. The reverse process is done for the received data.

Data link layer removes the data link header and sends to the Network layer.

Network layer removes the network header (IP header) and sends to the transport layer.

In TCP operation, the header is removed; the segments are assembled in order and sent to the Application layer. In UDP operation, the header is removed and data is sent to Application layer.

2.2 TCP/IP Operation in LANs

The widely used LAN protocol is Ethernet. That is IEEE 802.3 standard.

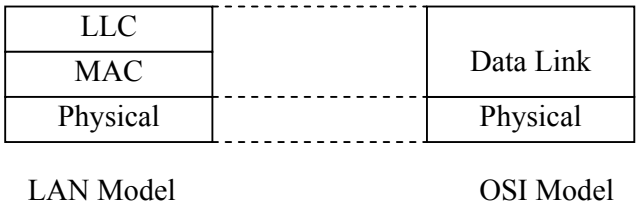


Fig. 2-3 LAN and OSI Comparison

The LAN model has three layers and it is equivalent to Physical and Data link layers of OSI model.

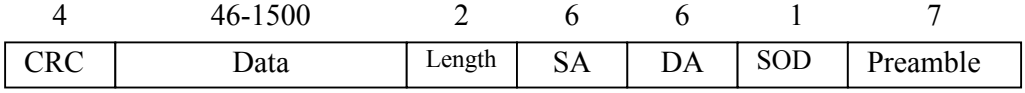


Fig. 2-4 Ethernet Frame Structure

The Ethernet frame structure is shown in fig 2-3. It can carry maximum of 1500 bytes of data.

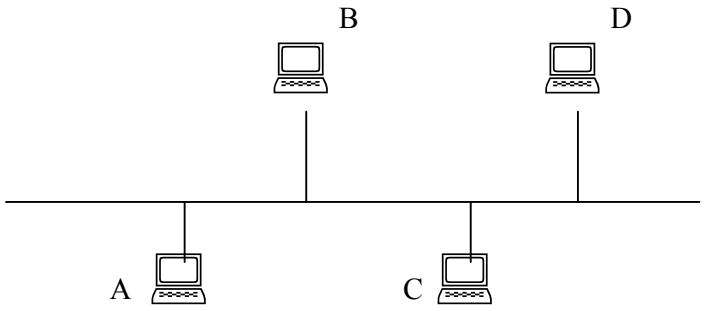


Fig. 2-5 Ethernet LAN

Fig 2-5 shows a typical Ethernet LAN. Suppose Host D is a web server. Host A wants to access the web server. Just the IEEE 802 LAN model is not sufficient for this task. Since the Application (HTTP) is in the Application layer of TCP/IP, the TCP/IP model should be combined with IEEE 802.3 model as shown in fig 2-6.

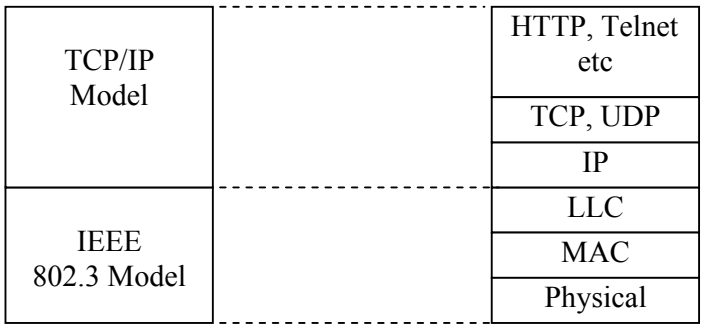


Fig. 2-6 TCP/IP in Ethernet LAN

The data flow is shown in fig 2-7.

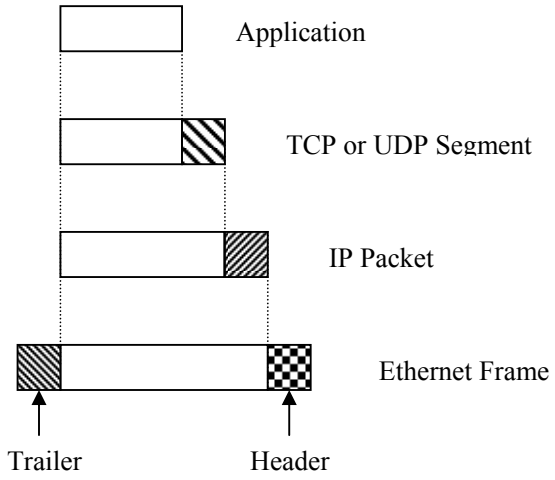


Fig. 2-7 Dataflow from Application to Ethernet

The application layer sends data to the transport layer. It adds a TCP or UDP header and sends to the IP layer. It adds the IP header and sends to the Ethernet frame. (LLC and MAC layers) It adds the header and trailer.

The Ethernet header consists of mainly destination MAC address and source MAC address.

2.2.1 Data Limitation of Ethernet Frame

The maximum size of an IP packet is 65535 bytes. The maximum data size of an Ethernet frame is 1500 bytes.

What happens if the IP packet is more than 1500 bytes?

It cannot be embedded into the Ethernet frame. Therefore a special process called fragmentation is done at the IP layer before it sent to the Ethernet frame.

Eg. Suppose the IP packet size is 2980 bytes.

The IP data part is separated and two new IP packets will be prepared.

Now the size of an IP fragment is 1480 + 20 = 1500 bytes. The new IP packet can be embedded to the Ethernet frame. At the receiving end all fragmented IP packets are defragmented (combined) before sending to the transport layer.

2.3 TCP/IP Operation in WAN

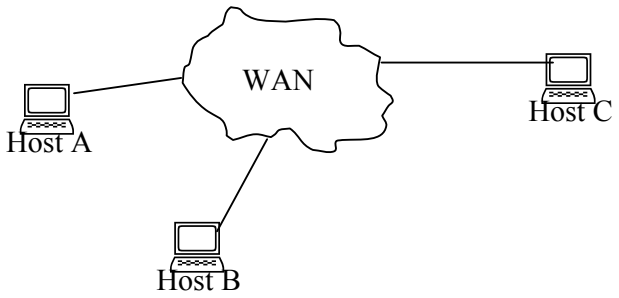


Fig. 2-8 WAN

Host A wants to access to web server (Host C) through the WAN.

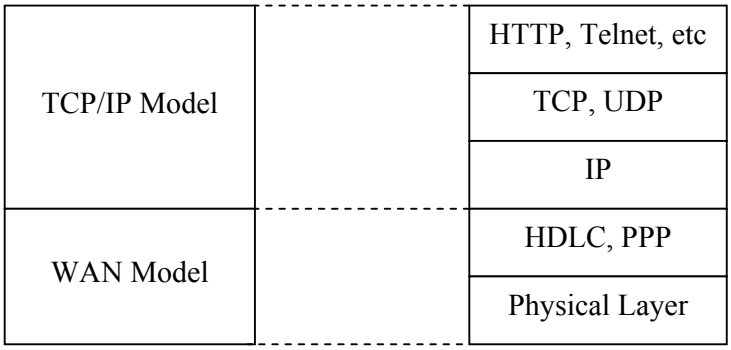
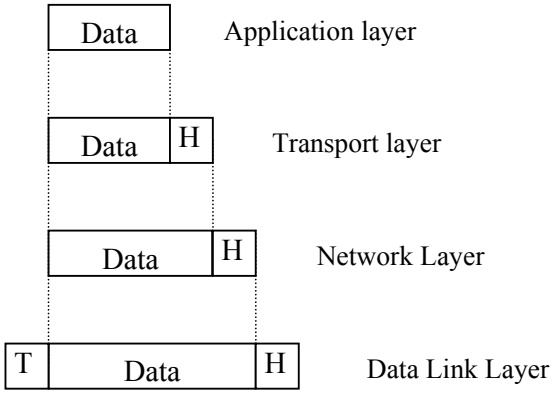


Fig. 2-9 TCP/IP in WAN



The operation is same as in LAN. The difference is that instead of Ethernet frame, the IP packet is sent to a WAN data link layer protocol frame such as HDLC or PPP.

2.4 TCP/IP in LAN and WAN

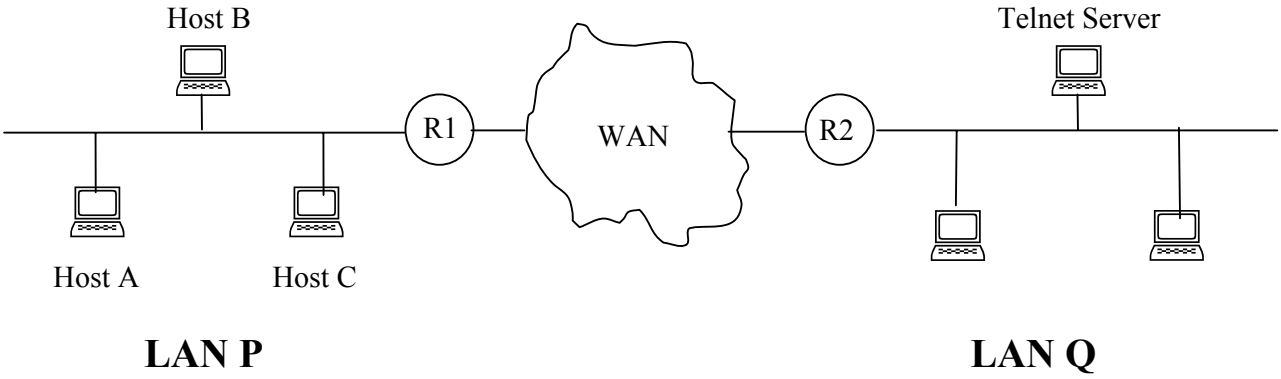


Fig. 2-10 LAN and WAN

Host A in LAN P needs to access the Telnet server in LAN Q

The operation is as follows.

- Host A application data is sent to the transport layer.
- It adds the TCP header and is sent to the IP layer. It adds the IP header and sends to the Ethernet layer.
- The Ethernet frame is sent to the Router (R1).
- Router removes the header and trailer of the Ethernet frame and data is send to the IP layer.
- IP layer modifies the IP header and sends to the data link layer of WAN protocol.

Eg. HDLC

The HDLC frame is sent to the other Router (R2) through WAN.

R2 removes the header and trailer of HDLC frame and sends data to the IP layer. It removes the IP header and sends the IP packet to the Ethernet layer. It adds the header, trailer and broadcasts to LAN Q.

Telnet server receives the Ethernet frame, removes header, trailer and sends data to the IP layer. It removes the IP header and sends data to the transport layer. The transport layer removes the TCP header and sends data to the application layer.

3 Client Server Application

3.1 Overview of Client Server Model

All programs connecting through Internet work as client and server combinations. This means that each program runs in a separate host. The program running in the server is called Server Process and the program running in client host is called Client Process. There can be confusion between server and server process. Normally server means a high-end computer. But server process is a program running in a computer (server) and one computer (server) can run several server processes at the same time. Eg. HTTP server, Telnet server. The client process can access the server process via the network (Internet).

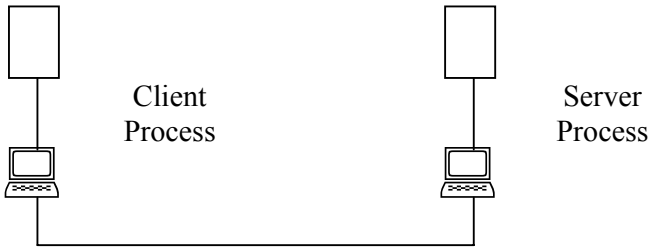


Fig. 3-1 Client – Server connection

There can be client-server, one to one connection. This is called iterative server. Such servers can give only one client connection at a time.

There can be client-server, many to one connection. This is called concurrent server. Such servers can give many client connections at the same time.

3.2 Identification of Processes

Identification of processes is done at the Transport layer. It assigns a special number called "port number" to each process. Each server process is assigned a unique port number. For example HTTP server port number is 80. Telnet server port number is 23.

Therefore, when a client needs to connect to server, it knows the destination port number. The source port number can be assigned arbitrarily by the client.

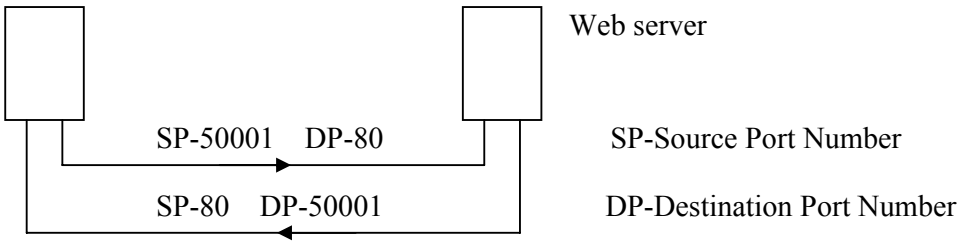


Fig. 3-2 Client and Server ports

Fig. 3-2 shows a web client-server connection. The client should originate the connection. Client knows the port number of the server process (port 80). Client arbitrarily selects a port number for its client process (port 50001). The client sends a segment of data to the server. Now the server knows that from where the request comes. Therefore it can send back segments with destination port 50001.

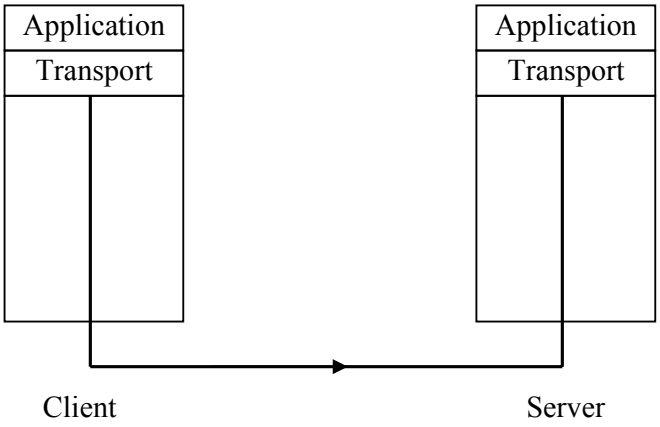


Fig. 3-3 Client Transport – Server Transport connection

The data from client transport layer to server transport layer can be sent in two different ways.

- Connection oriented
- Connectionless

3.2.1 Connection Oriented

In this method, before sending data, a connection is established between the client transport layer and the server transport layer. The connection request is sent by the client to the server. If the server positively acknowledges, a connection can be established. Then data is transferred. After completing sending data, the connection is terminated (same as in data link layer)

In this case, an acknowledgement is received for each data segment. Error control and flow control is also part of this process. Therefore data transfer is reliable. TCP uses this method.

3.2.2 Connectionless

In this method no connection is established prior to sending data. A data segment is released from the client with the destination port number. It will go through the network and reach the transport layer of the server. The server does not send any acknowledgement. Therefore the client does not know whether the data is received by the server or not. Hence this method is unreliable. UDP uses this method.

Normally UDP is used to send small amount of data in a periodic manner. Eg. Routing information. Suppose the period of sending some kind of data is 30 seconds. If server does not receive the first data segment, there is a chance to receive the second segment send after 30 seconds.

3.3 Data sent through a WAN

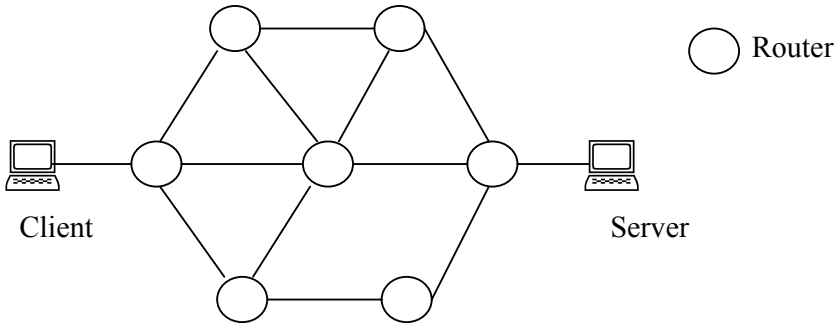


Fig. 3-4 Data through WAN

Normally a WAN is a switched network.

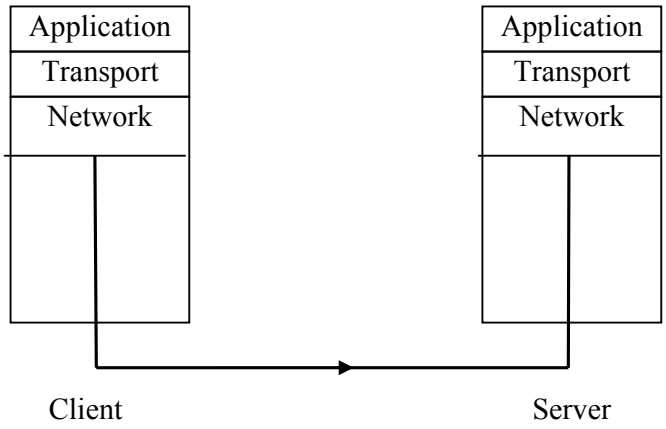


Fig. 3-5 Client Network layer & Server Network layer connection

The application layer sends data to the transport layer. It adds the source and destination port numbers and sends to the network layer. TCP/IP uses Internet Protocol (IP) as the network layer protocol. The network layer adds the corresponding destination IP address and source IP address and sends the data packet to the network as shown in Fig. 3-4. When it comes to the first router it checks the destination IP address, refer the routing table and find out the relevant output port and put the data to that port. Then the packet goes to second router and performs the same. This happens until the data packet reaches the router connected to the server and it delivers the data packet to the server.

It is the responsibility of the network layer to deliver the data packet to the correct destination network layer. It can be noticed that IP protocol is connectionless. No connection is established before sending data. Therefore error control and flow control does not occur at the network layer. Hence IP is an unreliable protocol. Since TCP is reliable, TCP and IP together will become a reliable combination of protocols. However UDP and IP together is not a reliable combination.

Although IP does not give feedback to the source, it uses a separate protocol called Internet Control Message Protocol (ICMP) to send back some error messages to the source.

At the server, network layer removes the IP header and sends the data segment to the transport layer. The transport layer checks the destination port number, removes the TCP or UDP header and directs data to the relevant application.

3.3.1 Data Link Layer Operation of a WAN

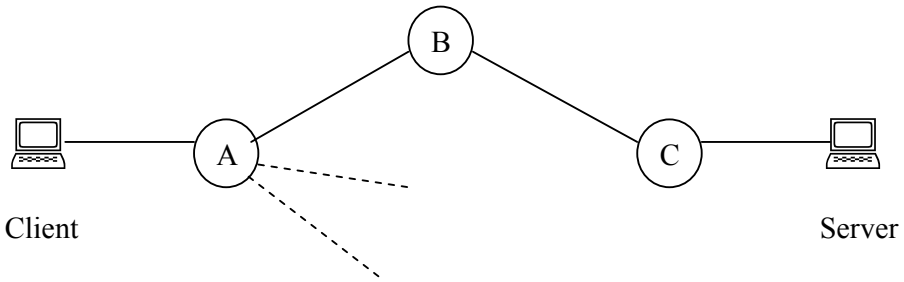


Fig. 3-6 Data encapsulation in WAN

The network layer selects the path ABC to send data. How many data links are involved in this path?

Client → A, A→ B, B→ C, C → Server

There are four data links involved.

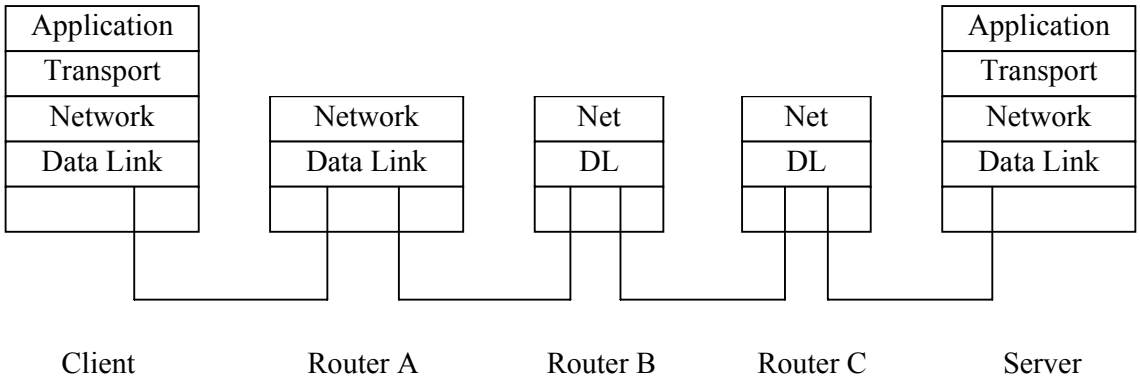


Fig. 3-7 Layers involved in WAN connectivity

At the client, IP packets are sent from network layer to the data link layer. It makes the data frame as per the data link layer protocol. It includes the source physical address and the destination physical address. The source is the client and the destination is the router A. If the client connects to router A through a LAN the physical address is the MAC address and the data frame is Ethernet frame. In this case source physical address is client's MAC address and destination address is router A's LAN port (Ethernet port) MAC address.

