Computer Networks

For

Bachelor in Computer Engineering

Bachelor in Engineering of Information Technology

Bachelor in Software Engineering

Seventh Semester

Pokhara University, Nepal

By
Akhil Mathema

Chapter -1:

Introduction to Computer Networks

Resource Sharing: It refers to making all programs, equipments and especially data available to anyone on the network disregard to the physical location of the resource and the user.

Client-server model: Here, communication generally takes the form of a request message from the client so the server asking for some work to be done. The server then doest the work and send back the reply.

Uses of Computer Network:

a) resource sharing
b) high reliability
c) saving money
d) client-server model
e) scalability/ performance
f) powerful communication medium

Future merits of computer networks:

a) access to remote information
b) person to person communication
c) interactive entertainment

Types of transmission technology:

a) **Broadcast networks:** they have a single communication channel that is shared by all the machines on the network. Short messages, called packets in certain contexts sent by any machine are received by all others. Broadcast systems generally allow the possibility of addressing a packet to all destinations by using a special code in the address field. When a packet is transmitted, it is received and processed by every machine on the network.

b) **Point to point networks***:* it consists of many connections between individual pairs of machines. To go from the source to destination, a packet may have to undergo one or more intermediate points.

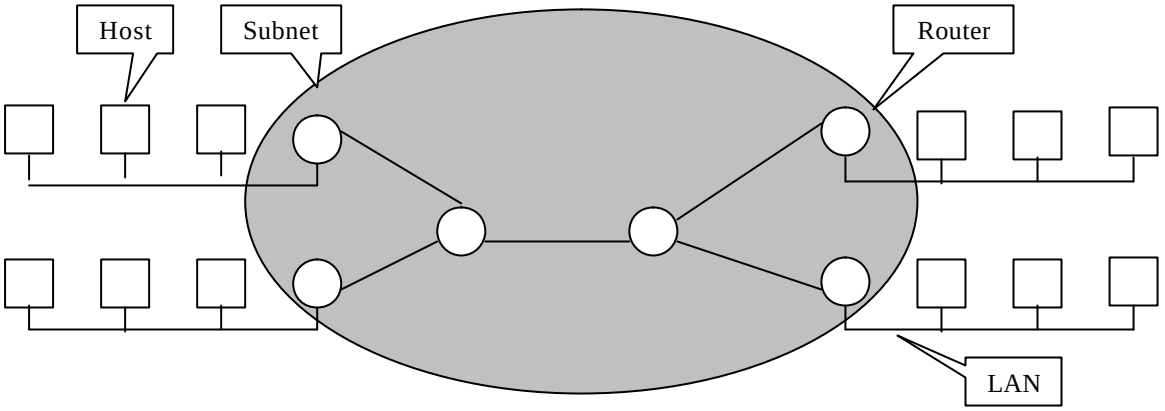
LAN: local area networks are privately owned networks within a single building of up to a few kilometers in size. LANs are widely used to connect PCs and workstations in company offices to share resources and exchange information.

Characteristics of LAN:

a) **Size***:* LANs are restricted in size and thus if there happens to be any faults, it is found and known in advance. Thus the network management is simplified.

b) **Transmission technology***:* LAN uses transmission technology consisting of a single cable to which all the machines are attached.

Subnet: is a collection of routers and communication lines that move packets from the source host to the destination host.



Relation between hosts on LANs and the subnet

Ethernet: is a bus-based broadcast network with decentralized control operating at 10, 100, or 1000 Mbps. Computers on an Ethernet can transmit whenever they want to; if two or more packets collide, each computer just waits a random time and tries again later.

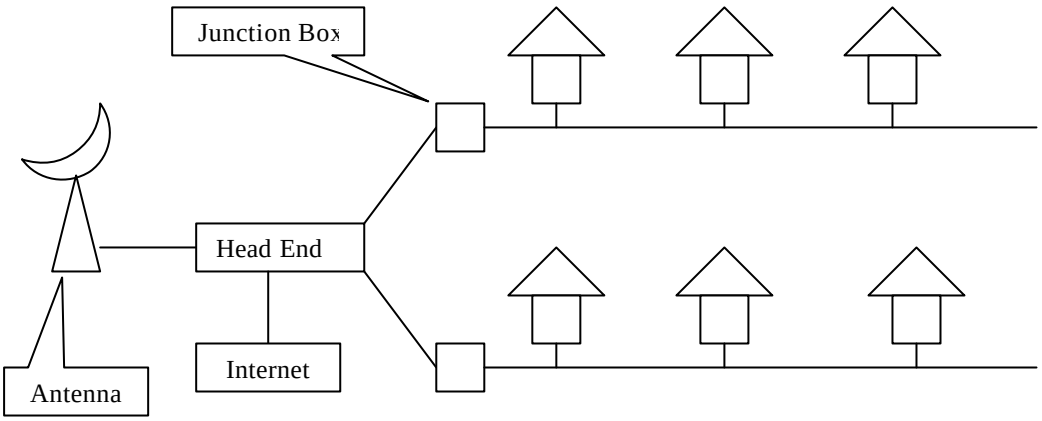
Ring: here, each bit propagates around its own; not waiting for the rest of the packet to which it belongs. Typically, each circumnavigates the entire ring in the time it takes to transmit a few bits, often before the complete packet has been transmitted.

Broadcast networks: are classified as follows:

a) **Static:** static allocation of channels divides up time into discrete intervals and run a round robin algorithm, allowing each machine to broadcast only when its time comes up. It wastes channel capacity when a machine has nothing to say during its allocated slot.

b) **Dynamic:** these are for the common channels either centralized or decentralized. In centralized channel allocation, there is a single entity (a bus arbitration unit) that determines who goes next. It might be done by accepting requests and making a decision according to some internal algorithm. In decentralized channel allocation, there is no central entity; each machine must decide for itself whether or not to transmit.

MAN: Metropolitan Area Network is basically a bigger version of LAN and normally uses similar technology. It might cover a group of nearby corporate offices or a city or public. MAN can support both data and voice; and might even be related to the local cable television network. As the Internet users started growing exponentially, the cable TV network was integrated with the Internet service. In this case, both television signals and the Internet are fed into the centralized head end for subsequent distribution to people's residence.



A Metropolitan Area Network based on cable TV

WAN: wide area network spans a large geographical area, often a country or continent. It contains a collection of machines called hosts, intended for running user applications (programs). The hosts are connected by a communication subnet which carries messages from host to host. Subnet consists of two distinct components: transmission lines and switching elements.

Transmission lines (circuits, channels, trunks) move bits between machines. The switching elements are specialized computers used to connect two or more transmission lines. When data arrive on an incoming line, the switching element mush chooses an outgoing line to forward them on. Since there is no standard terminology used to name these computers, they are variably called "packet switching nodes, intermediate systems and data switching exchanges", but a commonly used term is "router".

When a packet is sent from one router to another via one or more intermediate routers, the packet is received at each intermediate route in its entirety, stored there until the required output line is free and then forwarded. A subnet using this principle is called point-to-point, store-and-forward, or packet-switched subnet. Almost all WAN have store-and-forward subnets. When all the packets are small and of same size, they are known as **cells**.

Point-to-point subnet is dependent of the router interconnection topology. LANs were designed to have a symmetrical topology while WAN typically have irregular topologies.

Demerits of wireless LANs:

Though wireless LANs are easy to install, they also have some demerits as they have a capacity 1-2Mbps, which is much slower than wired LANs. The error rates are often much higher too and the transmissions from different computers can interfere with one another.

Gateway: is the machine which connects different and frequently incompatible networks together, providing the necessary translation, both in terms of hardware and software. A collection of interconnected networks is called an inter network or just internet.

A common form of internet is a collection of LANs connected by a WAN. The only real distinction between a subnet and a WAN is whether or not hosts are present. If the system within the cloud curve contains only routers, it is a subnet but if it contains both routers and hosts with their own users, it is a WAN.

Subnet: refers to the collection of routers and communication lines owned by the network operator like America Online.