The Router A receives the data frame and removes the header and the trailer and adds a new header and a trailer as per the data link layer protocol between router A and router B. For example it can use HDLC protocol. The new source address is router A's WAN port physical address, which connects to router B. The destination physical address is router B's WAN port physical address, which connects to router A. But practically this may not happen.

The Router B receives the data frame and removes the header and the trailer and adds a new header and a trailer as per the data link layer protocol between router B and router C. For example it can be PPP protocol. The new source address is router B's WAN port physical address, which connects to router C. The destination physical address is router C's WAN port physical address, which connects to router B. (May not happen practically)

The router C receives the data frame and removes header and trailer and adds a new header and trailer as per the data link layer protocol between router C and the server. If the router C connects to the server through a LAN the source address is router C's Ethernet port MAC address and destination address is server MAC address.

The server removes the Ethernet frame and header and sends the data packet to network layer. It removes the IP header and sends the data segment to transport layer. It checks the destination port number and directs the data to the application.

3.4 Data sent through a LAN

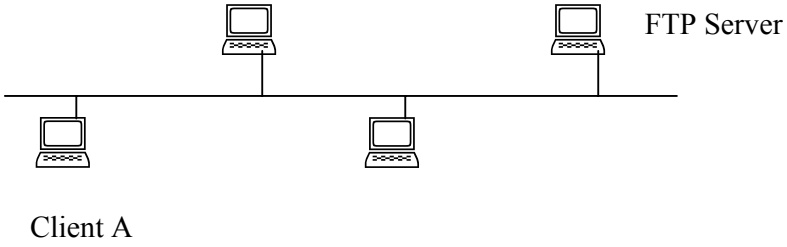


Fig. 3-8 Ethernet LAN

Fig. 3-8 shows an Ethernet LAN. Its network topology is a bus topology. All computers share the same media. If one computer sends data to the media it goes as an electric current. Since all computers are connected to same media this current is received by all the other computers. This means that if one computer transmits a signal to the media, it is received by all the other computers. Therefore this is a broadcasting network.

If client A wants to access the FTP server it sends a data frame to the media. All other computers receive this data frame and they check the destination MAC address in the Ethernet frame. The destination MAC address is FTP server's MAC address. Therefore only the FTP server accepts the data frame and the other computers ignore the data frame.

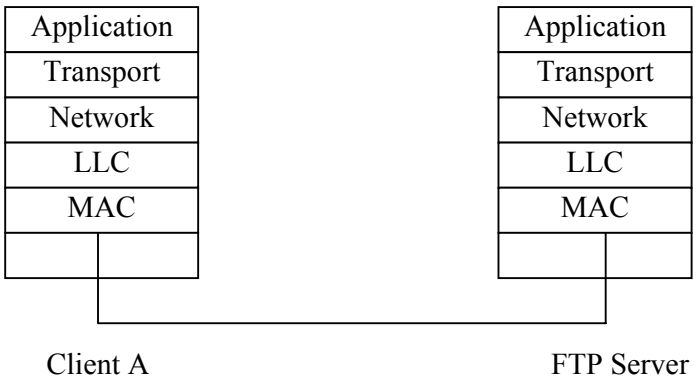


Fig. 3-9 MAC Layer in the data transmission

The operation of data flow is, the client application layer sends data to the transport layer. It adds the source and destination port numbers and sends to the network layer. The network layer adds the corresponding destination IP address and source IP address and sends the data packet to LLC and MAC layer. Those layers add corresponding destination MAC address and source MAC address and sends the

Ethernet frame to the media (bus). The server accepts the Ethernet frame at the MAC layer. At the server the other layers will do the reverse process of client layers and finally the transport layer sends data to the application layer.

3.4.1 How to find out destination MAC address?

In the above example the client MAC layer adds the destination MAC address to the Ethernet frame. The client should know the destination IP address. But the client may not have the information of the destination MAC address. For this purpose a separate protocol called "Address Resolution Protocol" (ARP) is used.

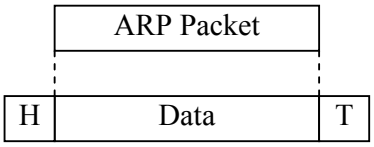


Fig. 3-10 Ethernet Frame with ARP packet.

The ARP packet is encapsulated in an Ethernet frame and broadcasted to find out the MAC address of a particular IP address.

In some applications (eg. DHCP server) we have to do the reverse process of ARP. That is, we know the MAC address and the corresponding IP address has to be found. For this purpose Reverse Address Resolution Protocol (RARP) is used.

3.5 How to send a message to a Group of Computers

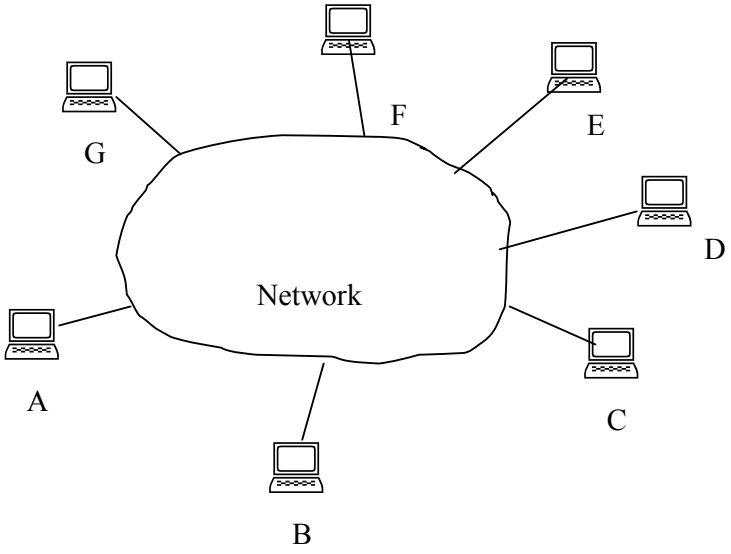


Fig. 3-11 Multicasting

Figure 3-11 shows a computer network. It can be a WAN. Computer A wants to send a message to computers B, E and F only. This is called multicasting. For this purpose a special protocol called "**Internet Group Management Protocol**" (IGMP) is used.

3.6 Routing Protocols

The routers in a WAN individually maintains routing tables. The router reads the destination IP address in the IP packet and the routing decision is taken (to which port the IP packet should be put) as per the information in the routing table. The routing table should have all routing information in the whole network. If a new network is connected to the existing network, all routers should update such routing information. If there are many routers in the network it is very difficult to perform this task manually. Therefore automatic routing information or dynamic routing is used. For this purpose routing protocols are used. The well-known routing protocols are RIP, OSPF and BGP.

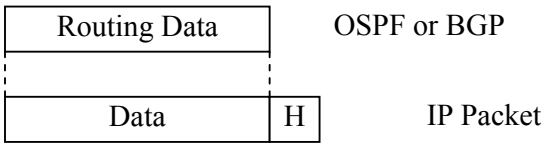


Fig. 3-12 Encapsulation of routing protocols.

The routing protocol data is encapsulated in the data part of the IP packet. This will become a special IP packet.

3.7 Application Layer Protocols

There are many application layer protocols. Some of them are briefly explained below.

3.7.1 Dynamic Host Configuration Protocol (DHCP)

IP address of a Host can be manually assigned. In large networks there is a possibility of allocating the same IP address to two computers. Then an IP conflict can occur which will prevent both computers or one of the computers from accessing the network. In order to avoid such a situation the IP address can be dynamically (automatically) allocated by using a special server process, which uses the DHCP protocol.

3.7.2 Domain Name System (DNS)

It is easy to remember a domain name such as www.slt.lk rather than remember the corresponding IP address. However TCP/IP needs the destination IP address to fill the destination IP address field of the IP packet. Therefore the domain name and corresponding IP address should be maintained in a separate server and this program is called the DNS server process and the corresponding protocol is called DNS protocol.

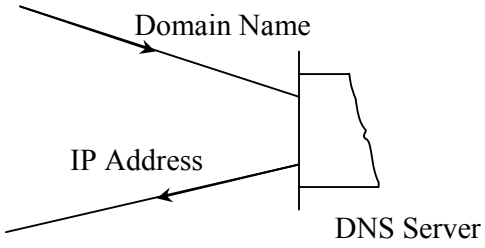


Fig. 3-13 DNS Operation

3.7.3 TELNET

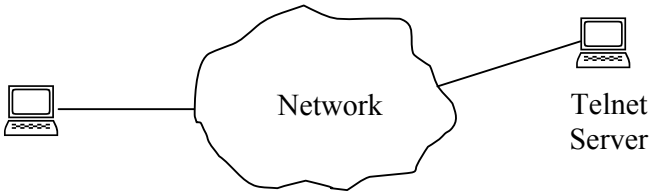


Fig. 3-14 TELNET Operation

In some occasions we may need to access a server located in a remote location in the network and to perform some changes in the database, application etc. In this case the Telnet protocol can be used. The remote host should run the Telnet server process. The computer which needs to access the remote database should run Telnet client process.

3.7.4 File Transfer Protocol (FTP)

We can transfer a file as an attachment of an e-mail. But this is difficult for a big file such as 10 MB. Since many mail servers restrict the attachment size. For this type of application FTP is an appropriate protocol.

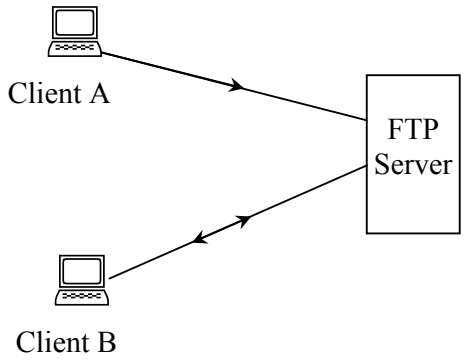


Fig. 3-15 FTP Operation

Client A needs to send a file to client B. Client A sends the file to an FTP server. Client B access the FTP server and obtains the file. FTP uses the TCP as the transport layer protocol.

3.7.5 Trivial File Transfer Protocol (TFTP)

TFTP is same as FTP. Normally it is used to send small files. Since TFTP uses UDP as the transport layer protocol it is an unreliable protocol.

3.7.6 Simple Network Management Protocol (SNMP)

This protocol is used to remotely manage network devices. The network device should run the SNMP server process. Then we can run SNMP client process in the network management server or any other computer and communicate with remote network device. This is mainly used to remotely configure the network devices and remotely diagnose the faults.

3.7.7 Simple Mail Transfer Protocol (SMTP)

SMTP is used to send e-mails.

3.7.8 Hyper Text Transfer Protocol (HTTP)

HTTP is used for web applications.

4 Transmission Control Protocol (TCP)

4.1 Introduction

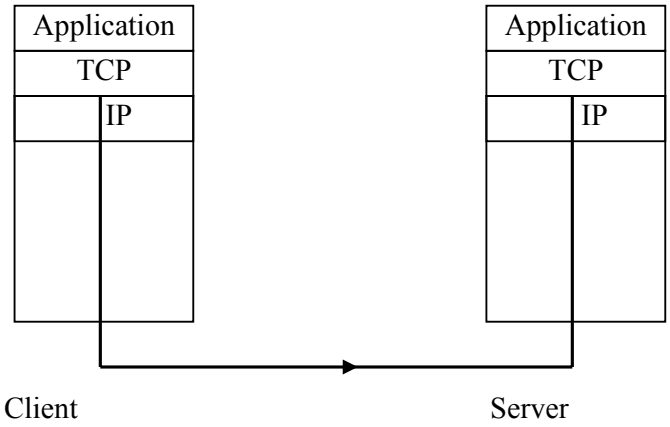


Fig 4-1 TCP Client Server Connection

TCP is one of the transport layer protocols of TCP/IP. TCP should communicate with the Application protocol and IP protocol of the same host. Also it should communicate with the TCP layer of the remote host.

4.2 TCP connection process

Let us first consider the TCP-Client and TCP-Server connection process. It has following three phases.

- Establish a connection
- Transfer Data
- Terminate the connection

4.2.1 Establish a connection

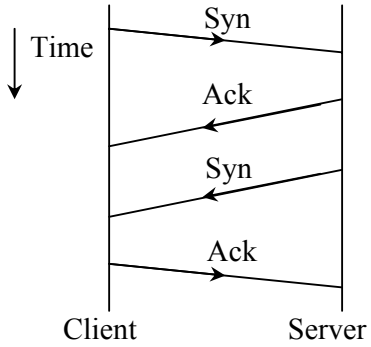
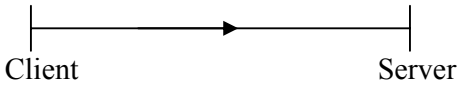
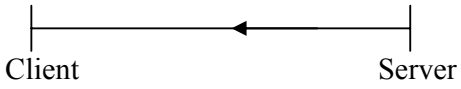


Fig 4-2 Client Server Connection Establishment

Connection establishment should be originated from the client. It sends a connection request message called SYN (Synchronization) to the Server. The Server accepts the request and sends an Acknowledgment (ACK) to the Client.



Now the client to server direction connection has been established. At the same time, server sends a SYN to the client to establish a connection from server to the client direction. The Client accepts the request and sends an ACK to the Server.



Now server to client direction connection is also established.

The above four steps can be reduced to three steps and it is called a three way handshake.

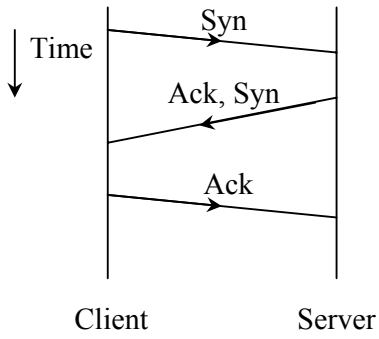


Fig 4.3 Three way handshake

The SYN, SYN/ACK and ACK of three-way handshake operations are 1-byte messages.

Once the connection is established full duplex data transfer can be done.

4.2.2 Data transfer

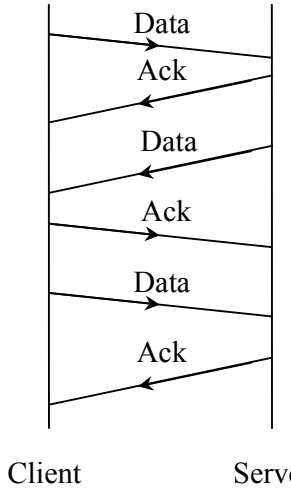


Fig 4.4 Client-Server Data Transfer Process

TCP is a reliable protocol. It sends an acknowledgement (ACK) for each segment of data sent from client to server or from server to client. This process is also further simplified as follows.

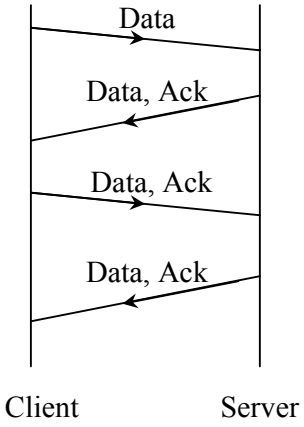


Fig 4.5 Piggybacking.

It is not necessary to send data and ACK separately. Both can be sent together. This is called **piggybacking**.

How to identify a data segment and the corresponding acknowledgement?

Data is transferred as segments. Each segment is given an identification called a sequence number.

Acknowledgement Number = Sequence Number (Received) + Number of bytes in the segment

4.2.3 Terminating the connection

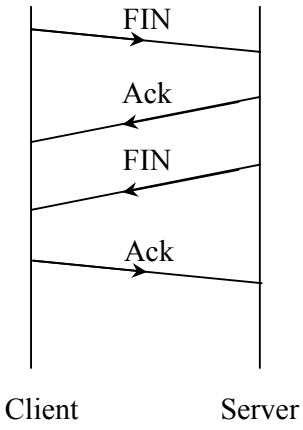


Fig 4.6 Four-way handshaking

TCP performs full duplex data transfer. It has the feature of terminating a connection for one direction at a time. This is done by FIN and ACK messages.

If client to server direction connection is needed to be terminated, client sends a FIN and server the sends an ACK. Now the client cannot send data to the server but the server can send data to the client. After the server has sent all data, it sends a FIN to the client and the client sends an ACK. Now the server to client direction connection is also terminated. This whole process is called four-way handshaking.

Normally the connection termination request is initiated by the client. But it is not mandatory and while keeping the client to server connection, server to client connection can be terminated. FIN and ACK are considered one-byte messages.

4.3 Problems related to data transfer

Three problems should be addressed in the data transfer process. They are

- Error control
- Flow control
- Congestion control

4.3.1 Error control

TCP uses checksum bits for error detection. The receive end checks the error in a segment. If there are no errors it sends an acknowledgement to the sender. If

errors are found, the receiver does not send any negative acknowledgement to the sender. Instead of that the receiver keeps silent.

The sender knows the time taken to go to the other end and come back. This is called the **Round Trip Time (RTT)**. The Retransmission Time is a factor of RTT. If Retransmission Time expires, the sender retransmits the same segment.

In this process, the sender and receiver should have the following facilities.

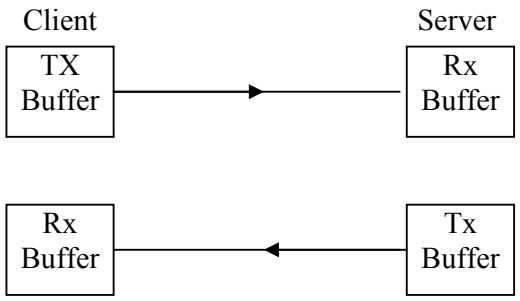
- Sender buffer
- Receiver buffer
- Timer

Sender buffer needs to keep the segments, until it receives an acknowledgement from the receiver.

Receiver buffer needs to keep the segment, until the error checking is over.

The sender needs a timer to check whether the Retransmission Time is expired after sending the segment.

4.3.2 Flow control



`

Fig 4.7 Transmit and Receive buffer

The buffer has limited memory space. The buffer keeps all transmitted segments until an acknowledgement is received. Tx buffer discards the segment after receiving the acknowledgement. If there is a delay in receiving an acknowledgement, the Tx buffer can get full. Then the data segment transmission should be temporarily stopped.

At the Rx buffer the received segments are temporally stored until errors are checked. If there are no errors the data segment is sent to the application layer. If errors are found the data segment is discarded. If the computer is slow or the application process is slow it takes time for error checking. Then Rx buffer can get full. If any new segment is received, the Rx buffer is overflowed and data is lost. Therefore the size of the remaining space of Rx buffer should be informed to the

sender. This is called the “window size”. If the buffer is full it is informed to sender that the window size is zero. Then the sender will not send any data until the window size become non-zero. This whole process is called flow control.

4.3.3 Congestion control

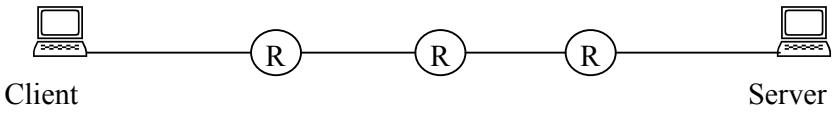


Fig 4.8 Client-Server path

If data goes through a WAN it may pass through many routers. If any of the routers have to handle high traffic its memory may not have sufficient space to accept all IP packets. Then it can discard some IP packets. This is called congestion of the network. There can be other reasons for congestions. In order to avoid discarding of IP packets, small IP packets should be sent. IP packet size depends on the TCP segment size. Therefore the **Maximum Segment Size (MSS)** of the TCP of sender should be decided by taking congestion of the network into account. Then the IP packets will not be discarded. This process is called congestion control.

4.4 Communication between TCP layer and Application layer.

So far we discussed the communication between client TCP layer and server TCP layer. Let us discuss the communication between application layer and TCP layer of the same host.