Chapter – 2:

Network Architecture

Network Topology:

A network configuration is also known as a network topology. A network topology is the shape or the physical connectivity of the network. The following three objectives are to be met while designing a topology:

a) Provide maximum possible reliability to assure proper recipient of all traffic
b) Route the traffic across the least-cost path within the network between the sender and receiver
c) Give the end user the best possible response time and throughput.

The various common topologies are as follows:

a) Hierarchical Topology:

It is one of the most common topology. Here, the software required to control the network is relatively simple. But also contains the potential bottleneck problems as well as the reliability problem. If the upper layer machine fails, the network capabilities are lost completely. Hierarchical topology is also known as “vertical network” or “tree network”.

b) Bus Topology:

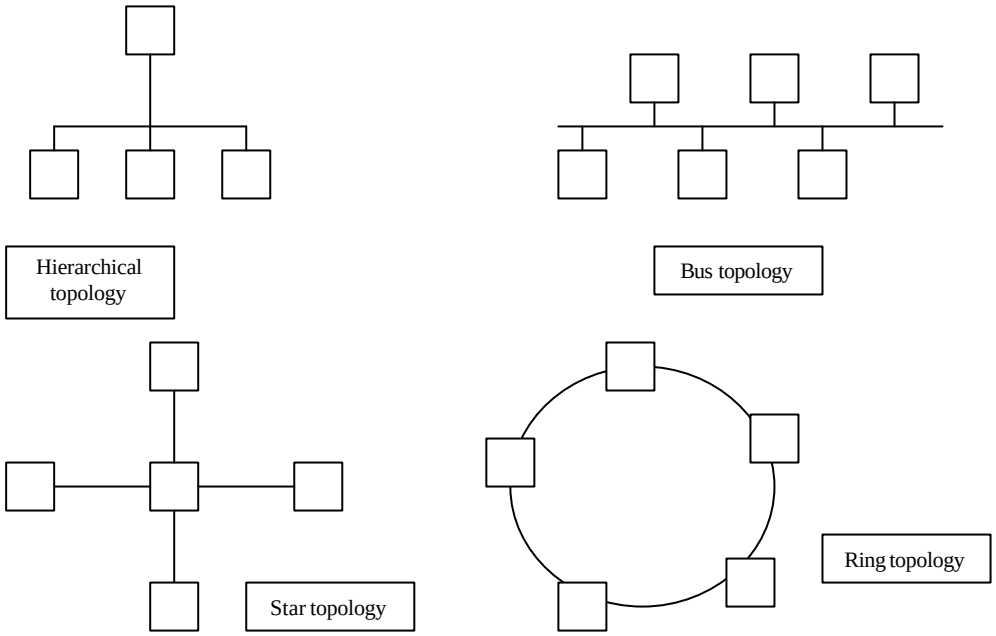
Bust topology is pretty popular in LAN. It is relatively simple to control traffic flow between and among machines on the network since the bus permits all stations to receive every transmission, i.e. a single station broadcasts to multiple stations. The main drawback of this topology is that it uses only one communication channel to serve all the devices on the network. Consequently in case of the failure of communication channel, the entire network is down. And the next problem is the difficulty in isolating faults to any one particular component inside the bus.

c) Star Topology:

At present, it is the most widely used topology for data communication systems. This came into existence since 1960s and is easy to control as the software is not so complex and the traffic flow is simple. All traffic originates from the hub of the star, the central node, which is responsible for routing traffic to the other components as well as for fault isolation. Fault isolation is relatively simple in a star network because the lines can be isolated to identify the problem. However, the star network is subjected to potential bottleneck and failure problems at the hub.

d) Ring Topology:

The data flow in only one direction, with one single station receiving the signal and relaying it to the next station on the ring. It is more user friendly since bottlenecks, such as those found in the hierarchical or star systems are very uncommon. Here, the token circumnavigates throughout the entire ring and the transmitting node holds the token and sends the network packet. Once done, the token is passed to an adjacent node.



Network Software:

To reduce the design complexity, most networks are organized as a series of layers or levels. The name, number, content and function of each layer differ from network to network. However in all networks, the purpose of each layer is to offer certain services to the higher layers.

Protocol:

A protocol is an agreement between the communicating parties on how communication is to proceed. Let a layer "n" on one machine carries on conversation with layer "n" on another machine. The rules and conventions used in thi8s conversation are collectively known as protocol. Practically, no data are directly transferred from layer "n" on one machine to layer "n" on another machine. Instead, each layer passes data and control information to the adjacent layer below it, until the lowest layer i.e. physical medium is reached through which an actual communication occurs. Between each layer, there is an interface that defines which primitive operations and services the lower layer offers to the upper one. A set of layers and protocols is called network architecture. Its specification must contain enough information to allow write the program or build the hardware for each layer so that it will correctly obey the appropriate protocol. Since a certain system uses a list of protocols, one protocol per layer is called a protocol stack.

Design issues for the layers:

In some systems, data flow only in one direction known as **simplex communication**, data flow in either direction but not simultaneously known as **half-duplex communication** and those data flow in both directions at once, known as **full-duplex communication.**

Interfaces and services:

The function of each layer is to provide services to the layer above it. The active elements in each layer are called entities. It can either be a software entity (say a process) or a hardware entity (say an intelligent I/O chip). Entities in the same layer on different machines are known as *peer entities.*

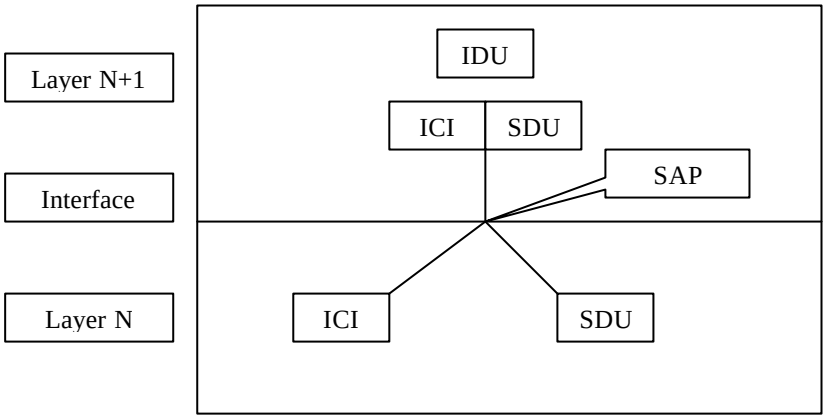
The entities in layer "n" implement a service used by layer n+1, where "n" is known as the "*service provider*" and the layer "n+1" is known as "*service user*".

Service Active Points:

The layer "n" SAPs are the places where layer n+1 can access the service s offered. Each SAP has an address that uniquely identifies it.

Interface Data Unit:

At a typical interface, the layer "n+1" entity passes an IDU to the layer "n" entity through SAP. The IDU consists of an SDU and some control information. The SDU is the information passed across the network to the peer entity and then up to layer "n+1". It should be noted that the control information is required to help the lower layer do its job but is not part of data itself. In order to transfer the SDU, the layer "n" entity may have to fragment it into several pieces, known as PDU (Protocol Data Unit) such as a packet. The PDU headers are used by the peer entities to carry out their peer protocol.



Connection oriented and connectionless services

Layers can offer two different types of service to the layers above them.

Connection oriented service is similar to telephone system. To use a connection oriented network service, the service user first establishes a connection, uses it and then releases the connection. It acts like a tube, where the sender pushes objects (bits) in at one end and the receiver takes them out in the some order at other end. In contrast, connectionless service is similar to postal system, where each message carries the full destination address and each one is routed through the system independent of all the others, here the first sent message can be delayed and later may arrive first. But it is not possible in connection-oriented service.

Unreliable connectionless service is often called datagram service, in analogy with telegram service that does not provide an acknowledgement back to the sender.

Service primitives:

A **service** is formally specified by a set of primitives (operations) available to a user or other entity to access the service. These primitives tell the service to perform some action or report on an action taken by a peer entity.

The relationship of services to protocols:

A **service** is a set of primates (operations) that a layer provides to the layer above it. The service defines what operations the layers are prepared to perform on behalf of its users. A service relates to an interface between two layers, with the lower layer being the service provider and the upper layer being the service user.

A **protocol** is a set of rules governing the format and meaning of the frames, packets or messages that a re exchanged by the peer entities within a layer. Entities user protocols in order to implement their service definitions. A protocol relates to the implementation of the service and as such is not known to the user of the service.

Network Architecture Model

Wireless LAN: A system of portable computers that communicate by radio can be regarded as a wireless LAN. These have different properties than conventional LAN and require special MAC sublayer protocols. A common configuration for a wireless LAN is an office building with base stations strategically placed around the building. All the base stations are wired together using copper or fiber. If the transmission power of the base stations and portables is adjusted to have a range of 3 or 4 meters, then each room becomes a single cell, and the entire building becomes a large cellular system. But when a receiver is within range of two active transmitters, the resulting signal will generally be garbled and useless.

IEEE named a wireless LAN as 802.11 and a common name for it is WiFi. Wireless LAN had to be basically operated in two modes:

1. In the presence of base station
2. In the absence of base station

In first mode, all communication had to go through the base station, called an **access point,** whereas in the later mode, the computers communicate each other directly. This mode is also known as **ad hoc networking.** By mid 1990s, as the Ethernet had already started to dominate the LAN, 8002.11 was decided to make compatible with Ethernet above the data link layer such that it should be possible to send an IP packet over the wireless LAN the same way a wired computer sent an IP packet over Ethernet. But in the physical and data link layers, there are several inherent differences between the Ethernet and a wireless LAN which had to be dealt with.

First, a computer on Ethernet always listens to the ether before transmitting. Only if the ether is idle, the computer begins transmitting. But it is not practicable in case of wireless LAN.

Second, a radio signal can be reflected off solid objects, so it may be received multiple times (along multiple paths), and hence such interference is known as **multipath fading.**

Third, most of the application will not be aware of mobility. For example, a Windows 95 machine may have one specific default printer. Which the computer is taken into new environment, it is unable to print at all until the changes are made.

Chapter: 3

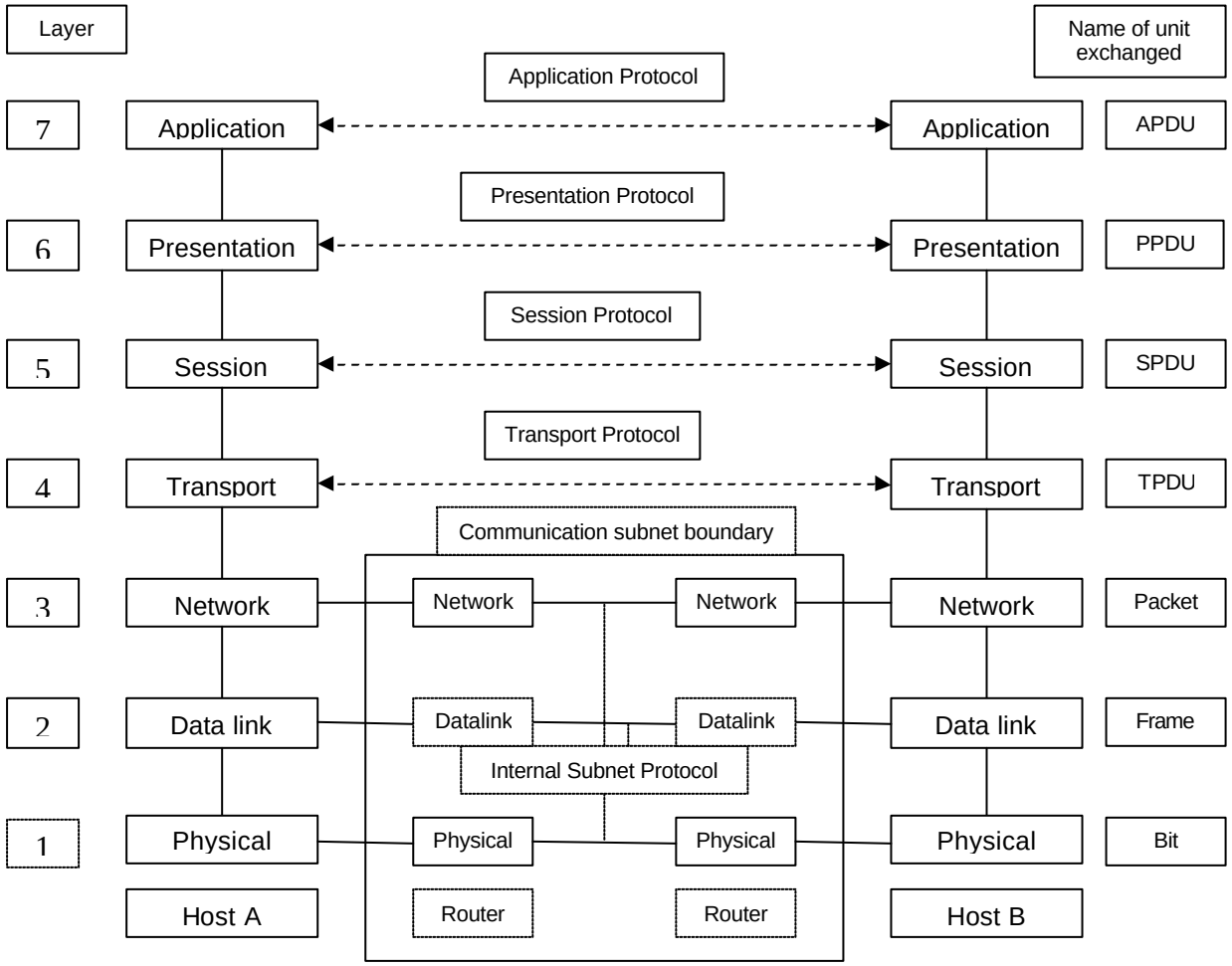
Reference Model

The OSI reference model:

Open Systems Interconnection Reference Model refers to connecting open systems i.e. systems are open for communication with other systems. Their principles are as follows:

1. A layer should be created where a different level of abstraction is needed.
2. Each layer should perform a well defined function.
3. The function of each layer should be defined by internationally standardized protocols.
4. The layer boundaries should be chosen to reduce the information flow across the interfaces.
5. The number of layers should be large enough that some distinct functions not be thrown together in the same layer of necessity and small enough that the architecture does not become unwieldy.

It should be noted that the OSI model itself is not network architecture, since it does not specify the exact services and protocols to be used in each layer. It only tells what each layer should do. It has also produced standards for all the layers, although these are not part of the reference model itself.



The OSI Reference Model

1. **The Physical Layer:** It is concerned with transmitting raw bits over a communication channel. The basic objective during the design process is when one side sends a 1-bit; it is received by the other side as a 1-bitm not as 0-bit. Here, the design issues largely deal with mechanical, electrical and procedural interfaces and the physical transmission medium, which lies below the physical layer.

2. **The Data-link Layer:** The main function of the data-link layer is to take a raw transmission facility and transform it into a line that appears free of undetected transmission errors to the network layer. It is done by breaking the input data into data frames (typically a few hundred or a few thousand bytes), transmit the frames subsequently and process the acknowledgement frames sent back by the receiver.

   The physical layer merely accepts and transmits a stream of bits without any concern with meaning or structure. It is up to the data-link layer to create and recognize frame boundaries. This can be done by attaching special bit patterns the beginning and end of the frame.

   A noise burst on the line can destroy a frame completely. In such cases, the data link layer software on the source machine can retransmit the frame. A duplicate frame could be sent if the acknowledgement frame from the receiver back to the sender were lost. It is up to this layer to solve the problems caused by damaged, lost and duplicate frames.

   Another issue that arises in the data link layer is how to keep a fast transmitter from drawing a slow receiver in data. Some traffic regulation mechanism must be employed to let the transmitter known how much buffer space the receiver has at the moment.

3. The Network Layer: It is related with controlling the operation of the subnet. A key design issue is to determine how packets are routed from source to destination. If too many packets are present in the subnet at the time, they will get in each other's way similar to bottlenecks. It is the duty of the network layer to control such congestion.