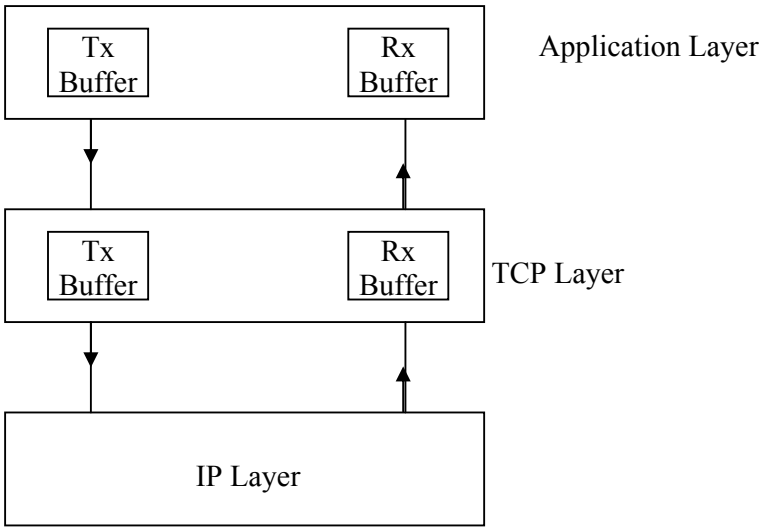


Fig 4.9 TCP layer and Application layer communication

Fig 4-9 shows the communication among application layer, TCP layer and IP layer of the same host.

The application layer can maintain a Tx buffer and Rx buffer to communicate with the TCP layer. But the application layer buffers are not essential and it depends on the application protocol. Also the application can directly communicate with the TCP buffer without having application layer buffers.

4.4.1 Slow Applications

An Application writes Tx data to the Tx buffer of the TCP layer. The TCP protocol adds a 20byte TCP header (standard size of TCP header) and sends to IP layer. The IP layer adds 20 bytes of IP header and sends the IP packet to the data link layer. It also adds a header and a trailer (say 26 bytes). You can notice that altogether 20+20+26=66-byte overhead is added to application data.

Suppose the application is very slow and it sends 1 byte at a time. If 1 byte of data is sent immediately it will be combined with 66 byte overhead bits. Therefore the efficiency of the data transfer will become 1/66 times. This is a very inefficient data transfer.

TCP has the facility to improve such a situation by defining the minimum data required for a segment. TCP waits until such an amount of data is collected. After that the TCP segment is sent to the IP layer.

4.4.2 Normal application with large amount of data

Application writes Tx data to Tx buffer of TCP layer. It segments data to small pieces called maximum segment size (MSS). Those segments are added with TCP headers and sent to the IP layer.

At the receiver TCP layer those segments are combined and makes up the application data and sends to the application layer.

4.4.3 Fast application/ Slow network/ Slow receiver

Application layer buffers and TCP layer buffer sizes are decided on the following factors.

- Speed of application
- Speed of computer
- Speed of the network
- Congestion of network

If the sender application is faster than the network it will overflow the TCP buffer and gives a " runtime error"

If the window size of Tx TCP is zero the application layer should stop writing to the Tx TCP buffer.

4.4.4 Multiplexing

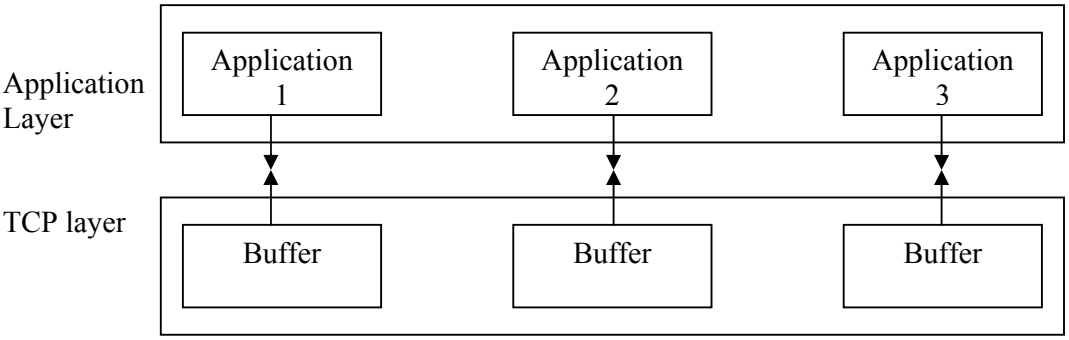


Fig .4 –10 Multiplexing

TCP layer can handle several application processes at the same time. This feature is called Multiplexing. Port Number identifies the application.

4.4.5 TCP with IP layer

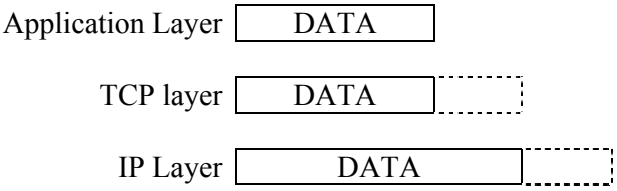


Fig 4-11 IP packet

The IP layer receives data from the transport layer or directly from IP layer (eg:ICMP). Therefore the type of data in IP packet can be TCP, UDP, ICMP, etc. The IP packets identify the type of data by the "protocol number" which is one of the fields of the IP header. Each type of data is given a unique protocol number.

4.5 Port number

The port number is a 16 bit binary number in the TCP. Therefore the port number is in the range of 0-65535. The port numbers are divided into three ranges.

- Well known ports
- Registered ports
- Dynamic ports/ Ephemeral ports

4.5.1 Well-known ports

The port numbers ranging from 0-1023 is called well-known ports. They are assigned to standard server processes such as FTP, Telnet. The numbers are assigned by IANA. The well-known port numbers used with TCP is given in table 4-1

Port	Protocol	Description
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active user
13	Daytime	Returns the data and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20	FTP, Data	File Transfer Protocol (data connection)
21	FTP, Control	File Transfer Protocol (control connection)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer protocol
111	RPC	Remote Procedure Call

Table 4-1 Well known Ports

4.5.2 Registered ports.

The ports ranging from 1024 - 49,151 are to be registered with IANA to prevent duplicating. They can be used for proprietary server processors or any client process.

4.5.3 Dynamic ports

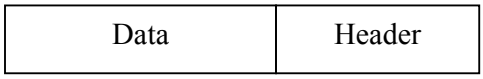
The ports numbers from 49,152 to 65,535 are dynamic or ephemeral ports. It can be frequently used. Normally they are used by client processes temporarily. The client process need not have a fixed port number. For example a client can access the server with client port number 50,000. After terminating that connection, if the client needs to make a connection with server again, the second time client can arbitrarily select port number 65004.

4.6 TCP header field

The TCP header should include all fields required to communicate between client TCP layer and server TCP layer.

- For connection establishment SYN massage should be sent. There should be a field to identify SYN massage
- For acknowledgement ACK field is needed.
- For connection termination FIN field is needed.
- To identify server process, server port number or “destination port number” is needed.
- To identify client process client port number or “source port number” is needed.
- To identify each segment a sequence number is needed.
- To acknowledge to each segment “acknowledgement number” is needed.
- For error detection “checksum field” is needed.
- To inform the window size to the other end “window size” field is needed.
- To identify the length of the segment “length” field is needed.

4.7 TCP segment



TCP segment consists of data and header. The standard header length is 20 bytes. With the option fields it can be expanded up to 60 bytes.

<table>
<tr><td colspan="8">Source Port Number (16 Bits)</td><td colspan="2">Destination Port Number (16 Bits)</td></tr>
<tr><td colspan="10">Sequence Number (32 Bits)</td></tr>
<tr><td colspan="10">Acknowledgement Number (32 Bits)</td></tr>
<tr><td>Header Length (4 Bits)</td><td colspan="2">Reserved Bits (6)</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td>Window Size (16 Bits)</td></tr>
<tr><td colspan="8">Checksum (16 Bits)</td><td colspan="2">Urgent Pointer (16 Bits)</td></tr>
<tr><td colspan="10">Options & Paddings</td></tr>
<tr><td colspan="10">Data</td></tr>
</table>

Fig 4.11 – TCP Header

The TCP header with its fields is shown in fig 4.11. The descriptions of these fields are as follows.

4.7.1 Source port address

This is a 16-bit address. Therefore port number can be ranging from 0 – 65535. For client TCP header this is a dynamic port number. For server TCP header this is a well-known port number.

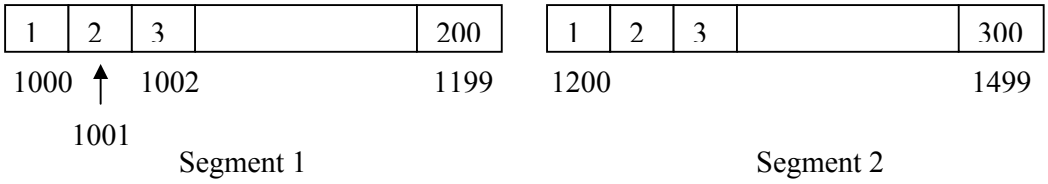
4.7.2 Destination port address

This is the destination process port number. This is also a 16-bit address.

4.7.3 Sequence number

This is a 32-bit field. Any segment will have its corresponding sequence numbers. At the time of establishing TCP connection the first sequence number should be decided. This is called **Initial Sequence Number (ISN)**. It can be any arbitrarily number between 0 and 2^{32}-1.

The allocation of sequence number uses the following process.
Suppose the first data segment has 200 bytes second data segment has 300 bytes
Suppose the sequence number of first segment is 1000. If we give a number to each byte;



The number of first byte of segment 2 is 1200. Therefore the sequence number of second segment is 1200. Similarly the sequence number of third segment is 1500.

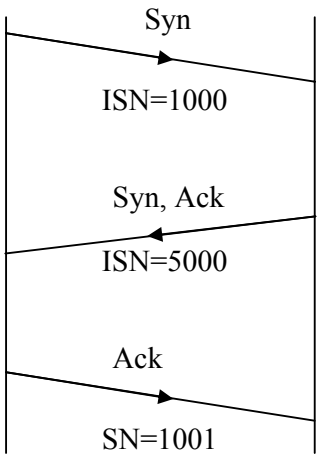


Fig 4-13 Sequence number of three-way handshake

The sequence numbers of connection established is shown in Fig. 4-13. Client selects the Initial Sequence Number (ISN) as 1000; server selects the ISN as 5000. The SYN message is considered as 1 byte. Therefore sequence number of ACK of client is 1001. ACK message is also considered as 1 byte. Next step is to send data from client to server. The first data segment of client will start from the sequence number 1002. Similarly the first data segment of server will start with SN 5001.

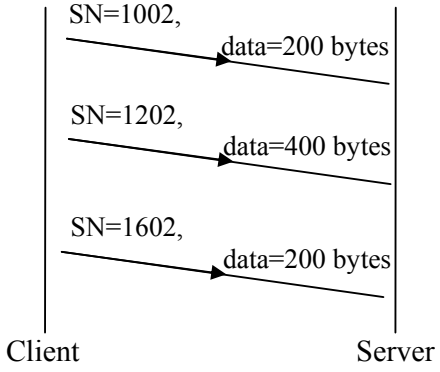


Fig 4 –14 sequence number of data segments

Figure 4-14 shows the continuation of Fig. 4-13 client-server data transfer. The SN of first 200 bytes data segment is 1002.The SN of 400 bytes data segment is 1202. The SN of next 200 bytes data is 1602. The next SN will be 1802.

Suppose server sends 210 bytes, 390 bytes and 400 bytes of data respectively to client. The corresponding sequence numbers will be 5001, 5211 and 5601. The next SN will be 6001.

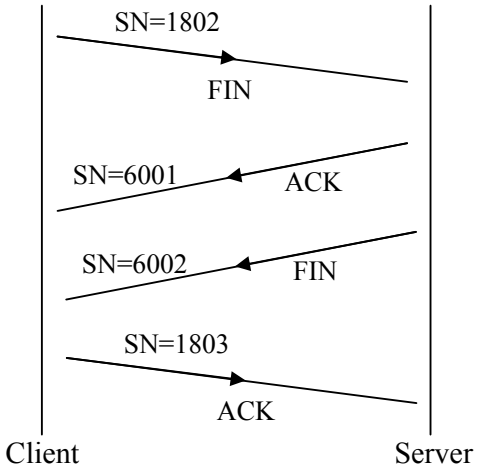
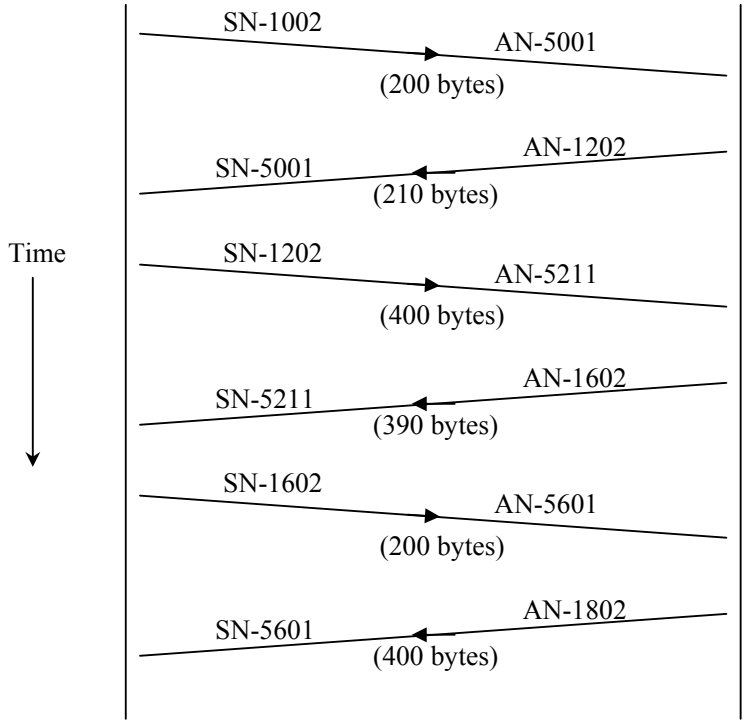


Fig 4 –15 sequence number of connection termination



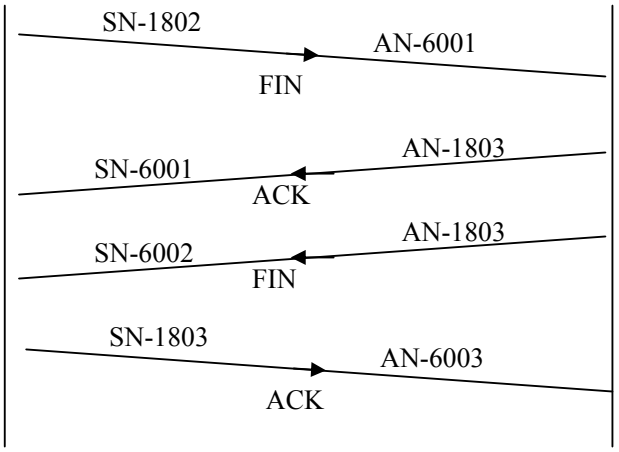


Fig 4 –16 Data transfer and connection termination

The Fig. 4-16 shows the connection termination (four way handshaking) of above client-server. The FIN message is also considered as 1 byte message. Now you can understand from the figure how the sequence numbers are allocated for four-way handshake message.

4.7.4 Acknowledgement Number

This is a 32-bit number. For each segment of data there is a sequence number. An acknowledgement number is sent by the other TCP layer for each segment. The acknowledgement is the next sequence number expected by the receiver from the sender. The following table shows the relationship among sequence number, number of bytes in a segment and the acknowledgement number sent by the receiver.

Sequence Number of segment	No. of bytes in the segment	Acknowledgement number send by receiver
1000 (SYN)	1	1001
1001 (ACK)	1	1002
1002	200	1202
1202	400	1602
1602	200	1802
1802 (FIN)	1	1803

Table 4-2 Sequence and Acknowledge Numbers

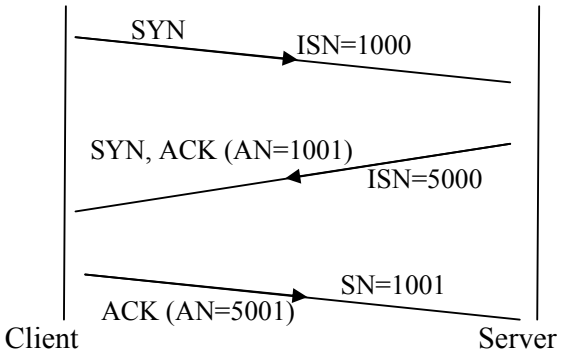


Fig 4 –17a - Connection establishment

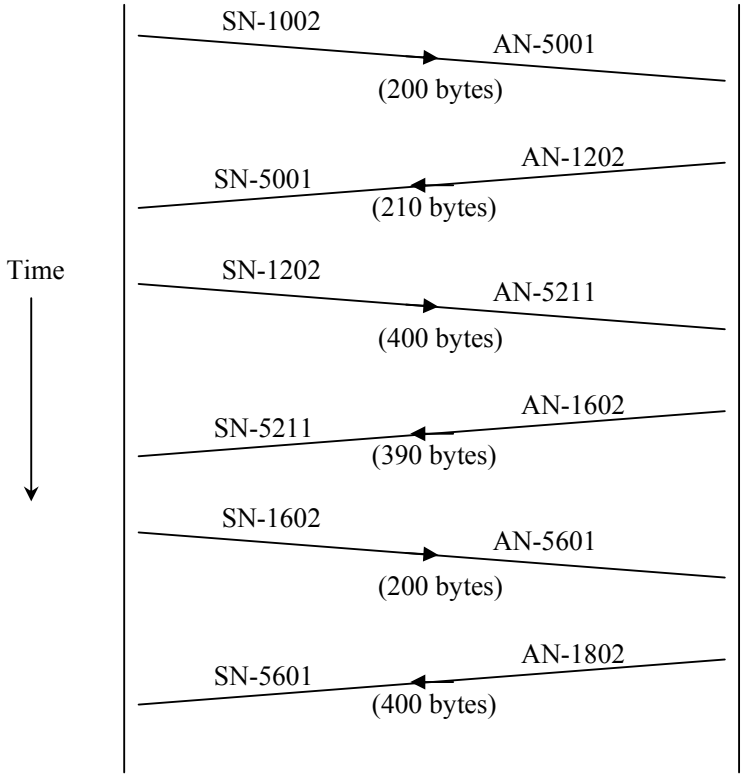


Fig 4-17 b

Fig. 4-17a and Fig. 4-17b explain the sequence number and acknowledgement number of each segment data during connection establishment, data transfer and connection termination.

The example is similar to stop and wait data transfer. But the actual case is different.

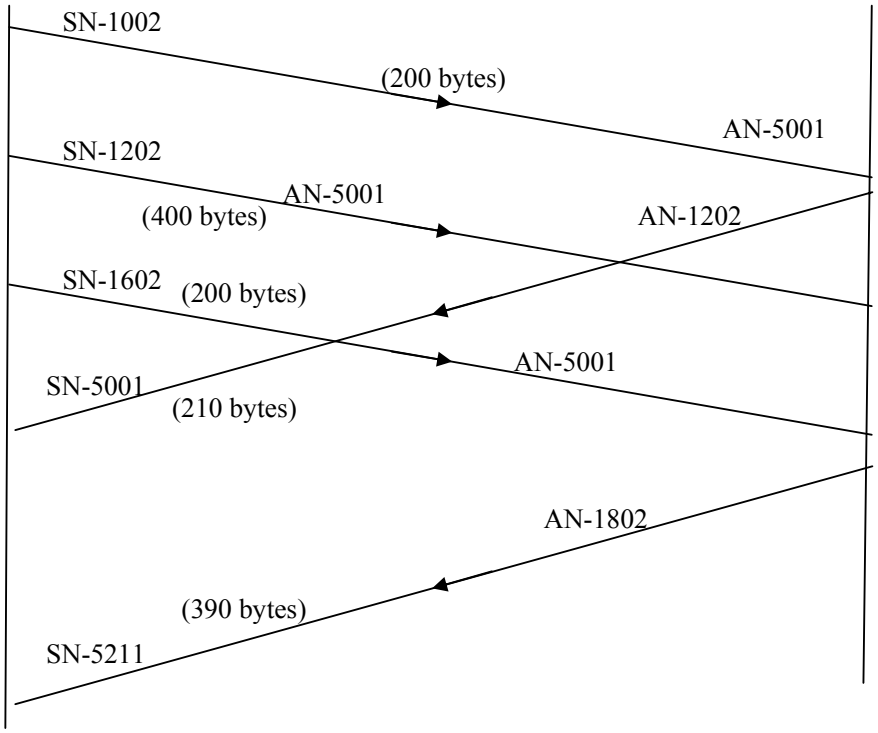


Fig 4 –18

Fig. 4-18 shows an actual type of data transfer. Client or server does not wait until an acknowledgement is received for each segment. It continuously transmits data. Always the following rule has to be followed.

N - the first byte number of segment
AN - the next sequence number to be received.

Client				Server			
Segment	No. of byte(s)	SN	AN	Segment	No. of byte(s)	SN	AN
SYN	1	5000	-				
				SYN, ACK	1	2000	5001
ACK	1	5001	2001				
Data	100	5002	2001				
				Data	1500	2001	5102
Data	1000	5102	3501				
Data	2000	6102	3501				
				Data	3000	3501	8102
Data	500	8102	6501				
				Data	1000	6501	8602
FIN	1	8602	7501				
				ACK	1	7501	8603
				FIN	1	7502	8603
ACK	1	8603	7503				

Table 4-3 SN and AN in Segment transfer

Another example is given in Table 4-4. Note how the SN and AN changes at each segment transfer.