Many problems arise when a packets travel form one network to another up to its destination. The addressing used by the second network may differ from the first one; the packet size may be large enough for the second network, protocols may differ and so on. Hence network layer has to overcome all these problems to allow heterogeneous networks to be interconnected.

4. The Transport Layer: Its basic function is to accept data from the session layer, split it up into smaller units if need be, pass these to the network layer and ensure that the pieces all arrive correctly at the other end. Under normal conditions, the transport layer creates a distinct network connection for each transport connection required by the session layer. If the transport connection requires a high throughput, the transport layer might create multiple network connections, dividing the data among the network connections to improve throughput. But if it sounds expensive, the transport layer might multiplex several transport connections on the same network to reduce cost. Overall, the transport layer is required to make the multiplexing transparent to the session layer.

The transport layer also determines what type of service to be provided to the session layer, and ultimately the users of the network. The most popular type of transport connections is an error-free-point-to-point channels that delivers message or bytes in the order in which they were sent.

The transport layer is a true end-to-end layer. Because a program on the source machine carries on a conversation with a similar program on the destination machine, carries on a conversation with a similar program on the destination machine. While in case of lower layer, the protocols are between each machine and its adjacent neighbors; not by the ultimate source and destination machines, which may be separated by any routes. The layers from 1 to 3 are chained, while the layers fro 4 to 7 are end-to-end.

In addition to multiplexing several message streams onto one channel, the transport layer should also establish and delete connections across the network. Thus same kind of naming mechanism is required that describe whom a process on one machine wishes to converse. In the same way there has to be another mechanism to regulate the flow of information so that fast host cannot overrun a slow one. Such mechanism is known as flow control. *[Note: it is distinct from flow control between routers]*

5. The Session Layer: It allows users on different machines tot establish sessions between them. A session allows ordinary data transport like transport layer do, but also provides enhanced services useful in some applications. A session might be sued to allow a user to log into remote timesharing system or to transfer a file between two machines.

One of the services of the session layer is to manage dialogue control. Session can either allow traffic to go in both directions at the same time, or only in one direction at a time. Such service is known as token management. For same protocols, it is not possible to have same operation at a time. In order to manage these activities, the session layer provides tokens that can be exchanged.

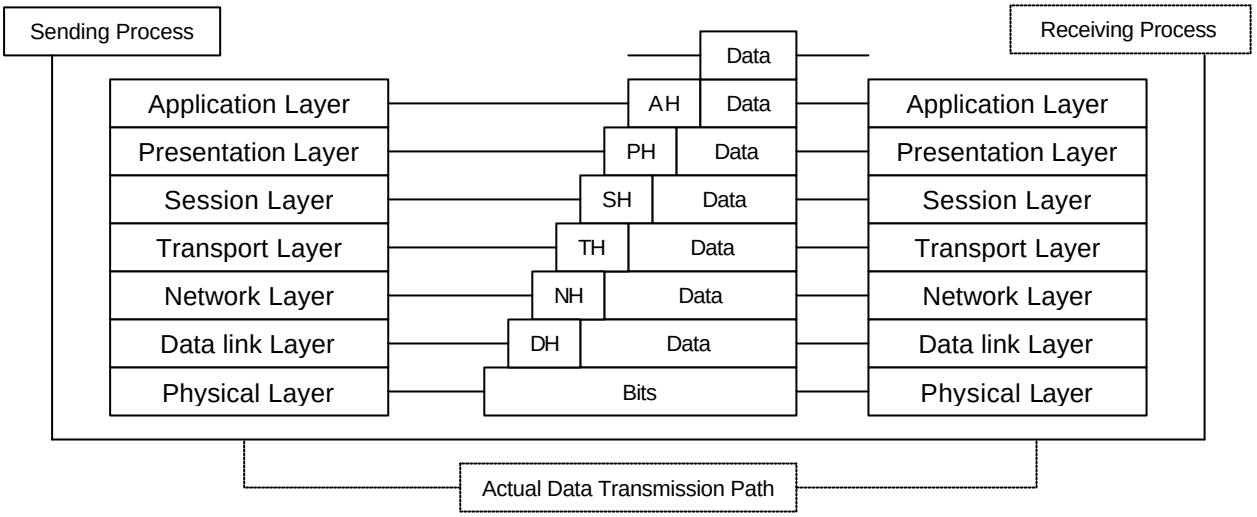
Another session service is synchronization. It avoids the whole transfer process to start again in case of the transfer being aborted in the middle. The session layer provides a way to insert a checkpoint into the data stream, so that after a crash, only the data transfer after the last checkpoint will have to be repeated.

6. The Presentation Layer: It performs certain functions that are requested sufficiently often to warrant t finding a general solution for them, rather than letting each user solve the problems. The lower layers just only move bits reliably from here to there, while the presentation layer is concerned with the syntax and semantics of the information transmitted.

One of a typical instance of presentation service is encoding data in standard way. Most user programs do not exchange random binary bit strings,; instead they exchange things like people's name, dates, amounts of money, invoices, etc. these items are represented as character strings, integers, floating-point numbers, data structures, etc. since different computers have different codes for representing character strings (ASCII and Unicode), integers and so on, in order to make it possible for computers with different representations to communicate, the data structures are to be defined in an abstract way along with a standard encoding to be used "on the wire". The presentation layer manages these abstract data structure and converts from the representations used inside the computer to the network standard representation and back.

7. The Application Layer: The application layer contains a variety of protocols that are commonly needed. Let there be hundred types of incompatible terminals in the world with different terminal types, each with different screen layouts, etc. For such environments, "Network Virtual Terminal" is to be defined which deals with different editors and programs. In order to handle each terminal type, a piece of software must be written to map the functions of the network virtual terminal on to real terminal. All the virtual terminal software is in the application layer.

The next function of this layer is file transfer. Different file systems have different file naming conventions, different ways of representing text lines and so on. Transferring a file between two different systems requires handling these and other incompatibilities. Thus electronic mail, remote job entry, directory look up, etc belongs to the application layer.



Networks:

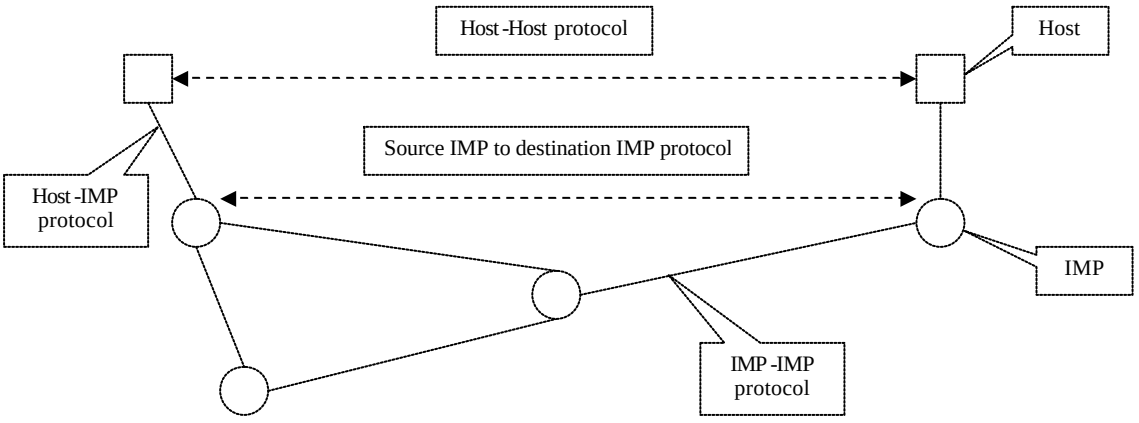
Numerous networks are currently operating around the world. Networks differ in their history, administration, facilities offered, technical design and user communities.

The ARPANET:

In mid 1960s, during the height of cold war, the Department of Defense wanted a command and control network that could survive a nuclear war. Traditional circuit-switched telephone networks were proved to be too vulnerable, since the loss of one line or switch would terminate all the conversations. In order to solve this problem, DoD started its research as ARPA (Advanced Research Projects Agency).

Initially ARPA had no scientists or laboratories; in fact, it had nothing more than an office and a small budget. It did its work by issuing grants and contracts to universities and companies whose ideas looked promising to it. Then after some discussions, with various experts, ARPA decided that it should have a packet-switched network, consisting of a subnet and host computers. The subnet would consist of minicomputers called IMPs (Interface Message Processors) connected by transmission lines. For high reliability, each IMP would be connected to at least two other IMPs so that if one IMP is destroyed, messages can be automatically routed along alternative paths. Here, each node of the network had an IMP and a host. A host could send messages up to 8063 bits to its IMP, which would then break up into packets of 1008 bits and forward them independently toward the destination.

Then ARPA started emphasis on the subnet. It selected BBN (a consulting firm) for it. BBN used specially modified Honeywell DDP-316 minicomputers with 12k 16-bits words of core memory as the IMPs. The IMPs were further interconnected by 56kbps leased telephone lines. The software was split into tow parts: subnet and host. The subnet software consisted of the IMP end of the host-IMP connection, IMP-IMP protocol and as source to IMP protocol designed to improve reliability.



The original ARPANET design

Similarly, outside the subnet also required the application software for host-IMP connection and host-host protocol. Later IMP software was changed to allow terminals to connect directly to a special IMP called TIP (Terminal Interface Processor), without having to go through a host. ARPA also funded research on satellite networks and mobile packet radio networks. It was found ARPANET protocols were not suitable for running over multiple networks. This caused the invention of TP/IP model and protocols (1974)/ TCP/IP was specifically designed to handle communication over internetwork, since more and more networks were being hooked up to the ARPANET. ARPA asked to develop a convenient program interface to the network (sockets) and many applications, utilities and management program to make networking easier. (FOR Berkeley UNIX). Previously most universities had 2nd or 3rd VAX computer and a LAN to connect them, but had no networking software, where 4.2 BSD introduced along with TCP/IP sockets and many network utilities, it was adopted immediately. With TCP/IP, it was easy for LAN to connect to the ARPANET. By 1983, the ARPANET was stable and successful with over 200 IMPs and hundreds of hosts. Then DCS (Defense Communication Agency) separated the military portion into a separate subnet, MILNET. During 1980s, additional networks, especially LANs were connected to the ARPANET. As the scale increased, finding hosts became increasingly expensive, so DNS (Domain Name Service) was created to organize machines into domains and map host names onto IP addresses. By 1990, the ARPANET was overtaken by newer networks and so it was shutdown and dismantled. But MILNET continues to operate, however.

The Internet:

As RCP/IP became only the official protocol in January 1983, the number of networks, machines and users connected to the ARPANET grew rapidly. When NSFNET & ARPANET were interconnected, the growth became exponential. In mid 1980s, people began viewing the collection of networks as an internet and later as the Internet. Growth continued, by 1990s, the Internet had grown to 3000 networks and 200,000 computers. In 1992, one millionth was attached, by 1995, there were multiple backbones, hundreds of mid-level networks, tens of thousands of LANs, millions of hosts and tens of millions of users. The most popular reason of the Internet is the TCP/IP reference model and TCP/IP protocol stack.

What does it actually mean to be on the Internet?

A machine is said to be on the Internet if it runs the TCP/IP protocol stack, has an IP address, and has the ability to send IP packets to all the other machines on the Internet. Traditionally, the Internet has four main applications: i) email ii) news iii) remote login iv) file transfer

i) Email: Since the early days of the ARPANET, electronic mail has been popular due to its ability to compose, send and receive. It is almost used by everyone in the Internet and email programs are available on virtually every kind of computer these days.

ii) News: News groups are specialized forums in which users with a common interest can exchange messages. Thousands of news groups exist on technical and non-technical topics.

iii) Remote Login: Using the telnet, rlogin and other programs, users anywhere on the Internet can log into any other machine on which they have an account.

iv) File Transfer: Using FTP programs, it is possible to copy files from one machine on the Internet to another. Vast number of articles, databases and other information are available this way.

World Wide Web:

By early 1990s, the Internet was largely populated by academic government and industries. One new application WWW changed all that and brought millions of new, non-academic users to the Internet. WWW made it possible for a site to setup a number of pages of

information containing text, pictures, sound and even video with embedded links to other pages. By clicking a link, the user is suddenly transported to the page pointed by that link.

Gigabit Testbeds:

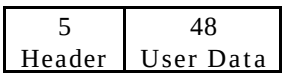
The Internet backbones operate at megabit speeds. With each increase in network bandwidth, new applications became possible and so the gigabit networks. Gigabit networks provide better bandwidth than megabit networks. Applications of such networks are in the field of telemedicine and video conferencing or virtual meeting.

Chapter – 4:

Physical Layers and its Design Issues

Broadband ISDN and ATM:

B-ISDN (Broadband Integrated Service Digital Network) offers video on demand, live television from many sources, CD-quality music, LAN interconnection, high-speed data transport and many other services that are not possible all over the telephone line. The underlying technology that makes B-ISDN possible is called ATM (Asynchronous Transfer Mode), because it is not synchronous as most long distance telephone lines are. The objective of ATM is to transmit all information in small, fixed-size packets called cells. The cells are 53 bytes long of which 5 bytes are header and 48 bytes are payload. ATM is both a technology and potentially a service. Services are sometimes called cell relay.



An ATM cell

The use of cell-switching technology is a revolution against the traditional circuit switching within the telephone system. This is because firstly, cell switching is highly flexible and can handle both constant rate traffic (audio, video) and variable rate traffic (data) easily. Secondly, at the very high speeds envision (gigabits per second are possible), digital switching of cells is easier than using traditional multiplexing techniques, especially using fiber optics. And thirdly for television distribution, broadcasting is essential; cell switching can provide this, while circuit switching cannot.

ATM networks are connection oriented. At first, the connection is to be setup for making a call to send a message. Then subsequent cells follow the same path to the destination. If cell 1 and cell 2 are sent in that order, they will arrive in the same order, never first 2 then 1.

ATM networks are organized like traditional WANs, with lines and switches (routers). Their standard speeds are 155Mbps and 622Mbps with the possibility of gigabit speeds later. The 155Mbps speed was chosen because this is the speed needed to transmit high definition television. The 622 Mbps was chosen for four 155 Mbps channels which could be sent over it. The long-distance telephone carriers and PTTs are interested in using ATM to upgrade the telephone system and compete with the cable TV companies in electronic video distribution. Similarly, the computer vendors see campus ATM LANs as the big money maker since they want low-speed analog telephone system to be replaced by high-speed digital system connecting by the Ethernets.

The ATM technology is especially to be used in the telephone system. Broadband ISDN using ATM has its own reference model that is different from the OSI and TCP/IP model. It consists of 3 layers, the physical, ATM and ATM adaptation layers; plus as per required by the user.

The physical layer deals with the physical medium: voltages, bit timing, etc. ATM does not prescribe particular set of rules, instead ATM cells may be sent on a wire or fiber by themselves, but they may also be packaged inside the payload of the other carrier system. Thus ATM has been designed o be independent of the transmission medium.

The ATM layer deals with cells and cells transport. It defines the layout of a cell and tells what the header fields mean. It also deals with establishment and release of virtual circuits.

Since most applications do not want to work directly with cells, the AAL (ATM adaptation layer) allows users to send packets larger than the cell. The ATM adaptation layer segments these packets, transmits the cell individually and reassembles at the other end.

The ATM model is defined as three-dimensional one. The user plane deals with data transport, flow control, error correction and other user functions, while the control plane is concerned with connection management.

The physical and AAL layers are each divided into two sub layers, one at the bottom that does the work and a convergence sub layer that provides the proper interface to the layer above it.

The PMD (Physical Medium Dependent) sub layer interfaces to the actual cable. It moves the bits on and off and handles the bit timing.