4.7.5 Header Length (HLEN)

This is a 4-bit field. It indicates the number of four byte of data in the header.

Eg. HLEN = 0101 (binary)
= 5 (decimal)
Header length = 5 x 4 bytes
= 20 bytes

HLEN = 1000 (binary)
= 8 (decimal)
Header length = 8 x 4 bytes
= 32 bytes

The **standard size of header is 20 bytes**. The maximum size of header is 60 bytes and it consists of 20 standard bytes and 40 optional bytes.

4.7.6 Reserved

This is a six bit field reserved for future use.

4.7.7 Control

This is also a six bit field. It defines six different flags or control bits. The six flags are shown in Table 4-4.

Flag	Name	Description
URG	Urgent	The value of urgent pointer is valid. Normally receiver reads segments according to ascending order. If this flag is set, that segment should be read immediately.
ACK	Acknowledgement	The value of acknowledgement field is valid. If data position is not available this is an ACK segment. If data is available in the segment, if the acknowledgement number is 'n' it indicates that data bytes upto byte n-1 is OK and waiting for sequence number n.
PSH	Push	The sender can set this flag and ask receiver to send this data to application immediately. If push flag is set it does not care about the window size recommended by other party.
RST	Reset	The connection must be reset. The meaning of reset is completely different. It says to destroy the connection. This can happen due to three reasons • The client requests a connection for server to an unidentified port eg.80, but still web server is not activated. In such situation server send RST to client to destroy the connection. • The connection has been established. After some time the client or server has a problem and request the other party to destroy the connection. Then it (client or server) sends RST to other party. • The other side TCP is idle for a long time. Then RST is sent to the other party to destroy the connection.

SYN	Synchronize	Synchronize to sequence numbers (Initial Sequence Number) suggested at the time of connection establishment.
FIN	Finish	Request to terminate the connection to that direction eg: If client sent FIN (that is set FIN flag) to server, it means that client request from server to disconnect the connection of client server direction.

Table 4-4 Flags

4.7.8 Window size

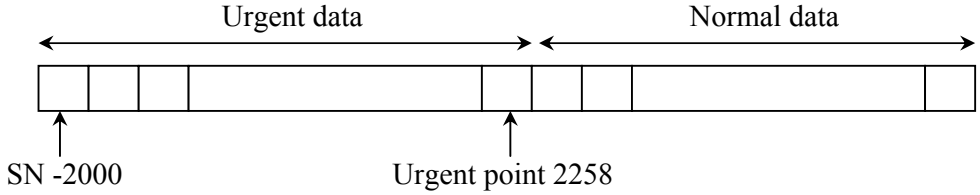
This is a 16-bit field. This number says how many bytes (segment data) can be maintained in the other party. We will discuss more about this under sliding window mechanism.

4.7.9 Checksum

This is a 16 bit field. It contains the checksum bits of header and data which is used to check whether there are any errors.

4.7.10 Urgent Pointer

This is a 16-bit field valid only if the urgent flag is set. This value gives the byte value at the end of urgent data.
Eg: data segment



The bytes 2000 to 2258 (ie, first 259 bytes of the segment) are urgent data. The rest of the data in the segment are normal data.

The URG flag is set, those 259 bytes of data is sent immediately to application for immediate processing.

4.8 TCP Timers

TCP uses different timers for different purposes. Table 4-5 shows the summary of TCP timers.

Timer	Purpose
Retransmission	For error control
Persistence	To avoid problems of zero window size advertisement
Keep alive	To check whether client is alive if it is idle for a long time
Time- waited	To avoid problems with delayed FIN segments

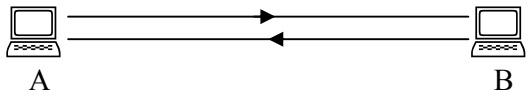
Table 4-5. TCP Timers

4.8.1 Retransmission Timer

This timer starts after sending a segment. Each segment has its own timer. If the acknowledgement is not received before this timer expires, the same segment is re-transmitted and the timer is reset.

If the acknowledgement is received before retransmission timer expires that timer will be destroyed. (Timer will be stopped)

Retransmission time =2 x RTT



Round Trip Time (RTT)

A send a segment to B. B process the segment and replay to A. The time involved here are

- Propagation delay from A to B
- Propagation delay from B to A
- Processing time at B

The total of above times are called RTT. Since IP packets can go through different paths RTT is not a fixed value for all the segments. The RTT set for a segment is

RTT = α x Previous RTT+(1-α) Current RTT

Normally α is 90%.

The RTT can be measured in two different methods.

Using the "time stamp" obtained from the option field of TCP header.
TCP sends a segment, starts a timer, and waits for acknowledgement. It measures the time between sending of the segment and receiving of the acknowledgement.

There is a difference between measured RTT and setting RTT. An example is shown in table 4-6.

Segment	Measured RTT (ms)	Set RTT (ms)
N	250	
n+1	200	
n+2	180	90%250+10%200
n+3	220	90%200+10%180

Table 4-6 Measured RTT and Set RTT

When we start to send n+2, we had the RTT measured from segment n+1 (current RTT) and RTT measured from segment n (previous RTT).

After receiving the acknowledgement of segment n+2, RTT is measured. This will be the current RTT for segment n+3. Note that we assume here that an acknowledgement is received for each segment. You can understand later that one acknowledgement can be received for several segments.

4.8.2 Persistence Timer

We will discuss the flow control window mechanism later. However it is same as the window mechanism of Data Link Layer.

If the receiving TCP announces a window size of zero, the sender should stop transmission until the receiving TCP sends an acknowledgement announcing a non-zero window. If this acknowledgement is lost on the way the sending TCP never sends a segment and the system comes to a deadlock. In order to avoid such a situation the persistence timer is used.

When the sending TCP receives an acknowledgement with zero window size it starts the persistence timer. If it does not receive an acknowledgement with non-zero window size before persistence timer expires it sends a special segment called probe. This segment contains only one byte of data. It has a sequence number but the receiver is not acknowledged for it and also the sequence number is ignored in calculating the sequence number for the rest of the data. The receiving TCP

immediately sends the previous segment, which was an acknowledgement with non-zero window size.

The value of persistence timer is set to the value of retransmission time. However, if a response is not received from the receiver, another probe segment is sent and the value of persistence timer is doubled and reset. The sender continues sending the probe segments and doubling and resetting the value of persistence timer, until the value reaches a threshold (usually 60 seconds). After that the sender sends one probe segment every 60 seconds until the window is responded.

4.8.3 Keep alive Timer

In TCP, once the connection is established, until four-way handshake termination process is activated, the connection remains. For each client connection the server allocate some amount of resources. If the client crashes it does not have a way to inform the server. Therefore the connection remains forever. In order to avoid such a situation, the keep alive timer is used.

Each time the server hears from the client, it resets this timer. The time out is usually two hours. If the server does not hear from the client after two hours, it sends a probe segment. If there is no response after 10 probes, each of which is 75 seconds apart, it ensures that the client is down and terminates the connection.

4.8.4 Time waited Timer

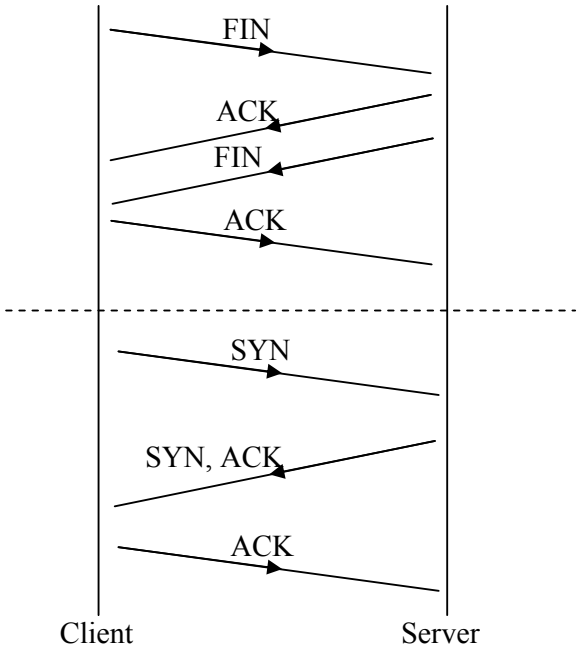


Fig 4-19 Connection re-establishment

There may be instance that a TCP connection can be established just after connection termination.

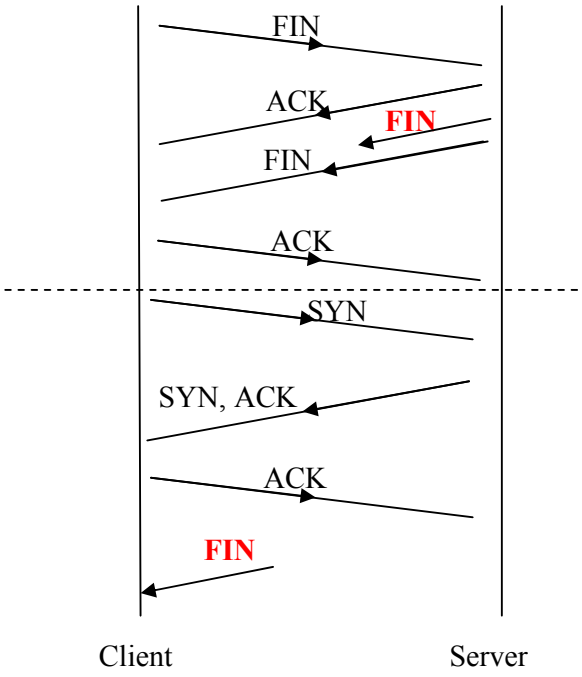


Fig 4-20 Server-Client direction FIN is lost

Suppose in the termination process, server sends a FIN and it was not received by the client. Server sends FIN again. Only after the establishment of the connection, the client receives the FIN previously sent by the server. Then the client thinks that the server needs to terminate server client direction connection. This will upset the whole process. In order to avoid such a situation the time-waited timer is used. The time-waited time is set to two times the expected lifetime of a segment.
Normally this value is set to 2 minutes.

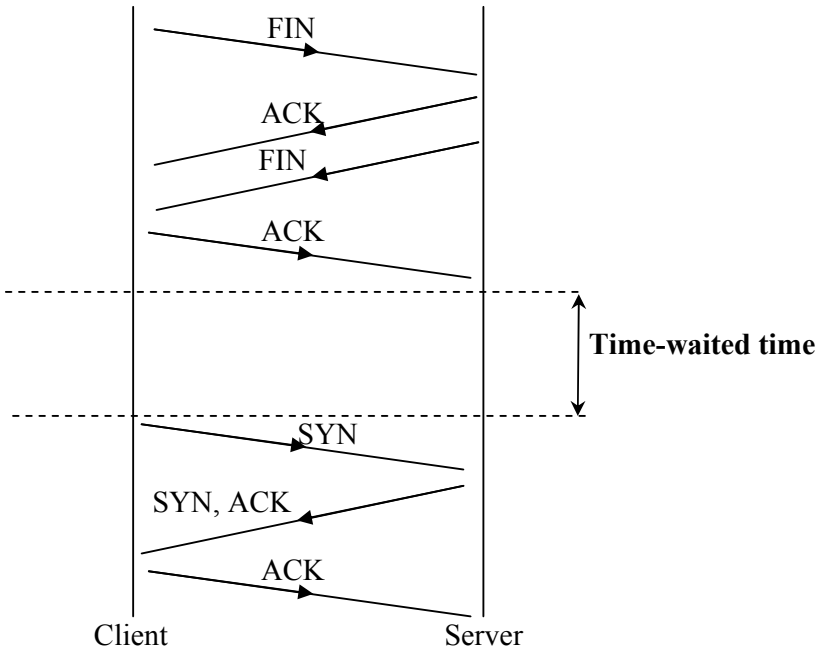


Fig 4 - 21

After a termination of a connection it should wait "Time-waited time", before making the next connection. During this period whatever segments coming to client or server is discarded. Therefore the problem explained before will not exist.

However, we have the option to remove the time-waited time for special applications. This should be done by socket programming.

4.9 Error Control

TCP uses the backward error control. In data link layer backward error control uses positive acknowledgement (ACK) and negative acknowledgement (NACK), and for error detection CRC is used. But there is some difference in TCP compared to data link layer operation.

For error detection, the checksum bits in the TCP header is used by the receiver. If the receiver detects errors, it will discard that segment only. The receiver will not send any feedback to sender.

Error control process considers,

- Errors in the received segment (corrupted segments)
- Segment is lost on the way before reaching the receiver (last segment)
- Duplicate received segments
- Out of order segments (the segment numbers are not received in order)
- Lost an acknowledgement on the way.

The following points are very important to understand the TCP error control mechanism.

- The receiver checks errors in a segment. If there is no error, the acknowledgement is sent. If there is an error the segment is discarded only. No negative acknowledgement is sent.
- It is not necessary to send acknowledgement for each and every segment. If the acknowledgement number is n, this means all bytes up to n-1 were received by the receiver, without any error.
- If duplicated segments are received, if first segment has no errors, the second segment is ignored. That means, no effect from the duplicate segments.
- If out of order segments are received, they will not be checked until the previous segments are received. After receiving all the segments in order, it will acknowledge to the last segment.
- Sender maintains a timer for each segment. If acknowledgement is received the timer is discarded. If acknowledgement is not received before the retransmission timer expires, the segment is retransmitted.

Next we can discuss each possibility of errors.

4.9.1 Corrupted segment

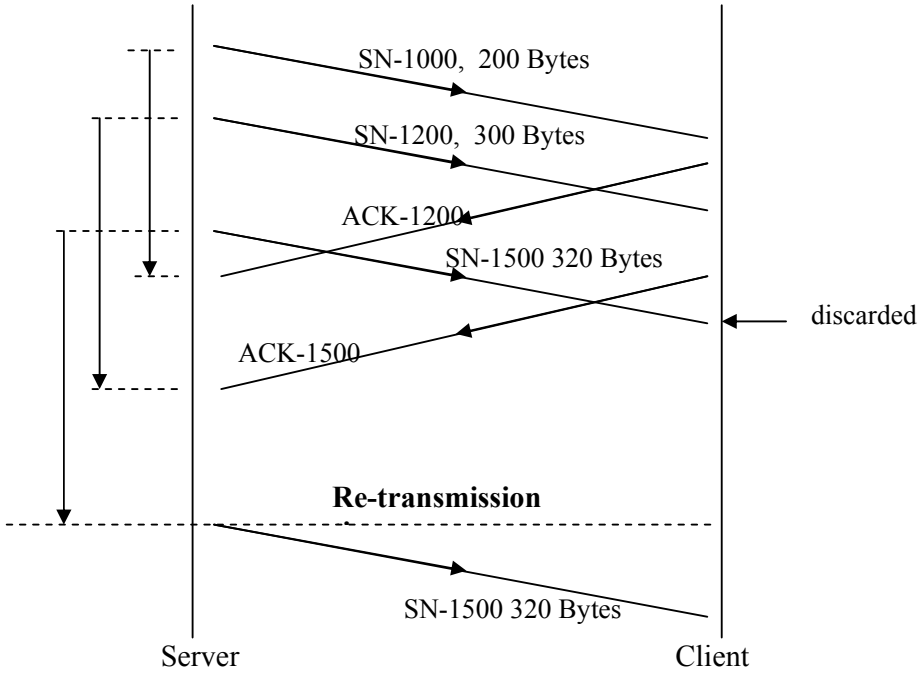


Fig 4 -22

The first segment (SN-1000) receives the acknowledgement. Therefore it discards its timer. The second segment (SN1200) also receives acknowledgement. But the third segment (SN1500) had errors. Therefore it was discarded by the receiver. The sender's third segment timer expired. The sender knows that there is a problem for the segment. Therefore it immediately retransmits the same segment. Note that

the timer is reset when it retransmits the segment. If any acknowledgement is received before the retransmission timer expires, the timer is discarded.

4.9.2 Out of order segment

The IP packets can travel through different routers. Therefore the segments can reach the receiver TCP layer out of order. The TCP layer waits until previous segment(s) are received and then acknowledges.

4.9.3 Duplicate segment

If IP packet travels through a long way the retransmission timer expire and sender retransmit the same segment. The receiver receives both segments. If the first segment received has no errors, it is accepted and acknowledged. The second segment is ignored by the TCP layer.

4.9.4 Lost acknowledgement

If the acknowledgement is lost , the sender does not receive it. Therefore the retransmission timer expires and retransmits the same segment. However it will duplicate the segment at the receiver. Since the receiver ignores the duplicate segment this will not be a problem.

4.10 Flow control

Same as data link layer, TCP also uses the flow control for the same purpose. It uses the sliding window mechanism.

4.10.1 Window size

The following points are important to understanding TCP flow control.

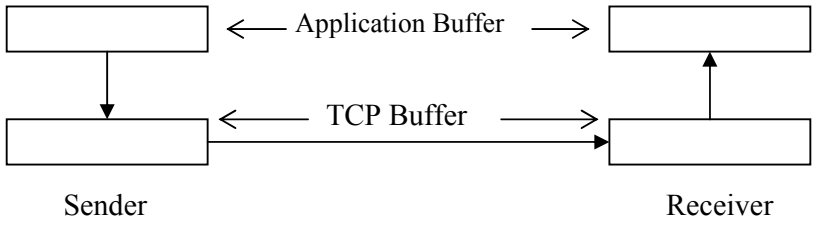
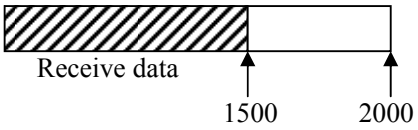


Fig 4 -23

The receiver TCP buffer should not be overflowed. It can be overflowed due to two reasons. One is the receiver TCP buffer receives data very fast. The second is the receiver application consumes data very slowly. In both cases receiver TCP

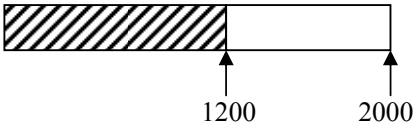
should inform the sender TCP how much bytes of data it can accommodate. It is called the window size.

Suppose the TCP buffer size is 2000 bytes.



If it receives 1500 bytes of data, 500 bytes buffer space is free. Then it informs the sender the window size is 500.

Then sender may send 500 bytes. By the time receiver receives this 500 bytes, the application may have consumed another 800 bytes, The remaining bytes in the buffer is 1500 – 800 = 700 bytes. With the new 500 bytes, now the buffer is filled with 700 + 500 = 1200 bytes.



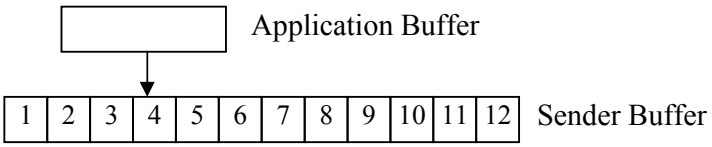
The free buffer size is 2000 – 1200 = 800 bytes. Then it informs to the sender, the window size as 800.

Then the sender may send 800 bytes. But the receiver application was busy and it did not consume any data. Therefore, now the buffer has 1200 + 800 = 2000 bytes. That means the buffer is full.

The receiver informs to the sender, the window size is zero.

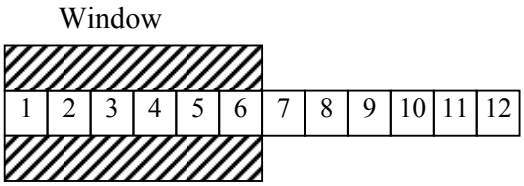
After receiver application consumed some data, say 200 bytes, the window size is informed as 200.

4.10.2 Sender buffer

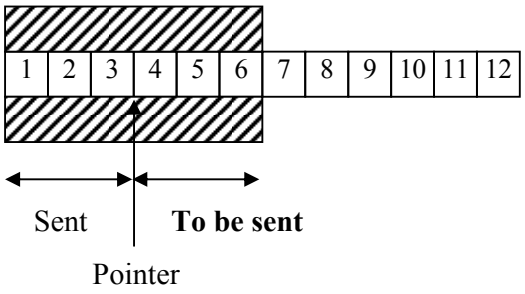


Suppose the sender buffer size is 12 bytes. It can he filled completely by the application data.

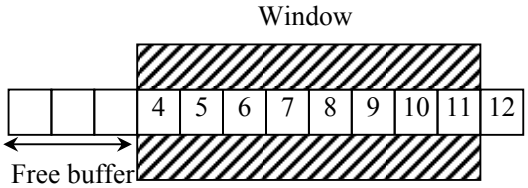
The amount of data that can be sent to the receiver is decided by the window size. If the window size informed by receiver is 6 bytes.



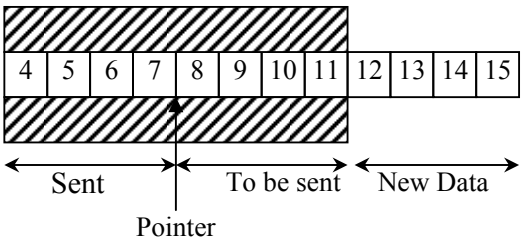
Suppose sender sends 1,2,3 bytes to the receiver. Then it keeps a pointer.



Suppose receiver receives three bytes and acknowledge with window size of 8. The acknowledged bytes are discarded from the buffer.



The sender gets the window size as 8 bytes. Suppose it sends another 4 bytes. Then it set the pointer to corresponding position. The first three positions can be filled with new three application data on 13,14 and 15 bytes.



The window at sender is called “Sliding Window” since it slides to right direction. Also it can be noticed that the size of the sliding window is a variable and it depends on the window size informed by the receiver.

How is the initial window size is determined?

At the time of connection establishment, the window size is informed with the ACK segment.

4.11 TCP Option

There are two types of options.

- Single-byte option
- Multiple-byte option

The actual optional information is in the Multiple-byte option. Single-byte options are used to fill the option field or to indicate the end of option.

4.11.1 Code

There is a 1-byte code number for each option. The single byte option has only the code number.

Code	Option
0 – 0000 0000	End-of-Option
1 – 0000 0001	No Operation
2 – 0000 0010	Maximum Segment Size
3 – 0000 0011	Window Scale Factor
8 – 0000 1000	Timestamp

4.12 Multiple-byte option

There are three types of multiple-byte options

- Maximum Segment Size
- Window Scale Factor
- Timestamp

4.12.1 Maximum Segment Size. (MSS)

This is a four-byte option. The code value is 2

Code	Length	MSS
1 Byte	1 Byte	2 Bytes

The length value is 4

This option indicates the maximum amount of data that can be included in a segment. Although the name is maximum segment size it indicates the maximum data size of a segment. The client informs the server that what is the MSS it can receive from server. Similarly the server informs to client the MSS it can receive from client. This parameter is decided at the connection establishment phase. The MSS cannot be decided at the time of data transfer.

If either side does not define the MSS, the default value 536 bytes is used by both client end server.

4.12.2 Window Scale Factor

This is a three-byte option. The code value is 3.

Code	Length	Scale Factor
1 Byte	1 Byte	1 Byte

The length value is 3.

The window size field of TCP header has 16 bytes. That means the window size can be in the range of 0 – 65535 bytes. Although it seems a very big number (65535) for some applications this amount is not economical. Suppose two computers connected by a fiber link that has throughput of 1,244,160 Mbps and they are in thousands of kilometers apart. Although the media has high throughput (more than 65535 bytes per segment) and there is a large propagation delay, it is economical to send more than 65535 bytes per segment. In order to avoid such problem, the window scale factor field is defined.

The relationship between new window size, window size defined in the header and the window scale factor is as follows.

$$\text{New Window Size} = \text{Window Size Defined in the Header} \times 2^{(\text{Window Scale Factor})}$$

It has 8 bits (1 Byte). Therefore it can go to maximum of 255. But the largest value allowed is 16.

If the window scale factor is 2, the window size defined in the header is 50,000.

The new window size is

$50{,}000 \times 2^2 = 50{,}000 \times 4 = 200{,}000$ bytes

4.12.3 Timestamp

This is a 10-byte option. The code value is 8

Code	Length	Timestamp Value	Timestamp Echo Reply
1 Byte	1 Byte	4 Bytes	4 Bytes

The real-time value is included in the Timestamp Value field by the sender (*A*). When the receiver (*B*) sends the acknowledgement the value in received Timestamp Value field is included in the Timestamp Echo Reply field. When *A* receive the acknowledgement segment it check the value in the timestamp echo reply field and the current time in the system. The difference is the Round Trip Time (RTT).

4.13 Single-byte option

There are two types of single byte options

- End of option
- No Operation

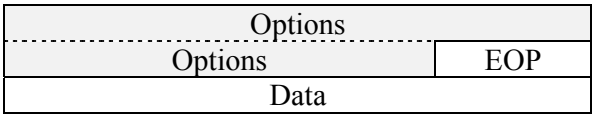
4.13.1 End of option (EOP)

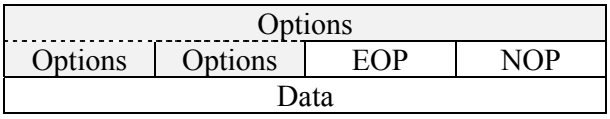
This is a one-byte option and it has only the code number.

Code
1 Byte

The code value is 0

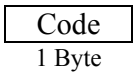
This option indicates the end of the option field. After this field the data is starting. If the EOP is not at end of 32bit word, the remaining bytes are filled with garbage (NOP). Therefore better definition for EOP is, No more options in the header and the remainder of the 32-bit word is garbage.





4.13.2 No operation option (NOP)

This is a one-byte option and it has only the code number.



The code number is 1.

This option is used as filler between options. It can be used to align the next option to a 16- bit or 32-bit boundary.

4.13.3 Examples of filling the options

Send only MSS option
The MSS has four bytes

Code	Length	MSS	
EOP	NOP	NOP	NOP
Data			

The complete TCP segment can be shown as below.

Standard TCP Header				(20 Bytes)
Code	Length	MSS		(8 Bytes)
EOP	NOP	NOP	NOP	Options
Data				

Three options send together. Suppose we send the options of MSS, Timestamp and window scale factor together.

MSS – 4 bytes
Timestamp – 10 bytes
Window Scale Factor (WSF) – 3 bytes

Standard TCP Header				(20 Bytes)
Code	Length	MSS		
NOP	NOP	Code	Length	(20 Bytes)
Timestamp Value				Options
Timestamp Echo Reply				
Code	Length	Scale Factor	EOP	
Data				

If you change the order of options, the above format can be changed.

4.14 TCP state Transition Diagram

Refer the last page

4.15 User Datagram Protocol (UDP)

4.15.1 Overview of UDP

The UDP operation is same as TCP with the following differences.

- UDP does not have a connection establishment process.
- UDP does not have a connection termination process.
- UDP does not have error control, flow control and congestion control mechanisms.
- UDP header has only 8 bytes

Since UDP does not get any feedback from the receiver, there is no guarantee of delivering data to the receiver by UDP. Therefore UDP is an unreliable simple protocol. Because of its simplicity it is used for specific applications especially the broadcast type applications.

4.15.2 UDP Port numbers

The concept is same as TCP. This has different port numbers for its applications.

The well-known port numbers of UDP is given in table shown below

Port Number	Application
69	TFTP
53	DNS
161	SNMP
520	RIP

4.15.3 UDP header format

Source port number (16 bits)	Destination port number (16 bits)
Data length (16 bits)	Checksum (16bits)

The explanation of each field is same as in TCP.

5 Internet Protocol (IP)

5.1 Overview

IP is the network layer protocol of TCP/IP. IP does not get any feedback from the receiver. Therefore error control, flow control and congestion control does not exist. Hence IP can be categorized as an unreliable protocol. However when it works together with TCP, the combination TCP/IP is reliable. Since TCP looks after the reliability of data, it is not necessary to do the same for IP also. But the UDP/IP is an unreliable combination.

The IP packets operate as datagrams. Therefore the IP packets originated from the same source can travel through different routes and reach the destination at different times. Therefore the IP packets may reach the destination out of order.

5.2 Features

5.2.1 Identification

Each IP packet is identified by a serial number called "Identification". If the identification of the first IP packet is n, the identification of the second IP packet is n+1. This will be helpful to the receiver to reassemble the packets in correct order, although they may receive in out of order.

5.3 Maximum Transmission Unit (MTU)

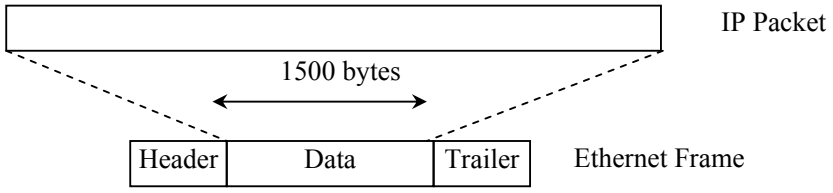


Fig 5-1

IP packet is sent to data link layer. If it is a LAN, the data link layer frame may be Ethernet. The maximum amount of data that can be accommodated in the Ethernet frame is 1500 bytes. Similarly other data link layer protocols also can accommodate some maximum amount of data. This amount is called Maximum Transmission Unit (MTU).

MTU for different protocols are given in table 5-1.

Protocol	MTU
Ethernet	1500
PPP	296
X.25	576
Token Ring (4mbp)	4,464
Token Ring (16 mbp)	17,4119

Table 5-1 MTU of different protocols

5.3.1 Fragmentation

If the IP packet size is bigger than MTU it should be fragmented.

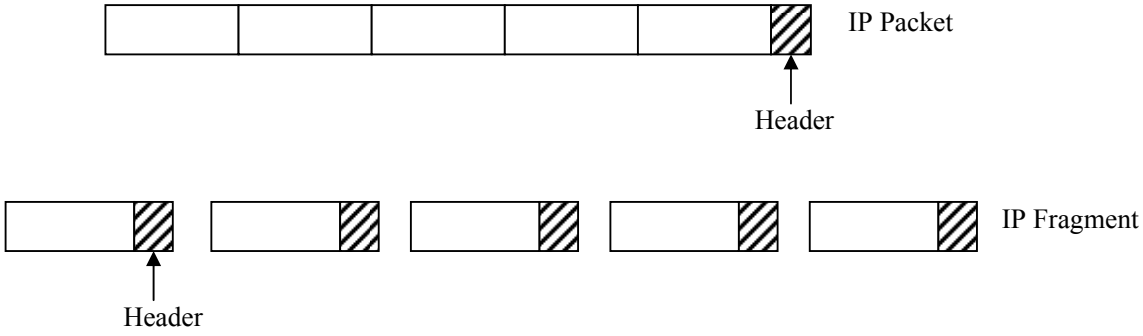
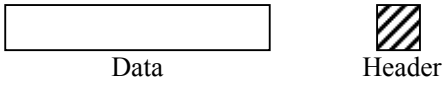
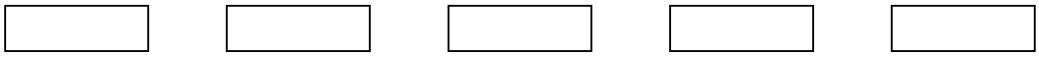


Fig. 5-2 Fragmentation

The data and header is to be separated

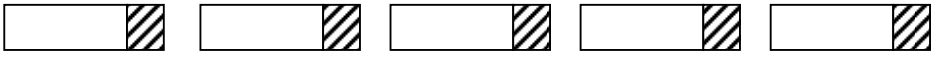


Data is to be chunked to required small parts



Add header to each data part. These are called fragmented IP packets. Total length should not exceed the MTU size.

eg. For Ethernet MTU = 1500 bytes

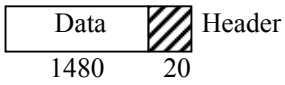


If the original packet has 6000 bytes of data. There can be five fragmented packets which have data, 4 x 1480 + 80 = 6000

1480, 1480, 1480, 1480, 80

The identification of each fragmented IP packet is equal to identification of original IP packet.

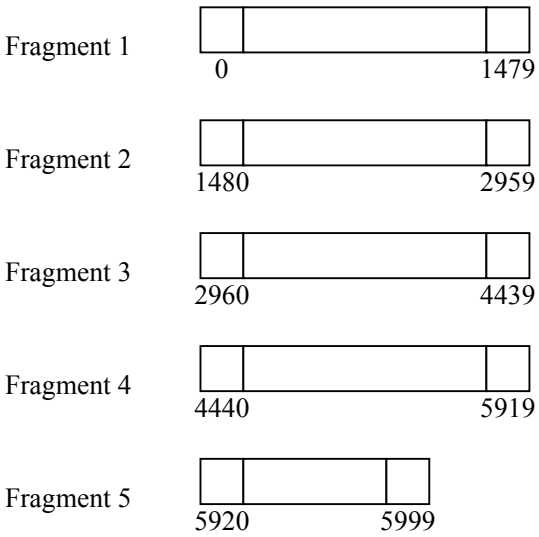
Eg. If the identification of original IP packet is 2000, identification of all five fragments is 2000.



In order to identity the order of fragments, another parameter called "fragmentation offset" is defined.

It says how many 8 bytes of data offset from the first fragmented IP packet.

Eg. In the above example the numbering of data bytes are as follows.



Offset value of fragment 1 = $\frac{0}{8} = 0$

Offset value of fragment 2 = $\frac{1480}{8} = 185$

(Fragment = starts with 1480. That means it is 1480 bytes offset from first packet)

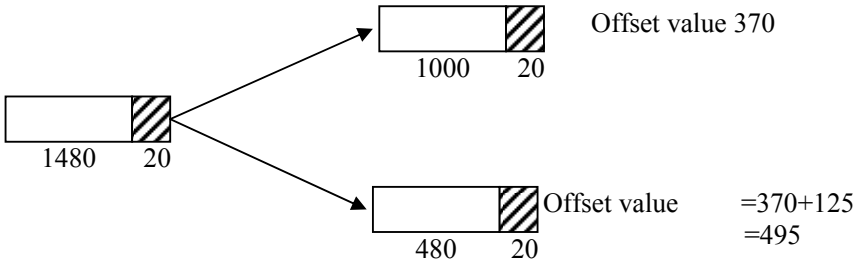
Offset value of fragments 3 $= \frac{2960}{8} = 370$

Offset value of fragment 4 $= \frac{4440}{8} = 555$

Offset value of fragment 5 $= \frac{5920}{8} = 740$

Therefore, except the last fragmented packet, all other fragmented packets should have some amount of data bytes, which could be divisible by 8.

The fragmented packets are combined (defragmented at the final destination. Therefore fragmented packets travel in dependently. They may travel through different routes to the destination. While it is traveling it can be further fragmented at another intermediate network. Suppose in the above example, fragment 3 further fragmented in to two fragments of 1000 and 480 bytes.



Now six fragments reach the final destination.

	No. of data bytes	Fragmentation offset	Identification
1	1480	0	2000
2	1480	185	2000
3	1000	370	2000
4	480	495	2000
5	1480	555	2000
6	80	740	2000

Table 5-2 Fragmentations and Offset Value

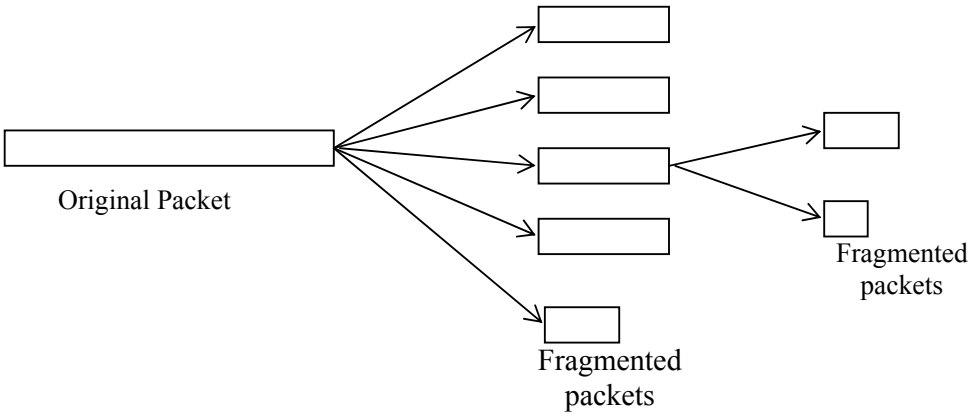


Fig 5-3

At the final destination, when combining fragmented packets, how can it identify the last packet? There should be a method to identify the last packet. For this purpose a flag is used. It is called "more fragments". If there are more fragments to receive the flag is set to 1. The value of flag of last fragment is 0.

In the above example, fragment 1 – 5 has the flag value of 1 and fragment 6 has the flag value of 0.

5.4 Time To Live (TTL)

IP packets may travel through many routers in the network. Each router routes the packet according to information in the routing table. If there is a problem in a routing table the packet may be sent in a wrong direction and it can be stranded in the network. This kind of stranded IP packets can even overload the network and finally crash the network. In order to avoid such a situation, a parameter called "Time To Live" (TTL) is defined for each IP packet. The value of TTL is decremented at each router. If the TTL value becomes zero at a particular router it will be discarded.

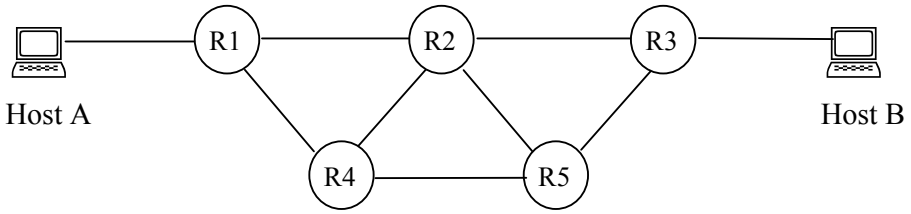


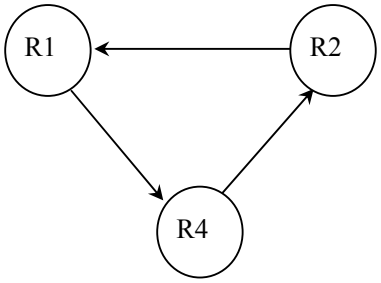
Fig 5-4

Suppose host A sends an IP packet to Host B. The TTL value is set to 6.

If the packet goes through Host A → R1 → R2 → R3 → Host B,

Router	TTL Value
R1	= 5
R2	= 4
R3	= 3

Suppose there is a routing problem and the packet loops through the routes
R1 → R4 → R2 → R1→ R4 → R2 → R1 → R4 → R2



Router	TTL Value	Action
R1	6 -1 = 5	
R4	5 – 1 = 4	
R2	4 – 1 = 3	
R1	3 – 1 = 2	
R4	2 – 1 = 1	
R2	1 – 1 = 0	Discards the Packet, Send ICMP message to Host A

The TTL value becomes zero at router R2. Therefore the IP packet is discarded and there will be no effect from the stranded packet

5.5 Protocol

The IP packet data can be UDP, TCP, ICMP, IGMP, EGP etc. TCP and UDP come from transport layer. Others directly come to IP layer. In order to identify the type of data a special field called "protocol" is used. Some protocol values are shown in table 5-3

Protocols	Value
ICMP	1
IGMP	2
TCP	6
EGP	8
UDP	17
OSPF	89

Table 5 –3 Protocols

5.6 IP header

The IP header is shown in the Fig. 5.5

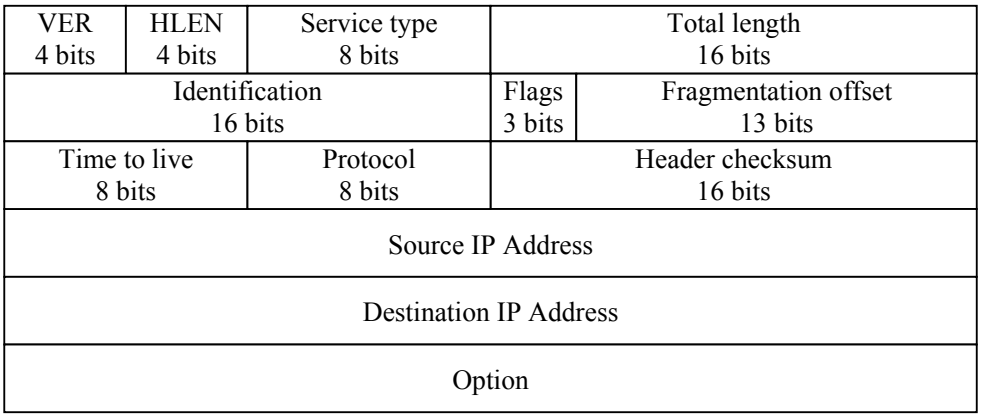


Fig 5-5 IP Header

5.6.1 Version (VER)

There are two versions of IP IPv4 and IPv6. Normally we use IPv4. This a 4 bits field and the value for value show the version eg: For IPv4 value is 0100.