The TC (Transmission Convergence) sub layer is of the physical layer when cells are transmitted, the TC layer sends them as a string of bits to the PMD layer. At the other end, the TC sub layer gets a pure incoming bit stream from the PMD sub layer. Here, it converts the bit stream into a cell stream for the ATM layer. In case of AAL layer, SAR (Segmentation And Reassembly) sub layer breaks packets up into cells on the transmission side and puts them back again at the destination. Meanwhile the CS (Convergence Sub layer) makes it possible for ATM systems to offer different kinds of services to different applications like (file transfer and video on demand have different requirements concerning error handling, timing, etc.

TRANSMISSION MEDIA

Twisted Pair: It is the oldest and still most common transmission medium. It consists of two insulated copper wires, typically about 1mm thick. The wires are twisted together in a helical form (like DNA molecule) for reducing electrical interference from similar pairs. The most common application of the twisted pair is the telephone system. Twisted pair can run several kilometers without amplification, but for long distances, repeaters are required. They can be used for either analog or digital transmission. The bandwidth depends on the thickness of the wire and the distance traveled. Twisted pair cabling is in several varieties, two of which are important for computer networks.

Category 3 twisted pairs consist of two insulated wires gently twisted together. Four such pairs are typically grouped together in a plastic sheath for protection and to keep eight wires together.

Category 5 are similar to category 3 pairs, but with more twists per quality signal over a long distances. Both of these wiring types are known as UTP (Unshielded Twisted Pair).

Base-band Coaxial Cable: It is also known as “coax”. It has better shielding than twisted pairs, so it can span longer distances at higher speeds. There are two types of coax: 50-ohm cable, used for digital transmission and 75-ohm cable, used for analog transmission. A coaxial cable consists of a stiff copper wire as the core, surrounded by an insulating material. The insulator is encased by a cylindrical conductor and is further covered in a protective plastic sheath.

Its construction and shielding results a good combination of high bandwidth and excellent noise immunity. The bandwidth depends on the cable length; for 1-km cables, a data rate of 1 to 2Gbps is feasible. Longer cables can be used but at lower data rates or with periodic amplifiers. Users: telephone system, cable television and LANs.

Broadband Coaxial Cable: It is used for analog transmission on standard television cabling. As per telephone system, broadband refers to anything using analog transmission. Since broadband networks use standard cable television technology, the cables can e used up to 300Mhz and can run nearly for 100km because of analog signaling, which is much less critical than digital signaling. To transmit digital signals on an analog network interface must contain electronics to convert the outgoing bit stream to an analog signal and the incoming analog signal to a bit stream.

Fiber Optics: Using fiber optics, bandwidth more than 50,000Gbps (50TBps) can be achieved. But by the date, only 100Gbps has been proved to be possible. An optical transmission system has three components: the light source, the transmission medium and the detector. The transmission medium is an ultra-thin fiber of glass; the detector generates an electric pulse when light falls on it. By attaching a light source to one end of an optic fiber and a detector to the other, unidirectional data transmission system is achieved that accepts an electrical signal, converts and transmits it by light pulses and recovers output to an electrical signal at the receiving end. The transmission of light rays is based on the “total internal reflection”.

In a single fiber, multiple light rays can be transmitted at different angles. Each ray is said to have different mode so a fiber having this property is called a multimode fiber. If the

fiber's diameter is reduced to few wavelengths, the fiber acts like a wave guide, and the light can propagate in a straight like, without bouncing resulting single-mode fiber.

Transmission of light through fiber: The attenuation of light through glass depends on the wavelength of the light. The attenuation in decibels is given by the formula:

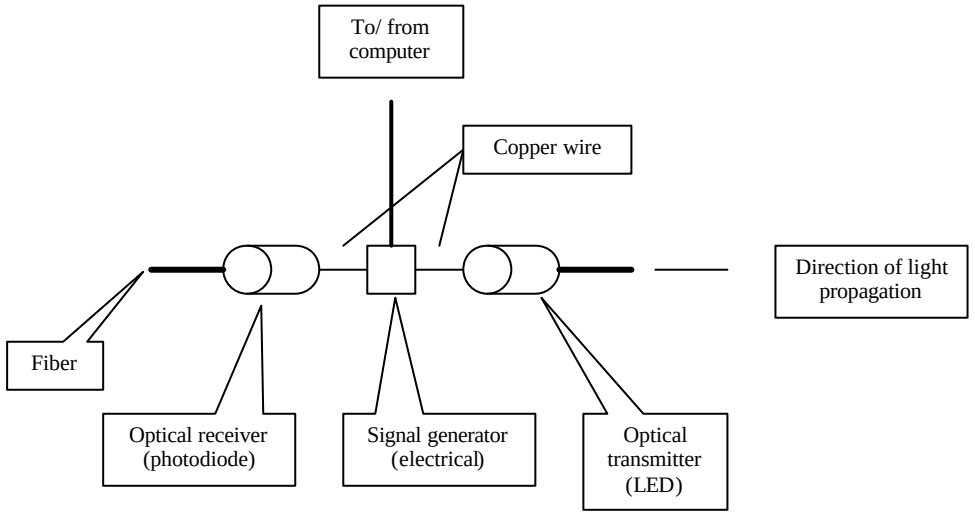
$$\text{Attenuation in decibels} = 10 \log_{10} \frac{\text{transmitted power}}{\text{Received power}}$$

Light pulses disperse in length as they propagate. The amount of dispersion is wavelength dependent. It can be reduced by increasing the distances between them and it is only possible by reducing the signal rate. It has been discovered that by making pulses in a special shape related to the reciprocal of the hyperbolic cosine, all the dispersion effects can cell out, and it may be possible to send pulses for thousands of kilometers without appreciable shape distortion. These pulses are known as solitons.

Fiber Optics Network: Fiber optics can be used for LANs as well as for long-haul transmission, although tapping onto it is more complex than connecting to an Ethernet. Here the ring network is of really just a collection of point-to-point links. The interface at each computer passes the light pulse stream through to the next link and also serves as T junction to allow the computer to send and accept messages. There are two types of interfaces used.

A passive interface consists of two taps fused onto the main fiber. One tap has an LED or laser diode at the end of it (for transmitting) and the other has a photodiode (for receiving). The tap itself is completely passive and is thus extremely reliable since a broken LED or photodiode does not break the ring.

Another one is the active repeater, where the incoming light is converted to an electrical signal and retransmitted as light even by strengthening the signal if it is weakened.



A fiber optic with active repeaters

Comparison of Fiber Optics and Copper Wire:

1) Fiber optics handles much higher bandwidth than copper. Thus used in high-end networks.
2) In fiber optics, due to low attenuation, repeaters are needed only about every 30k on long lines, versus about every 5km for copper.
3) Fiber optics are not affected by power surges, electromagnetic interference or power failures and also not affected by corrosive chemicals.
4) Because of being thing and lightweight fiber optics are replacing copper wires. Thus maintain lower installation cost.
5) Fiber optics doesn't leak light and are quite difficult to tap, thus resulting excellent security.

6) When electrons move in wire, they affect one another and are themselves affected by electrons outside the wire. Photons in fiber optics do not affect one another (absence of electric charge) and are not affected by stray photons outside the fiber.

WIRELESS TRANSMISSION

The Electromagnetic Spectrum: When electrons move, they create electromagnetic waves that can propagate through free space. The number of oscillation per second of an electromagnetic wave is called frequency "f" and is measure in a Hertz (Hz). The distance between two consecutive maxima (or minima) is called the wavelength (?). By attaching an antenna of the appropriate size to an electric circuit, the electromagnetic waves can be broadcast efficiently and received by a receiver some distance away. All wireless communication is based on this principle. In vacuum, all electromagnetic waves travel at the speed of light, no matter whatever is their frequency. In copper fiber, the speed slows to about 2/3rd of this value and becomes slightly frequency dependent. The fundamental relation is ?f = c ---- (i)

The radio, microwave, infrared and visible light rays can all be sued for transmitting information by modulating the amplitude, frequency or phase of waves. UV X-rays and gamma rays would be better but they are hard to produce and modulate and are dangerous to living things.

The amount of information that an electromagnetic wave can carry I related to its bandwidth. With current technology, it is possible to encode a few bits per Hz at low frequencies, but more on high frequencies. So a cable with a 500 MHz bandwidth can carry several frequencies. So a cable with a 500 MHz bandwidth can carry several gigabits/sec.