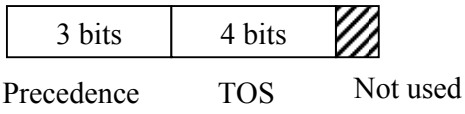
5.6.2 Header Length (HLEN)

This is a four-bit field. It gives how many four bytes are in the header. That means header (bytes) = HLEN x 4

The normal header size is 20 bytes. 20 = 5 x 4. Therefore HLEN value should be 5. That is 0101.

5.6.3 Service Type

This field has 8 bits. This has two parts



Precedence has 3 bits. It defines the priority of the packet. The lowest priority or normal packet has the value of 0. The highest priority is 7. However this is not used in version 4.

Type Of Service (TOS) bits is a 4-bit field, each bit having a special meaning. There are five types of services which is shown in table 5-4

TOS bits	Description
0000	Normal
0001	minimize cost
0010	maximize reliability
0100	Maximize throughput
1000	Minimize delay

Table 5-4 TOS Bits

The application can select a specific type of service. The default values for some applications are shown in table 5-5

Protocol	TOS Bits	Description
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP (data)	0100	Maximize throughput
FTP (control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP (command)	1000	Minimize delay
SMTP (data)	0100	Maximize throughput
DNS (UDP query)	1000	Minimize delay
DNS (TCP query)	0000	Normal
DNS (zone)	0100	Maximize throughput

Table 5-5 Protocols and their TOS Bits

5.6.4 Total length

This is a 16-bit field. This gives the total length of the IP packet.
That is data + header.

Total length = data length + header length

If total length value is 500, if this is a normal IP packet.

Header length = 20 bytes

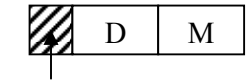
Data length = 500 – 20
= 480 bytes

5.6.5 Identification

Identification number of the IP packets. This is a 16 bit field

5.6.6 Flags

This is a 3-bit field.



D –Do not Fragment

M –More Fragments

If D = 1 it is not allowed to be fragmented. However if fragmentation is necessary for such a packet it will be discarded.

If M = 1, that means it is a fragmented packet and it is not the last fragmented packet.

If M = 0, it may be the last fragmented packet of some identification or it is not a fragmented packet.

5.6.7 Fragmentation offset

This is a 13-bit field. This gives the offset value of the fragment.

5.6.8 Time To Live

This is a 8 bit field which defines the number maximum number of hops the packet can travel.

5.6.9 Protocol

This is a 8-bit field which defines the protocol number.

5.6.10 Header Checksum

This is a 16-bit field packet. It does not check the errors for the whole packet. However it checks the errors of header. The header checksum is used for error detection of header. If errors are found in the header, the whole IP packet is discarded.

5.6.11 Source IP Address

This gives the IP address of the source

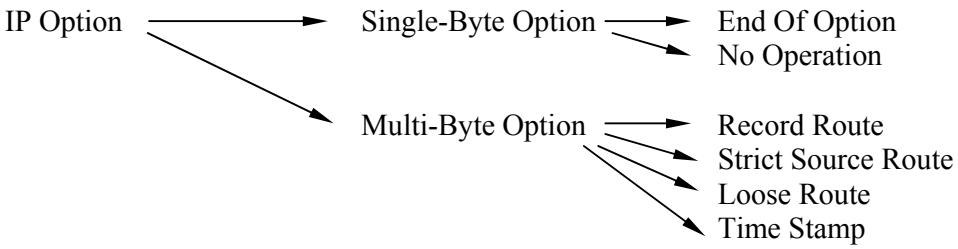
5.6.12 Destination IP Address

This gives the IP address of the destination

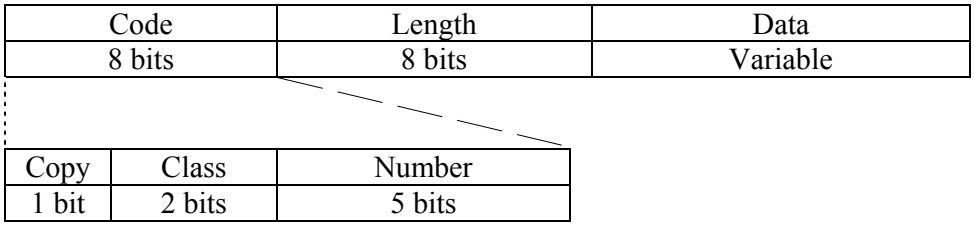
5.6.13 Options

The options field can go up to 40 bytes. This is a variable length field.

5.7 IP option



5.7.1 Format of the option field



5.7.2 Code field

Code field is divided into three parts. i.e. copy, class and number

5.7.2.1 Copy

This bit says how to copy the option field data in case IP packet is fragmented.

0 - copy the only to the first fragment

1 - copy into all fragments

5.7.2.2 Class

These two bits define the general purpose of the option.

00 – Datagram control
01 – Reserved
10 – Debugging and Management
11 – Reserved

5.7.2.3 Number

These five bits define the type of the option. There are six IP option.

00000 – End of Option
00001 – No Operation
00011 – Loose Source Route
00100 – Timestamp
00111 – Record Route
01001 – Strict Source Route

5.7.3 Length

The length field defines the total length of the option including the code field and the length field itself. This field is not present in all of the option types.

5.7.4 Data

Contains what ever the data relevant to the option used.

5.7.5 Option Type

5.7.5.1 End Of Option

Code – 0 00 00000

Used to indicate the end of option field

5.7.5.2 No Operation

Code – 0 00 00001

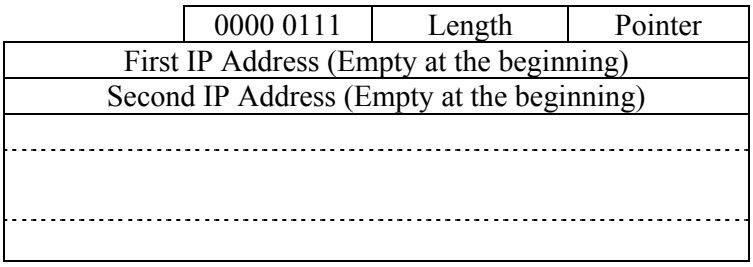
Used to fill the empty spaces as in the TCP option field

5.7.5.3 Record Route

Code value 7 – 0 00 00111

This code value 7 indicates that this option will be copied on to the first fragment only (if fragmented), used for Datagram control and number says 00111, this is record route option.

In record route, IP packet record the IP address of the router interface a sit leaves the router. Pointer will point to the next empty space to record the next IP address.



4 Bytes (32 Bits)

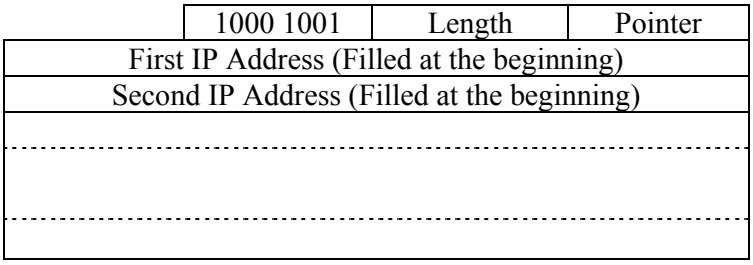
5.7.5.4 Strict Source Route

Code value 137 – 1 00 01001

This code value 137 means,

First bit 1 – will copy in to all the fragments
Next 00 means used for Datagram control
01001 mean this is the option strict source route.

Here format is more like the above, but the IP addresses of the routers to go through are filled at the beginning. So the IP packet will follow the specified route only (strictly).



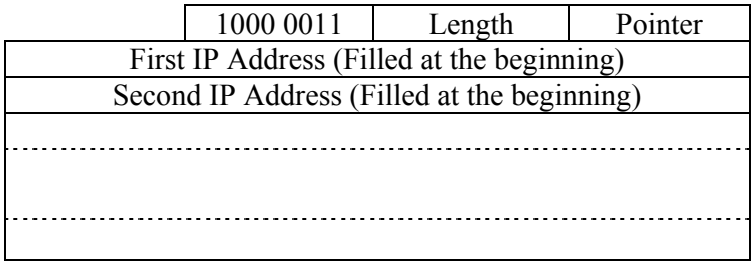
4 Bytes (32 Bits)

Also when it leaves a router, it replaces IP address (out going interface) in the correct order.

5.7.5.5 Loose Source Route

Code value 131 – 1 00 00011

This is also like the strict source route. The difference is the routers to go through are not strict as in the earlier case.

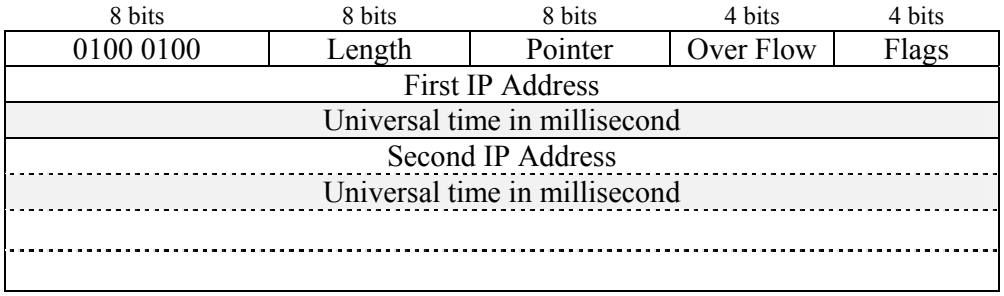


4 Bytes (32 Bits)

5.7.5.6 Timestamp

Code value 68 – 0 10 00100

In here it not only record the IP addresses it visits but also the universal time at each router is recorded. The format is little different in Timestamp.



4 Bytes (32 Bits)

Over flow indicate how many values were missed (not recorded) because of the lack of space for entries.

The use of the flag will decide what should be done at the router.

Flag

0- 0000- record only the universal time at each router
1- 0001- record time spent and the IP address
3- 0011- IP address of routers to visit is given, it records the outgoing IP addresses and universal time as well

Flag 0

Enter timestamps only

8 bits	8 bits	8 bits	4 bits	4 bits
0100 0100	Length	Pointer	Over Flow	0
Universal time in millisecond				
Universal time in millisecond				
Universal time in millisecond				
Universal time in millisecond				

4 Bytes (32 Bits)

Flag 1

Enter IP addresses and timestamps

8 bits	8 bits	8 bits	4 bits	4 bits
0100 0100	Length	Pointer	Over Flow	1
First IP Address (Empty at the beginning)				
Universal time in millisecond				
Second IP Address (Empty at the beginning)				
Universal time in millisecond				

4 Bytes (32 Bits)

Flag 3

IP addresses are given, enter timestamps

8 bits	8 bits	8 bits	4 bits	4 bits
0100 0100	Length	Pointer	Over Flow	3
First IP Address (Filled at the beginning)				
Universal time in millisecond				
Second IP Address (Filled at the beginning)				
Universal time in millisecond				

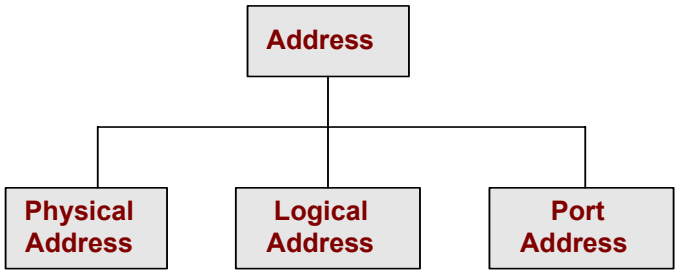
4 Bytes (32 Bits)

6 Addressing

6.1 Overview

Addressing means identification. In order to identify a house we use the address of the house. Similarly to identify a computer an address is used.

In TCP/IP there are three different levels of addresses.



The *physical address* and *logical address* are used to identify a computer. The port address is used to identify a process or program that runs in the computer. (Not the Input/Output Ports).

6.2 Physical Address

The physical address is in the Network Interface Card (NIC). It is a hardware setting and normally we cannot change that setting and it was set by the manufacturer of NIC. This is also called the *MAC address*.

For Ethernet, the MAC address is a 48 bit or 12 Hex number (one Hex number is 4 bits).

Example of a physical address is,

5AB387F1937C

The MAC address operates in the Data Link Layer (Layer 2).

Ethernet Frame

Here Source Address (SA) and Destination Address (DA) are MAC addresses of the originating and terminating hosts (Computers).

6.3 Logical Address

For the computers logical address scheme depends on the protocol used. The widely used protocol is TCP/IP and the logical address is called IP Address. (The logical address operates in the Network Layer-Layer 3).

6.4 IP Address

The worldwide IP Address (high level) is decided by Internet Assigned Numbers Authority (IANA). (Same as ITU-T for telephone numbers). Within Sri Lanka Internet Address authority is Council for Information Technology (CINTEC). (Same as TRC for telephone number system).

There are two versions of IP Addresses. IP version 4 (IPv4) and IP version 6 (IPv6). The latter is also called IP next generation (IPng). IPv4 is a 32-bit scheme and IPng is a 128-bit scheme.

6.4.1 IP Version 4 (IPv4)

The 32 bits are represented in following manner.
Byte 1. Byte 2. Byte 3. Byte 4
(Note: one byte is 8 bits).

The minimum value of a byte is 00000000=0
The maximum value of a byte is 11111111=255

Therefore, the minimum and maximum IP Addresses are,
0.0.0.0 and 255.255.255.255
(This is called the dotted decimal representation)

6.4.2 Network ID and Host ID

The network address is same as the telephone number system. We have the flexibility to have a telephone number system in a logical way. For instance all numbers in Kandy area starts with 081 Area Code. Therefore, if we get a call from Colombo to Kandy the Colombo Exchange analyzes the first three digits only. Then it can decide the correct route. Similarly all the exchanges up to Kandy analyze only the first 2 digits only. Therefore, the processing and routine becomes simpler.

Telephone number	-	Area Code	+	Telephone Number
IP Addresses	-	Network ID	+	Host ID

IP means Identification

Part of the IP Address is allocated to Network ID and the remaining part is allocated to Host ID (Computer ID).

If there are fewer networks, less number of bits can be allocated to Network ID.

Therefore, IANA divided the IP Address into 3 main classes, called Class A, Class B and Class C.

Class	Net ID	Host ID
A	1 Byte	3 Bytes
B	2 Bytes	2 Bytes
C	3 Bytes	1 Byte

Class	Theoretical Maximum number of Networks	Theoretical Maximum number of Hosts per Networks
A	2^8 =256	2^{24}=16777216
B	2^{16}= 65536	2^{16}= 65536
C	2^{24}=16777216	2^8 =256

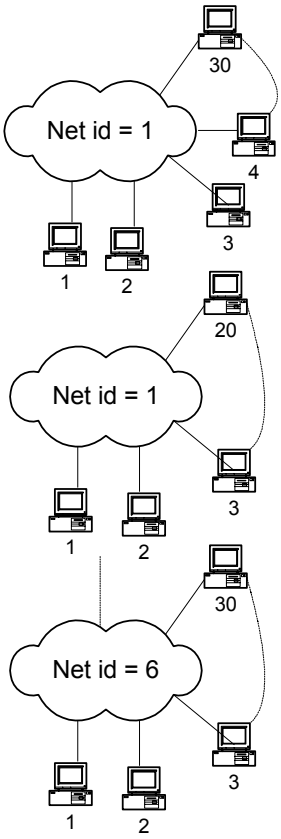
Note: *Also the Class D is introduced for Multicasting and Class E is reserved.*

Example:
Suppose a particular address scheme has only 8 bits. If 3 bits allocated for Network ID (Net ID) and remaining 5 bits is allocated to Host ID.

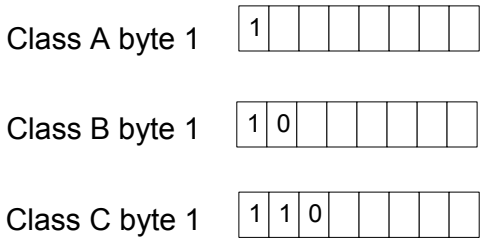
The maximum number of Network - $2^3 = 8$
The maximum number of Host per each network - $2^5 = 32$

Note: Both in Network ID and Host ID all 0s and all 1s are reserved for special purposes.

Therefore, the actual maximum no. Networks $= 2^3 - 2 = 6$
The actual maximum no. of Hosts per Network $=2^5 - 2 = 30$



Class A, B and C



Class	Minimum Network ID	Maximum Networks ID
A	00000000 0	01111111 127
B	10000000.00000000 128.0	10111111.11111111 191.255
C	11000000.00000000.00000000 192.0.0	11011111.11111111.11111111 223.255.255

6.4.3 Network Address

For the Network Address, the Host ID part of the IP Address will be considered as 0.

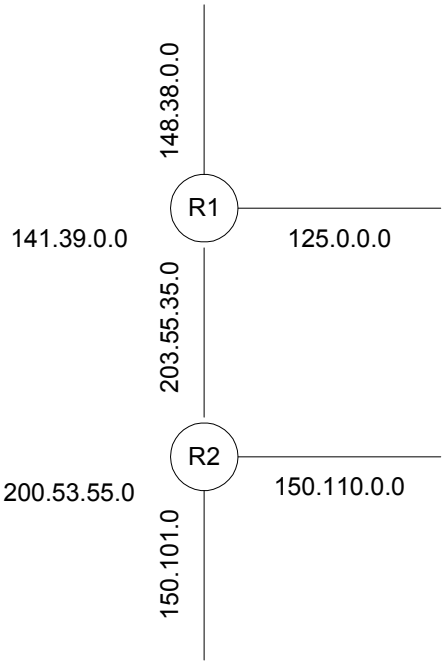
Eg: 103.58.35.1
This is a Class A address
Therefore, Net ID is = 103
Host ID is = 58.35.1
Network Address is = 103.0.0.0

153.105.25.10
This is a Class B Address
Therefore, Net ID is = 153.105
Host ID is = 25.10
Network Address is = 153.105.0.0

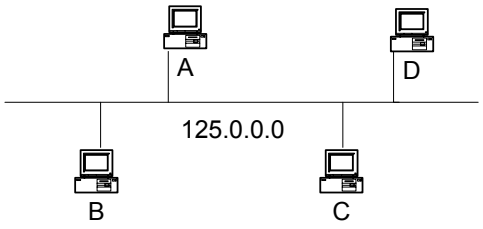
IP Address	Class	Network Address
140.35.45.55	B	140.35.0.0
50.60.70.5	A	50.0.0.0
201.35.40.201	C	201.35.40.0
125.38.55.185	A	125.0.0.0.
193.201.55.105	C	193.201.55.0
127.53.35.10	A	127.0.0.0

6.4.4 IP Address of a Router

A router has many ports. (LAN Ports and WAN Ports) An IP address can be assigned to each port.

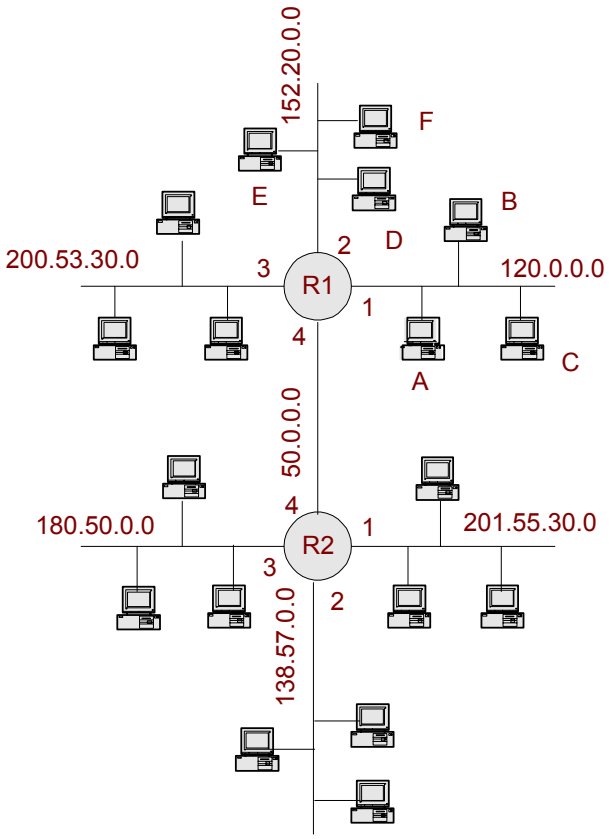


Example :
Router R1 and R2 have four ports each.
The network address of network that is connected to a port is shown in the figure.



The Host A,B,C and D of network 125.0.0.0 can be given the IP address 125.0.0.1, 125.0.0.2, 125.0.0.3 and 125.0.0.4

The IP Address of relevant LAN port of router can be assigned as 125.0.0.5 or any other IP Address such as 125.0.0.50.
(Note : *The LAN Port is also called as Ethernet Port*)



Host/Port	IP Address
A	120.0.0.1
B	120.0.0.2
C	120.0.0.3
Port 1 R1	120.0.0.50
D	150.20.0.1
E	150.20.0.2
F	150.20.0.3
Port 2 R1	150.20.0.50
Port 3 R1	200.53.30.50
Port 4 R1	50.0.0.1
Port 4 R2	50.0.0.2
Port 1 R2	201.55.30.50
Port 2 R2	138.57.0.50
Port 3 R2	180.50.0.50

6.4.5 Gateway IP Address

The IP address of router port which is connected to a particular LAN is called the “Gateway IP Address” of the LAN.

E.g. The Gateway IP Address of above example is as follows.

Network Address	Gateway IP Address
120.0.0.0	120.0.0.50
152.20.0.0	152.20.0.50
200.53.30.0	200.53.30.50
201.55.30.0	201.55.30.50
138.57.0.0	138.57.0.50
180.50.0.0	180.50.0.50

6.5 Public IP Addresses and Private IP Addresses

6.5.1 Public IP Addresses

The Internet is a Public Computer Network which is spread all over the world. (Same as Public Telephone Networks) Therefore, the IP addresses cannot be duplicated. Hence the allocation of IP addresses are controlled by the Internet Assigned Number Authority (IANA). They have already allocated different IP address ranges to different countries. The government of each country assigned a government body to deal with IANA and the government body controls the allocation of IP addresses within the country. In Sri Lanka this is handled by Council for Information Technology (CINTEC)

In telecommunication, since there are more than one telephone service providers in Sri Lanka, in order to not to duplicate the numbers, the logical pattern criteria of telephone numbers (upper level) is decided by the Telecom Regularity Commission (TRC). Similarly, worldwide it is decided by ITU-T.

The CINTEC assigns different range of IP address to different Internet Service Providers (ISPs). The ISPs allocate IP addresses to their customers. For example Sri Lanka Telecom provides allocates 8 IP addresses for each 64 kb/s leased line.

6.5.2 Private IP Addresses

If we have a network which is not connected to Internet (not a part of Internet) any IP address range can be used without obtaining any permission. However, it is not advisable to use any arbitrary IP address since the network may be connected to Internet in future. In order to avoid such problems, IANA reserved some IP address ranges for private use. These IP addresses are called private IP addresses.

Class	Private Network Address	No. of Networks
A	10.0.0.0	1
B	172.16.0.0 to 172.31.0.0	16
C	192.168.0.0 to 192.168.255.0	256

6.6 IP special addresses

There are 6 special IP addresses

		Net ID	Host ID	Remarks
1.	Network Address	Specific	All 0's	None
2.	Direct Broadcast Address	Specific	All 1's	Destination Address
3.	Limited Broadcast Address	All 1's	All 1's	Destination Address
4.	This Host on this Network	All 0's	All 0's	Source Address
5.	Specific host on this network	All 0's	Specific	Destination Address
6.	Loopback Address	127	Any (All 0's	Destination Address e.g.

			All 1's, some combinations)	127.0.0.0 127.255.255.255 127.0.0.1

6.6.1 Network Address

We have already discussed about the network address. There are three classes i.e Class A, Class B and Class C. (e.g. 192.168.0.0)

6.6.2 Direct Broadcast Address

This address is used to broadcast a message to another network i.e to send a message to all the computers in some other network. Here Net ID part will be a specific number and the Host ID part will be all 1's. (e.g. 192.168.255.255)

6.6.3 Limited Broadcast Address

This address is used to broadcast a message to the same network. i.e. to send a message to all the computers in the network as the send is in. Here destination IP address will be all 1's (e.g. 255.255.255.255)

6.6.4 This Host on this Network

Once a new computer is connected to a network and if that computer is not given an IP address (manually) it should get an IP address automatically. This is what happens if there is a DHCP server running. Initially the new computer will send a message requesting an IP address to the DHCP server. There the source address is all 0's (i.e. 0.0.0.0)

6.6.5 Specific Host on this Network

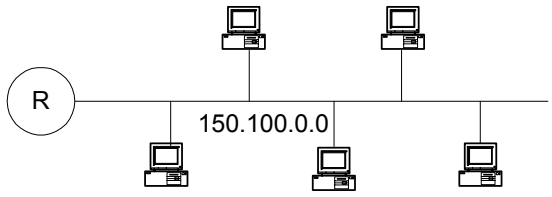
This is used to send a message to a specific host in the same network. (e.g. 0.0.60.30).

6.6.6 Loopback Address

This is used to mainly for testing. If a software to test with server and client program, we can run server and client program in that same machine and use the IP address in the form of 127.0.0.0. In this sort of situation IP packet never leaves the machine.

6.7 Subnetting (Classless Addressing)

Suppose you are given a network address 150.100.0.0 for your network.



You have the computers of Finance, Production and Administration Sections. In order to enhance the efficiency of network you want divide this into three networks. But you cannot get another two network addresses. This requirement can be satisfied from the same network address by using the subnet concept.

Now the IP address is divided into three parts.

Net ID Subnet ID Host ID

The important points to be noticed here is

1. The original Net ID number of bits is not changed.
2. Part of "Host ID" is allocated as the "Subnet ID".

In the above example 150.100 (i.e. first 16 bits) is not changed. In the remaining 16 bits the most significant bits are allocated as Subnet ID. Since we need three subnets at least two bits are required for Subnet ID.
i.e. 00, 01, 10 and 11

XXXX XXXX. XXXX XXXX. XXXX XXXX. XXXX XXXX
Net ID Subnet ID Host ID

The binary number of 150 and 100 are,
10010110 and 01100100 respectively.

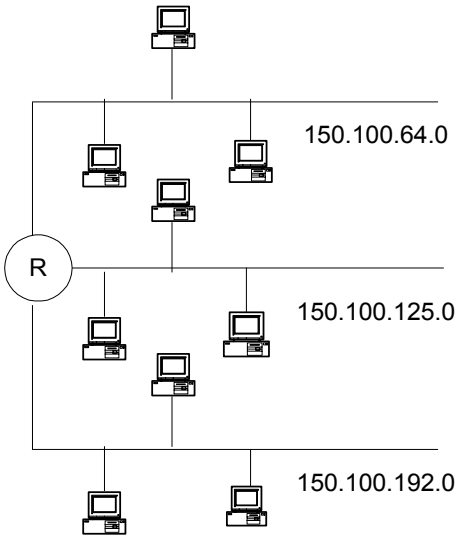
Therefore the subnets can be written as

Subnet 0 10010110.01100100.00000000.00000000
Subnet 1 10010110.01100100.01000000.00000000
Subnet 2 10010110.01100100.10000000.00000000
Subnet 3 10010110.01100100.11000000.00000000

Therefore in dotted decimal, it can be written as,
Subnet 0 address 150.100.0.0
Subnet 1 address 150.100.64.0
Subnet 2 address 150.100.128.0
Subnet 3 address 150.100.192.0

Note : Practically all possible values cannot be considerable as subnet address since some are used as special addresses.

For the above example the Finance, Production and Administration Sections can be put to three subnets as follows.



Consider the hosts in subnet 150.100.64.0. The IP addresses can be given as
150.100.64.1
150.100.64.2
150.100.64.3 etc.

This is called *classless addressing*.

In Class A, Class B and Class C addressing, by identifying the first byte, the class that belongs can be decided. Hence the number of bytes allocated to the Net id and Host id can be decided.

In classless addressing the number of bits allocated to network address cannot be decided. Therefore, the number of bits allocated for the network address is indicated with a "/" symbol. For the above example, 18 bits are allocated for the network addresses. Therefore, the IP address is written as,

150.100.64.1 /18
150.100.64.2 /18

150.100.64.3 /18

This information is sent through the network by another 32 bits where all bits allocated to Network Address part is indicated as '1' s and remain bits as '0' s and this is called the "Subnet Mask".

For above example the subnet mask will be,
11111111.11111111.11000000.00000000

In the above example, suppose we need six subnets. Then we have to allocate at least three bits for subnet ID.

Therefore, the subnet address will be
150.100.000XXXXX.XXXXXXXX
150.100.001XXXXX.XXXXXXXX
150.100.010XXXXX.XXXXXXXX
.
.
.
150.100.111XXXXX.XXXXXXXX

Subnet 0 address – 150.100.0.0
Subnet 1 address – 150.100.32.0
Subnet 2 address – 150.100.64.0
Subnet 3 address – 150.100.96.0
.
.
.
Subnet 7 address – 150.100.224.0

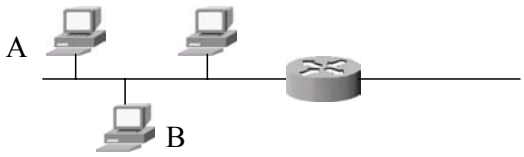
The IP address of host is subnet 1 can be written as,
150.100.32.1
150.100.32.2
150.100.32.3

The subnet mask will be,
11111111.11111111.11100000.00000000
255.255.224.0

7 Routing and Routing Protocols

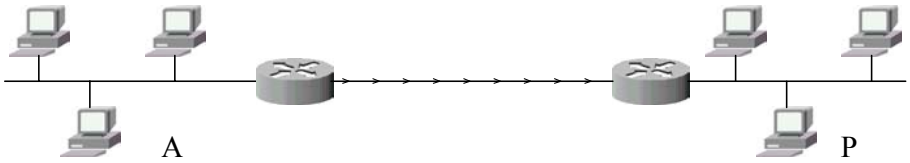
7.1 Direct Delivery

If we want to send a message to a machine in the LAN we can send that message directly



Send an IP packet from A to B
All the PC's will receive the data, so no need to direct (guide) the IP packet.
Since the IP packet is directly delivered this method is called " Direct Delivery"

7.2 Indirect Delivery



If we need to send a packet from A to P, need to go to the network i.e. that packet should go from router to router. This is called "Indirect Delivery".

In here, IP packet is analyzed at the router and correct path (most suitable path) is selected and the packet is sent though that path.

Indirect delivery is done using the routing strategies.

7.3 Routing strategies

There are four routing strategies

- Fixed Routing
- Flooding
- Random Routing
- Adaptive Routing

7.3.1 Fixed Routing

Routing information is centrally maintained. This is called a Directory. (A central database).

Advantage – Updating new information is easy as need to change at one location (central Database)
Disadvantages – Each and every IP packet should be analyzed and routing information is taken form the central database. Therefore network traffic may increase also at the central database it has to serve lots of requests from routers. Dynamic changes are not possible.

7.3.2 Flooding

When an IP packet comes to a router, router will send it on all paths. i.e. retransmitted to neighbors.

Advantage – Simple mechanism, IP packets are not analyzed at the router, most likely it will reach the destination.
Disadvantage – Causes high network traffic, duplicate packets might reach the destination

7.3.3 Random Routing

When an IP packet comes to the router, it decides the path randomly and sends the IP packet in that path.

Advantage – Will not cause unnecessary network traffic, simple
Disadvantage - No guarantee that the IP packet will reach the destination

7.3.4 Adaptive Routing

Each router maintains a routing table. Also it can be changed according to the network changes. (Adaptive)

Advantage – Network traffic is minimized. The best route will be selected most of the time
Disadvantage – Routers need to keep a routing table. Process each IP packet (mostly the case). Need to update routing tables automatically with the changes in the network.

7.4 Routing Methods used in Adaptive Routing

Methods used in adaptive routing,

- Next hop routing
    - Host specific
    - Network specific

And default routing

Host specific routing – Each router keeps a table entry for each host (one record for one host). Table entry has Host IP and the Interface

Host Address	Interface
A	E0
B	S0
C	S1

Disadvantages- Large number of records, if multiple paths are available number of records increases, table updating is difficult as it should be done for each and every host.(if the host IP changes)

Network specific routing – Each router keeps a table entry for each network (one record for one network). Table entry has Network address and Interface

Network Address	Interface
A	E0
B	S0
C	S1

Advantages - number of records are limited, table updates are not for each host but for a network. This updates is easy.

Default routing

This is just another record in the routing table to say if any of the records does not match with the IP packet destination IP what path should be taken. That is the default path to take.

7.5 Routing Table Update methods

Basically there are three methods to update routing tables.

- Connected
- Static
- Dynamic

7.5.1 Connected

Once the router is connected to the network its interfaces are given IP addresses. With that router automatically identifies the network addresses to which it connected.

7.5.2 Static

User can manually give routing table records. These types of records are called static. In here updating the routers are difficult if there are large number of routers available.

7.5.3 Dynamic

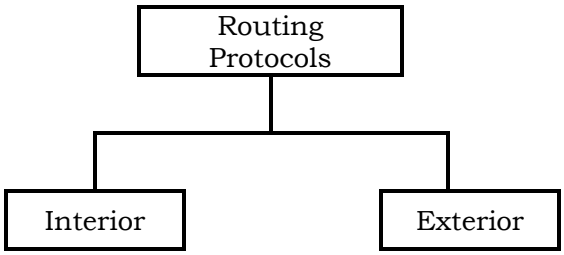
Using some protocols (set of rules) routing tables are updated automatically. Initially we might have only 'connected' records then we might add few 'static' records then it will get dynamic updates.

7.6 Features of routing protocols

- If there are any network changes (addition or removal or fault) it should be automatically updated in routing tables of all routers
- Such information should be immediately updated
- If there are many routes to a destination, the best route should be selected or
- Share the traffic through different route

7.7 Routing Protocols and Routing algorithms (Bellman-Ford & Dijkstras)

(Refer: William Stallings – Data and Computer Communications)

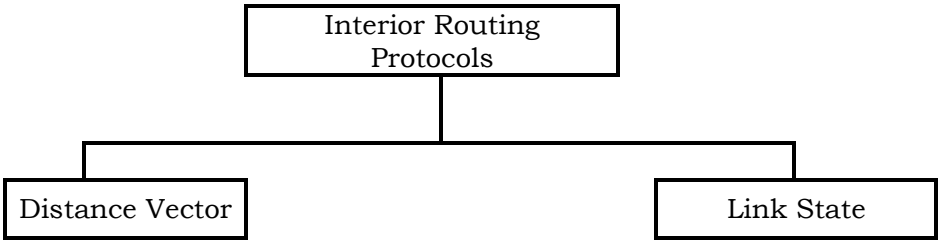


There are two types of routing protocols.

- Interior routing protocols.
- Exterior routing protocols.

A network comes under one administration is called an Autonomous System (AS). For example, SLT Intranet is an AS. An interior routing protocol is used within an autonomous system to update routing information. An exterior routing protocol is used to exchange routing information between two autonomous systems.

The interior routing protocols can be classified as follows.



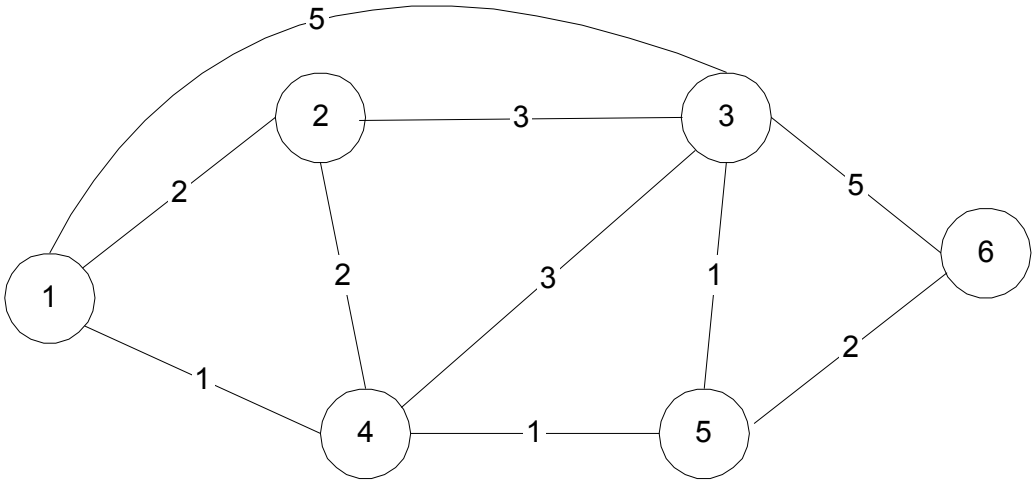
RIP is a distance vector routing protocol.
OSPF is a link state routing protocol.
BGP is an exterior routing protocol.

Any routing protocol can [Distance vector or Link State] use a Least Cost Algorithm to determine the optimum path of a packet. There are two types of widely used least cost routing Algorithms.

- Dijkstra's Algorithm
- Bellman Ford Algorithm

RIP uses Bellman Ford Algorithm and OSPF uses Dijkstra' s Algorithm.

In order to explain the two algorithms, we will use the following packet switched network, as an example. Each link indicates its path cost.

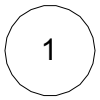


7.7.1 Dijkstra's Algorithm

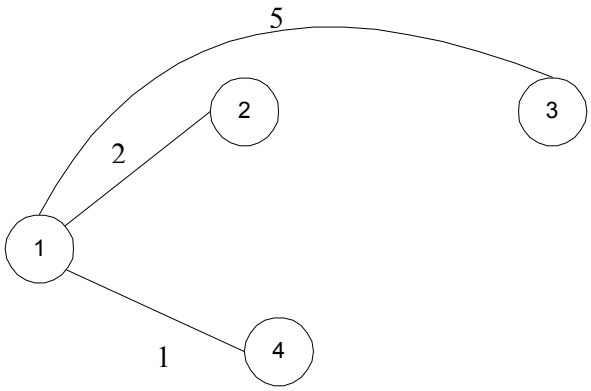
First we have to select one node as the source. Then we can explore the least cost path for every other node.

The method is as follows.

1. Select one node as the source. Suppose we selected the node 1.



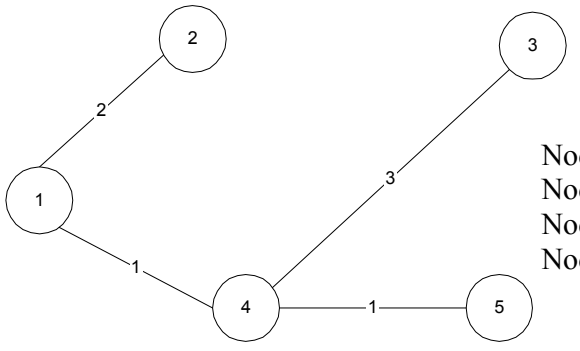
2. Find the neighboring nodes and find the distance to those nodes. Consider only single hop nodes.



D_n= cost of least-cost path for n^{th} node

Node 2	path 1-2	D_2=2
Node 3	path 1-3	D_3=5
Node 4	path 1-4	D_4=1

3. Next another node is selected. Suppose we selected the node 4. Least cost paths via both node 1 and node 4 are considered.

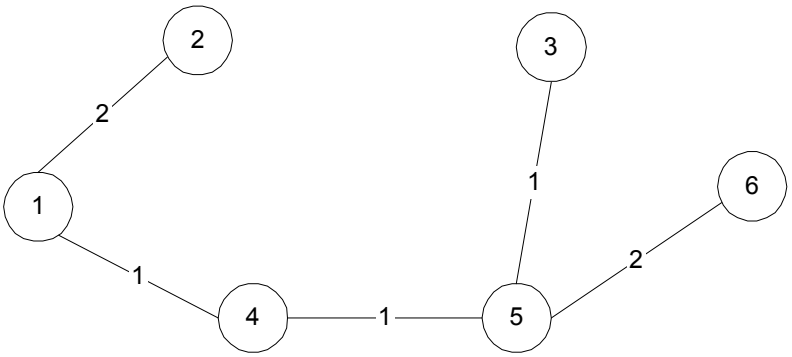


New paths and corresponding least costs are,

Node 2	path 1-2	D_2=2
Node 3	path 1-4-3	D_3=4
Node 4	path 1-4	D_4=1
Node 5	path 1-4-5	D_5=2

4. Another node is selected. Least cost path via all selected three nodes are considered. In this case node 2 is selected.
The diagram will be same as above and the path costs are not changed.