On differentiating equation (i) with respect to ?,

df / d? = - c/?

or, ?f = c ??/ ?2 ------- (ii)

Radio Transmission: Radio waves are easy to generate, can travel long distances and penetrate building easily, so they are widely used for the communication. Radio waves are omni directional i.e. they travel in all directions from the source, so that the transmitter and receiver do not have to be carefully aligned physically. The properties of radio waves are frequently dependent. At low frequencies, radio waves pass through the obstacles well, but its power falls off sharply with distance from the source, roughly a $1/r^3$ in the air. At high frequencies, radio waves tend to travel in straight lines and bounce off obstacles. They are also absorbed and interfere with other devices. In the HF and VHF bands, the ground waves tend to be absorbed by the earth. However, the waves that reach the ionosphere (a layer of charged particles circling the earth at the height of 100 to 500km) are refracted by it and sent back to earth. Under certain atmospheric conditions, the signals may bounce several times.

Microwave Transmission: Above 100 MHz, the waves travel in straight lines and thus can be narrowly focused. Concentrating all the energy into a small beam using a parabolic antenna (disk) gives a much higher signal to noise ratio but the transmitting and receiving antennas must be accurately aligned with each other. In addition, this directionality allows multiple transmitters lined up in a row to communicate with multiple receivers in a row without interference.

But some waves may be refracted of low-lying atmospheric layers and may take slightly longer to arrive than direct waves. The delayed waves may arrive out of phase with the direct wave. This effect is called multipath fading and is often serious problem. It is weather and frequency dependent. Microwave communication is so widely used for long-distance telephone communication, cellular telephones and television distributors and also inexpensive. For short distances, it is used in cordless telephones, garage door openers, wireless hi-fi speakers, security gates, etc. its higher bandwidths require more expensive gadget and are subjected to interfere from microwave ovens and radar installations.

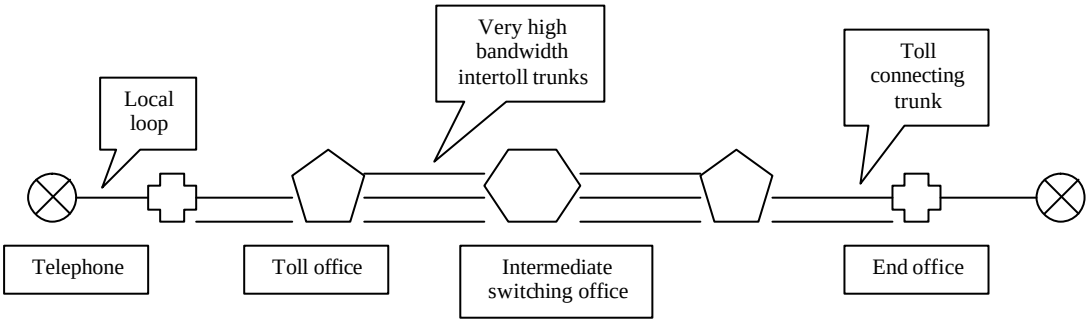
TELEPHONE SYSTEM:

The telephone system consists of three major components:

1. Local Loops (twisted pairs, analog signaling)
2. Trunks (fiber optics or microwave, mostly digital)
3. Switching offices

The two-wire connection between each subscriber's telephone and the nearest end office is known as the **local loop**. By this time, analog signal are being replaced by the digital ones and so the long-distance trunks are now largely digital in the more advanced countries, but the local loops a re still analog and are likely to remain even in near future due to the enormous cost of conversion. Thus when a computer sends digital data over a dial-up line, the data must first be converted to analog form by a modem for transmission over the long-haul trunks, then back to analog over the local loop receiving end and finally back to digital by another modem for storage in the destination computer. Though it is possible to have leased lines form start to end, but these are expensive and are only useful for building intracompany private networks.

Each telephone has two copper wires coming out of it that go directly to the telephone company's nearest **end office** (**local call office)**. The distance is typically 1 to 10 km. Each end office has a number of outgoing lines to one or more nearby switching centers (called **toll offices)**. These liens are called **toll connecting trunks.** Primary, sectional and regional offices form a network by which the toll offices are connected. The toll, primary, sectional and regional exchanges communicate with each other via high-bandwidth **intertoll trunks** (also known as interoffice trunks). The number of different kinds of switching centres and their topology varies from country to country depending on the country's telephone density.



A typical circuit route for a medium-distance call

X.25 and Frame Relay:

X.25 is a first public data network, deployed in 1970s. It is a connection-oriented, introduced at the time during the complete monopoly of telephone service. To use X.25, first a computer used to establish a connection to the remote computer, i.e. placed a telephone call. This connection was given a connection number to be used in data transfer packets. Data packets were very simple, consisting of a 3-byte header and up to 128 bytes of data. The header consisted of a 12-bit connection number, a packet sequence number, an acknowledgement number and a few miscellaneous bits.

By 1980s, new kind of network was introduced, called frame relay. It is a connection-oriented network with no error control and no flow control. It was widely used to interconnecting LANs at multiple company offices.

X.25 defines the procedures for the exchange of data between user devices (DTEs i.e. Data Terminal Equipment) and a packet network node (DCE i.e. Data Circuit Terminating Equipment). In order to define establish the procedures for 2 packet-mode DTEs to communicate with each other through a network, X.25 defines 2 DTE's sessions with their respective DCEs. It is done to provide common procedures between a user DTE and a packet network DCE for establishing a session and exchanging data. The procedure includes functions such as identifying packets of specific user terminals and computers (with logical channel numbers LCN), acknowledging packets, rejecting packets, error control and flow control.

But on the other hand, X.25 does not contain any routing algorithms. The fixed and dynamic packet-routing schemes within a network are left to specific vendor implementations.

Transmission Hazards:

Transmission lines have three major problems: attenuation, delay distortion and noise. Attenuation is the loss of energy as the signal propagates outward. On guided media (example wires and optical fibers), the signal falls off logarithmically with the distance. The loss is expressed in decibels per kilometer. The amount of energy lost depends on the frequency. To see this effect let the signal be considered not only a waveform, but as a series of Fourier components. Each component is attenuated by a different amount, which results in different Fourier spectrum at the receiver, and hence a different signal. In case of an excess attenuation, the receiver may be unable to detect the signal at all, or the signal may fall below the noise level. Such problem can be minimized by placing an amplifier in between. Delay distortion is transmission impairment, caused by the fact that different Fourier components travel at different speeds. For digital data, fast components from one bit may catch up and overtake also components from the bit ahead, mixing the two bits and increasing the probability of incorrect reception. And the third impairment is noise, which is unwanted energy from sources other than the transmitter. "Thermal noise" is caused by the random motion of the electrons in the wire and is unavoidable. "Cross talk" is caused by inductive coupling between two wires that are close to each other. And the "impulse noise" is caused by spikes on the power line or other causes. For digital data, impulse noise can wipe out one or more bits.

Modem: It is a device that accepts a serial stream of bits as input and produces a modulated carrier as output (or vice versa). Modem (modulator-demodulator) is inserted between the (digital) computer and the (analog) telephone system.

Since both the attenuation and propagation speed are frequency dependent, it is undesirable to have a wide range of frequencies in the signal.

Unfortunately in digital data, square waves have a wide spectrum and thus are subject to strong attenuation and delay distortion. These results base band (DC) signaling unsuitable except at slow speeds and over short distances.

On telephone lines, AC signaling is used to get around the problems associated with DC signaling. Here, a continuous tone in the 1000 to 2000 Hz range called a sine wave carrier is introduced. Its amplitude, frequency or phase can be modulated to transmit information.

In amplitude modulation, two different voltage levels are used to represent 0 and 1 respectively. In frequency modulation, (frequency shifting key) two (or more) different tones are used. In the simplest form, the phase modulation, the carrier wave is systematically shifted 45, 135, 225 or 315 degrees at uniformly spaced intervals, where each phase shift transmits 2 bits of information.