5. Next node 5 is selected.



New Least cost paths and cost of the least cost paths are,

Node 2	path 1-2	D_2=2
Node 3	path 1-4-5-3	D_3=3
Node 4	path 1-4	D_4=1
Node 5	path 1-4-5	D_5=2
Node 6	path 1-4-5-6	D_6=4

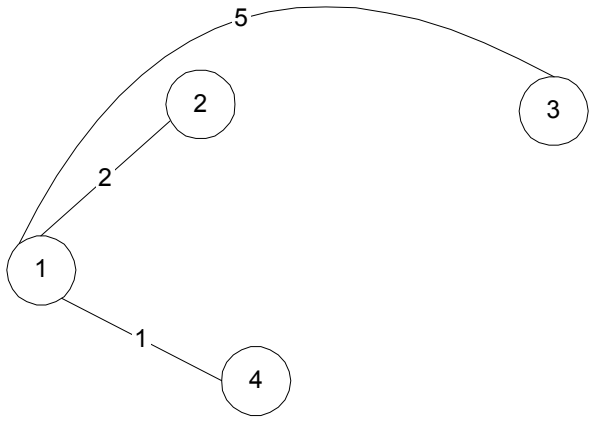
6. Next node 3 is selected. You can see that the diagrams are same as above and least path costs are not changed.
7. Next node 6 is selected.
   The least cost paths are not changed.
   Therefore, the final least cost paths from source 1 is,

Destination Node	Path	Least cost (for the path)
2	1-2	2
3	1-4-5-3	3
4	1-4	1
5	1-4-5	2
6	1-4-5-6	4

7.7.2 Bellman Ford Algorithm

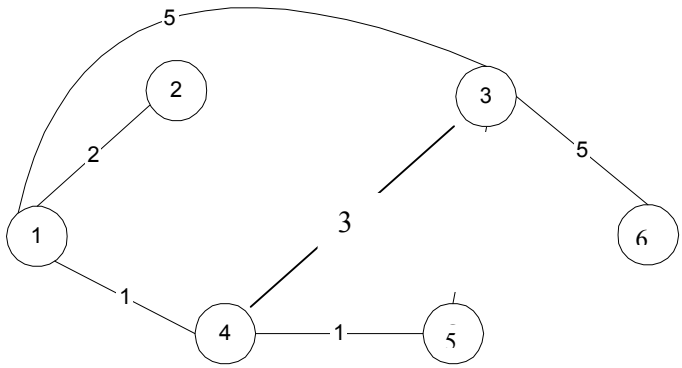
Firstly we have to select one node as the source. Consider that we selected node 1 as source.

1. Consider the nodes, which have maximum of one hop to the source.



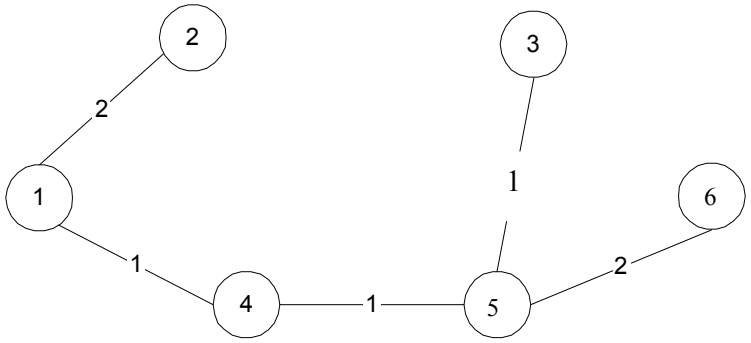
Destination Node	Path	Least cost (for the path)
2	1-2	2
3	1-3	5
4	1-4	1

2. Consider the nodes, which have maximum of two hops from source and find out the least cost path.



Destination Node	Path	Distance (Least cost for the path)
2	1-2	2
3	1-4-3	4
4	1-4	1
5	1-4-5	2
6	1-3-6	10

3. Consider the nodes, which have maximum of three hops from source and find out the least cost path.



Destination Node	Path	Least cost for the path
2	1-2	2
3	1-4-5-3	3
4	1-4	1
5	1-4-5	2
6	1-4-5-6	4

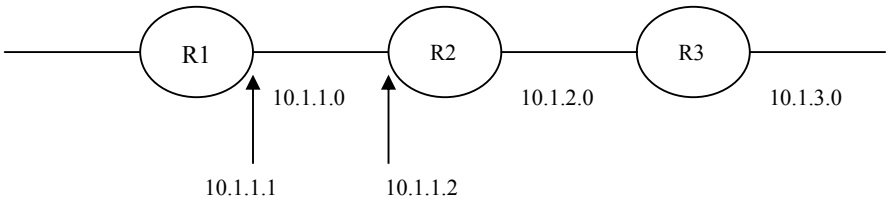
4. For the maximum of four hops the result will be same as three hops.
   Therefore, the least cost paths are
   $D_2 = 2$, $D_3 = 3$, $D_4 = 1$, $D_5 = 2$, $D_6 = 4$
   The result is same as the result of Dijkstra's Algorithm.

   This will be continued for other nodes also, for example the node 2 is considered next and do the same iteration.

7.7.3 Routing Table

Each router maintains a routing table. The main items of routing tables are,

Destination
Hop count
Next hop



7.7.3.1 Routing table of R1

Destination	Hop Count	Next Hop
10.1.2.0	1	10.1.1.2
10.1.3.0	2	10.1.1.2

7.8 Routing Information Protocol (RIP)

7.8.1 Timers in RIP (Periodic & Expiration)

7.8.1.1 Periodic Timer

Each router sends its routing table information to it neighbors every 30 second. In the router there is a timer that keeps the time for this purpose. Since it sends routing information every 30 second (periodically) we call it as the periodic timer. Even if a router does not receive the updates at a particular time at next update it will probably receive it. Since the updates are periodically sent it does not require highly reliable protocol to deliver it. Datagram approach it used for this purpose.

7.8.1.2 Expiration Timer

If a router does not get the updates from a neighbor, that could be due to many reasons, problem with the connection, problem with router etc… If the router does not get any updates for a long time it means it is a problem with the router and router removes the updates got from that particular router. This time period is call expiration time and timer involved in that is called expiration timer. For RIP it is 180 second.

7.8.2 Problems with RIP & Solutions

7.8.2.1 Slow convergence

Routing tables are sent to neighbors every 30 seconds (periodic time). If there are large number of routers in the network to get all the details (updated details) to each and every router will take some time. That is there is a delay in getting updated. This is called slow convergence.

Therefore information that needed to be updated immediately (change of a network, network is down etc…) is informed to the other routers without waiting for the periodic time. This is called triggered updates.

7.8.2.2 Route poisoning

If a network goes down the router that is connected to that network will get that information first. So that router updates its table saying this network is down (possibly down). In the routing table it says number of hops for that particular network as infinity (or in RIP as 16). This means that, particular network is unreachable. Assigning number of hops as 16 for a particular network is called Route poisoning.

7.8.2.3 Instability

Once a router (P) get some updates from other router (Q) router P will updates it routing table. At next update router P will send routing table information to the router Q and router Q might update the same information that it sent in the previous occasion. With time this will lead to having wrong updated tables in the routers and ultimately to an unstable situation. Solution for this problem is Spilt Horizon.

7.8.2.4 Split Horizon

Split horizon means, when the router sends routing table information to the neighbors, it will not send the information that it got from that particular router. So the routing table information will be selected and send. This process is called split horizon.

7.8.2.5 Hold down Timer

This is another timer used in RIP. Once a network goes down, that will be immediately sent to the other routers. Because of the network connections it has there is a possibility to get some wrong information about that particular network from other routers. Therefore once a network out information is received, the router will start the hold down timer, during which time any updates regarding that particular network is ignored. This timer is called hold down timer.

7.8.2.6 Poison Reverse

In general split horizon will apply for information passing. But the split horizon will not be applied in the case of the information like network is out. This situation is called poison reverse.

7.8.3 RIP Message Format

Command	Version	Reserved
Family		All 0s
Network Address		
All 0s		
All 0s		
Distance		

Command - It indicates whether it is a request message or response message.
Values - request - 1
response - 2

Version	-	Version Value 1 or 2 The above format is version 1.
Family	-	IP protocols value. Protocols number - 2.
Address	-	The address of the considered network.
Distance	-	Hop count to that network.

Note:

The network address and distance can have more than 1 depending on the amount of information required to be sent.

7.8.4 RIP Version 2 Message Format

Command	Version	Reserved
Family		Route tag
Network Address		
Subnet Mask		
Next hop address		
Distance		

Route tag – Autonomous System Address

7.8.5 Encapsulation of RIP Message

It encapsulate in UDP. The port number is 520.

8 Asynchronous Transfer Mode (ATM)

ATM is another technique, which is used to provide layer 2 connectivity. It has the ability to provide different type of treatments for different services. The following are different type of services.

8.1 Class A

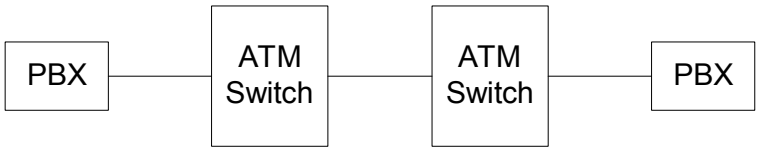
Fixed bit rate (bandwidth), real time connection, connection oriented.
Eg: Voice

8.2 Class B

Variable bit rate, real time connection, connection oriented.
Eg: Compressed video

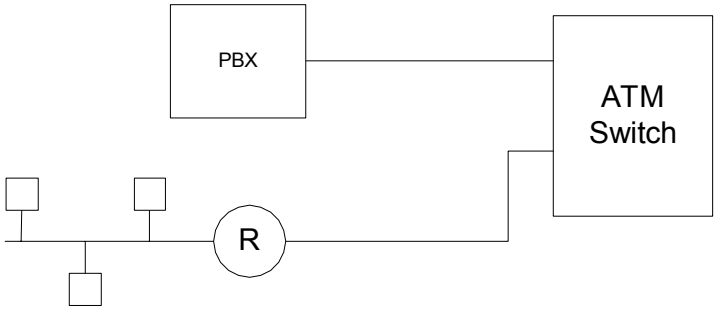
8.3 Class C/D/E

Variable bit rate, non-real time connection, connectionless.
Eg: Transfer of data between two LANs through a router.



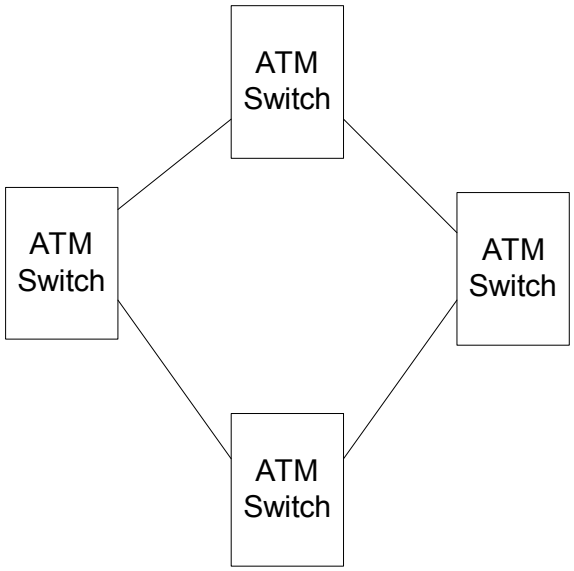
PBX provides voice services. Two PBX can be connected through ATM switch as shown in the figure.

Two remote LANs can be connected using ATM switches as shown in the above figure.



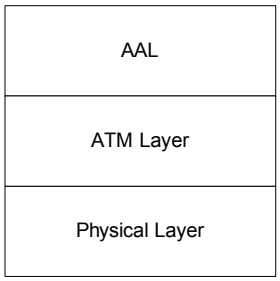
ATM switch has many ports. It can be connected to many services.

8.4 ATM Network



ATM network is made by connecting several ATM switches. They connect by using transmission links. Normally they are fiber. The bit rates are 155 Mb/s, 620 Mb/s, and 2.5 Gb/s etc.

8.5 ATM Protocol Architecture



8.5.1 ATM Adaptation Layer (AAL)

AAL has four Classes of services
AAL1 - Class A
AAL2 - Class B
AAL3/4 - Class C/D
AAL5 - Class E

Class A, B, C, D and E was explained earlier.

This layer does the segmentation of higher layer data.

ATM switch can be configured to one of the above services. The AAL layer segments the Application data and adds overhead bytes or adds overhead bits and segments the data so that total number of bytes is 48 bytes. Those 48 bytes are sent to ATM layer.

In AAL5, for the data received from the upper layer 8 bytes are added and then it will be segmented into 48 bytes. 8 bytes added are as follows,

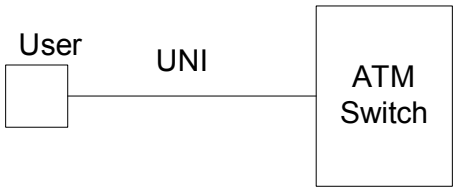
U-U - User to user ID - 1 byte
T - Type (Reserved) - 1 byte
L - Length - 2 bytes
CRC - 4 bytes

TCP/IP belongs to AAL 5 category.

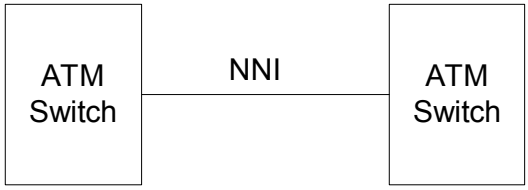
8.5.2 ATM Layer

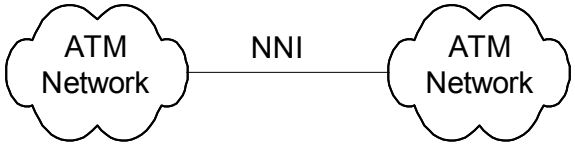
The ATM layer receives 48-byte cell payload (data) from ATM Adaptation Layer. The ATM layer adds 5-byte header to 48-byte payload. Then the ultimate size of a cell is 53 bytes.

There are two types of cell headers.



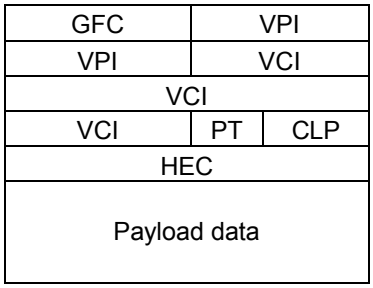
User Network Interface (UNI) header is used between a user and ATM switch.



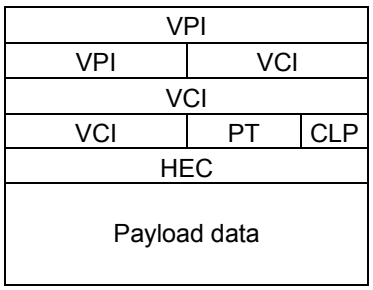


The Network to Network Interface header is used between two ATM switches or two ATM networks.

8.5.2.1 UNI Cell Format



8.5.2.2 NNI Cell Format



Generic Flow Control (GFC)- This field provides flow control for the UNI cell.

VPI- VPI is an eight-bit field in a UNI cell and a 12-bit field in and NNI cell.

VCI- VCI is a 16-bit field in both cells.

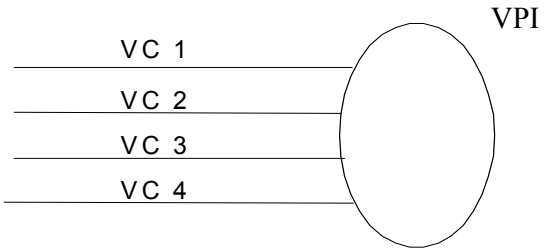
Payload type (PT)- This defines the type of payload.

Cell Loss Priority (CLP)- This bit indicates to a switch, which cell may be dropped and which must be retained.

Header Error Control (HEC)- This is an eight- bit field to detect multiple–bit errors and correct single-bit errors in the header.

8.5.3 VPI & VCI

ATM is a connection-oriented protocol. It established a permanent virtual channel between the source and destination. To identify a virtual channel Virtual Channel Identifier (VCI) is used. A group of virtual channels is called a virtual path. To identify virtual path, the Virtual Path Identifier (VPI) is used. VCI and VPI fields are in the header.



In the above example VPI consists of 4 virtual channels. i.e. VCI, VC2, VC3 & VC4.

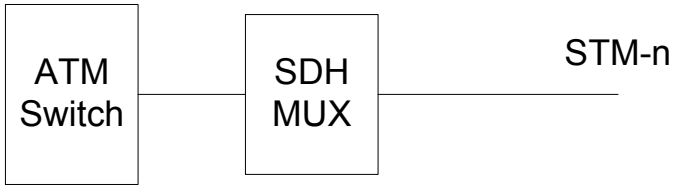
The ATM switches direct the cell to the out port by cell switching. It can be switching of virtual channels, virtual paths or combination of both.

ATM Switch	Input		Output	
	VPI	VCI	VPI	VCI
1	10	51	20	48
2	20	48	15	37

This layer provides routing, traffic management, switching and multiplexing services.

8.5.4 Physical Layer

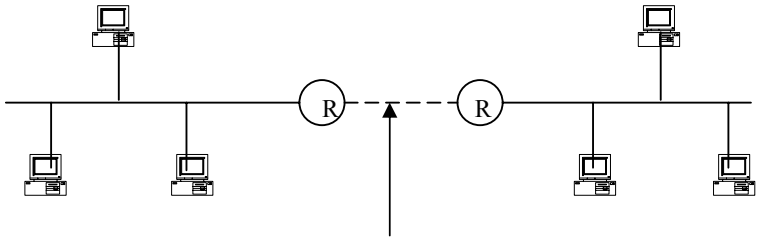
The physical layer defines the transmission medium, bit transmission, encoding and electrical to optical conversion.



The ATM switches can be connected to STM – n optical transmission system. The n can be 1, 4, 16, 64 etc…
STM – 1 = 155.52 Mb/s
STM – 4 = 622 Mb/s
STM – 16 = 2.5 Gb/s
STM – 64 = 10 Gb/s

9 MultiProtocol Label Switching (MPLS)

There are different techniques to get wide area network connectivity.



It can be a dial up, ISDN, Leased line, frame relay or ATM connectivity.

If the connectivity gives from frame-relay by a service provider the cost is comparatively less but all IP services will get same type of treatment. The ATM connectivity treats IP traffic as AAL5 type.

But the present day IP traffic contains different type of IP services such as normal computer to computer data, FTP data, VOIP, e-mail, web, different critical application etc… MPLS can provide different level of services for those applications. Also it has a very fast switching technique. Therefore it is safety for broadband services also.

MPLS is a layer 2 switching technique. It uses a label to decide the path. This label can be separately inserted or any other existing field in layer 2 protocols can be used.

Data Link Header	MPLS SHIM	Network Layer Header	Data

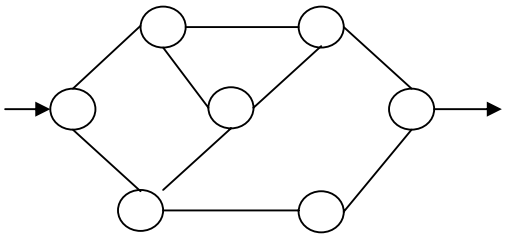
9.1 MPLS SHIM

This is a 32-bit field. The format is as follows

20 bits	3	1	8
Label	Exp	BS	TTL

Label - Actual label
Exp - Experimented use
BS - Bottom of stack bit
TTL - Time To Live
The function of TTL field is same as the function of TTL field of IP header.

9.2 MPLS network



Each router in the MPLS network performs the label switching. Therefore, they are called Label Switched Router (LSR). The router of the access to the MPLS network is called Label Edged Router (LER). LSR is a high-speed router device in the core of an MPLS network. LER is a device that operates at the edge of the access. The input LER is called ingress LER and the output LER is called egress LER.

9.3 Label creation

Initially the traffic (data) is divided in to several classes. This is called Forward Equivalence Class (FEC). Any traffic enter the network will belong to one of these classes.

Then the labels are allocated for different FECs.

This information is distributed among all routers in the MPLS network. This can be done by using a protocol called Label Distribution Protocol (LDP) or BGP or OSPF or Resource Reservation Protocol. (RSVP).

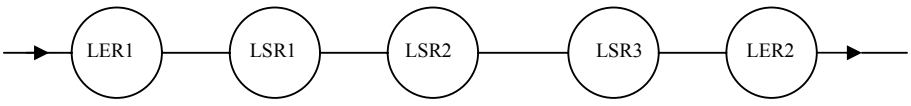
9.4 Table Creation

On receipt, of relationship between FEC and labels, each LSR will create an entry in Label Information Base (LIB). It has the information mapping between Label and FEC. Also LIB will maintain the path information, after creating the paths. That is mapping between input port, label and output port, label.

9.5 Label Switch Path Creation

A virtual path create in MPLS network is called a label switch path. The path can be decided in two different methods.

1) Hop by hop routing – each LSR independently selects the next hop for a given FEC.
2) Explicit router – The ingress LER defines the path.



LER1 will indicate FEC and send the request towards LSR1. This request will propagate through LSR2, LSR3 to LER2.

LER2 decides a label for input and inform to LSR3. This is output to LSR3. LSR3 decides a corresponding label for its input and inform to LSR2. This will continue up to LER1. Then the path for that FEC is automatically created.

9.6 Packet Forwarding

When a packet receives by ingress LER it check the relevant FEC. Then find out the relevant label from the LIB. Then it can check the output port and label. Then it will travel through the path and at the egress LER the label is removed and send out form MPLS network.