Most advanced modems use a combination of modulation techniques to transmit multiple bits per baud. In order to achieve higher and higher speeds, it is not possible just to increase the sample rate, instead more bit are required per sample (per baud).

In Quadratic Amplitude Modulation (QAM), 9600 bps can be transmitted over a 2400-baud line. It has 16 valid combinations and thus can be used to transmit 4 bits per baud, 16 different combinations of amplitude and phase shift are used.

Trunks and Multiplexing: Financial aspect plays a key role during the development; and the telephone system is not away form it. Since the costing is equally the same to install and maintain a high-bandwidth trunk as low-bandwidth trunk between two switching offices. Consequently, Telephone Company has decided for multiplexing many conversations over a single physical trunk. These multiplexing schemes are categorized in two types: FDM (Frequency Division Multiplexing) and TDM (Time Division Multiplexing). In FDM, the user take turns (in a round robin), each one periodically getting the entire bandwidth for a little burst of time.

Switching:

Generally the telephone system is divided into two parts: outside plant (local loops and trunks) and the inside plant (switches). Switching techniques are of two types: circuit switching and packet switching.

Circuit Switching: The telephone system that seeks out a physical "copper" (including fiber and radio) path all the way from the dialer (sender's telephone) to the receiver's telephone is known

as circuit switching. When a call passes through a switching office, a physical connection is established between the lines on which the call came in an do one of the output lines. Here, the basic idea is once a call has been setup, a dedicated path between both ends exists and will continue to exist until the call is finished. Hence the main property of circuit switching is to setup an end-to-end path before any data can be sent. It should be noted that before data transmission can even begin, the call request signal must propagate all the way to the destination and be acknowledged. But once the setup has been completed, the only delay for data is the propagation time for the electromagnetic signal about 5msec per 1000 km. finally; there won't be any busy signals and no danger of congestion.

Message Switching: Here, no physical copper path is established in advance between sender and receiver. Instead, a block of data to be sent from the sender is stored in the first switching office (i.e. router) and then forwarded later, one hop at a time. Each block is received fully, inspected for errors and then retransmitted. A network using this technique is called a store-and-forward network.

Telegram was the first electromechanical telecommunication system which used message switching. The message was punched on the paper tape off-line at the sending office and then retransmitted over a communication lien to the next office along the way, where it was punched out of the paper tape. With message switching, there is no limit on block size, i.e. routers must have disks to buffer long blocks. It also means that a single block may tie up a router-router line for minutes, rendering message switching useless for interactive traffic.

Packet Switching: Packet switching networks restricts a block size, allowing packets to be buffered in router main memory instead of on disk. By making sure that no user can hold any transmission line for very long time (milliseconds), packets switching networks are suitable for handling interactive traffic. Its main advantage is the first packet of a multipacket message can be forwarded before the second one has fully arrived, reducing delay and improving throughput circuit switched, but never message switched.

ISDN

The conventional circuit switched telephone system was designed for analog voice transmission and is inadequate for modern communication needs. Hence there happened to be considerable user demand for an end-to-end digital service was increased and finally a new fully digital circuit –switched telephone syste was created. It is known as Integrated Services Digital Network, having its primary goal for the integration of voice and non voice services.

ISDN Services: Telephones with multiple buttons for instant call setup to arbitrary telephones anywhere in the world. Another feature is the display of the caller's telephone, name and address while ringing. Its more sophisticated service that it allows the telephone to be connected to a computer, so that the caller's database record is displayed on the screen as the call comes in.

Advanced non voice services are remotely electricity meter reading, on-line medical, burglar and smoke alarms that automatically call the hospital, police and fire department, respectively and give their address to speed up response.

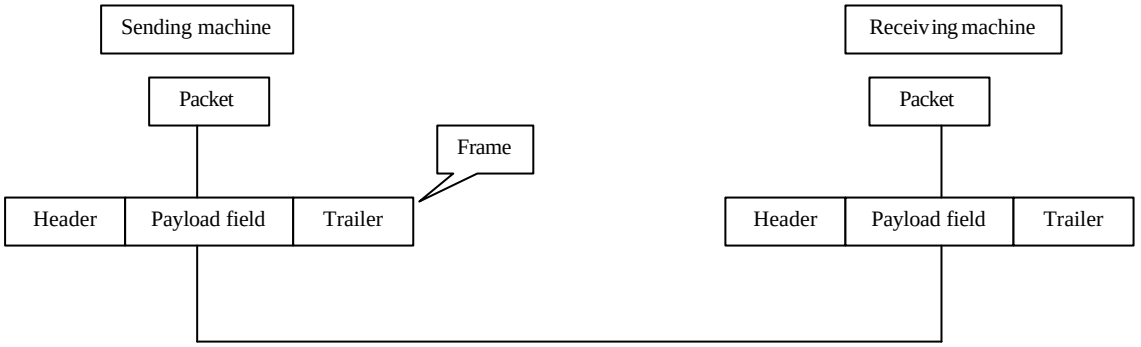
ISDN System Architecture: ISDN system architecture especially refers to the customer's equipment and the interface between the customer and the telephone company. The digital bit pipe is the core idea of ISDN which is a conceptual pipe between the customer and the carrier through which bits flow. Here the bits can flow through the pip in both directions. The digital bit pip can normally does support multiple independent channels by time division multiplexing of the bit stream. Two principal standards for the bit pipe have been developed, a low bandwidth standard for home sue and a higher bandwidth standard for business use that supports multiple channels that are identical to the home channel.

Chapter – 5:

The Data Link Layer

Design: The data link layer has a number of specific functions to carryout. These functions include providing a well-defined service interface to the network layer, determining how the bits of the physical layer are grouped into frames, dealing with transmission errors and regulating the flow of frames so that slow receivers are not swapped by fast senders.

In order to meet those objectives, the data link layer takes the packet from the network layer and encapsulates them into frames for transmission. Each frame contains a frame header, a payload field for holding the packet and a frame trailer.



Relationship between packets and frames

Services Provided to the Network Layer:

The function of the data link layer is to provide services to the network layer. The principal service is transferring data from the network layer on the source machine to the network layer on the destination machine. On the source machine, a process (entity) in the network layer hands same bits to the data link layer for transmission to the destination. Hence the data link transmits the bits to the destination machine, so that they can be handed over the network layer.

The data link layer can be designed to offer various services. The actual services offered can vary from system to system. There can be 3 reasonable possibilities which are as follows:

i) *Unacknowledged connectionless service:* Here, the source machine send independent frames to the destination machine without having the destination machine acknowledge them. No connection is established before hand or released afterward. If a frame is lost due to noise on the line, no attempt is made to recover it in the data link layer. Such service is appropriate when the error rate is very low and so recovery is left to higher layers. It is also suitable for real-time traffic like speech where the late data are worse than bad data. Also most LANs use unacknowledged connectionless service in the data link layer.

ii) *Acknowledged connectionless service:* Here, no connections are used, but each frame sent is individually acknowledged. Thus the sender knows whether or not a frame has arrived safely. This service is useful over unreliable systems, such as wireless systems. Here, the transport layer can always sent a message and wait for it to be acknowledged. If the acknowledgement is not received before the timer goes off, the sender can just send the entire message again. On reliable channels such as fiber, the overhead of a heavyweight data link protocol may be unnecessary, but on wireless channels, it is well worth due to their inherent reliability.

iii) *Acknowledged connection-oriented service:* It is the most sophisticated service the data link layer can provide to the network layer. Each frame sent over the connection is numbered, and the data link layer guarantees that each frame sent is indeed received. Furthermore, it also guarantees that each frame is received exactly in the right order. When connection oriented service is used, transfers have 3 distinct phases. In the first phase, the connections established by having both sides initialize variables and counters that track which frames have been received and which ones have not. In the second phase, one or more frames are actually transmitted. In the

third phase, the connection is released; the variables buffers and other resources used to maintain the connection are freed up (released).

Framing: In order to provide service to the network layer, the data link layer must use the service provided to it by the physical layer. The physical layer simply accepts a raw bit stream and attempt to deliver it to the destination. This bit stream is not guaranteed to be error free. The number of bits received may be less than, equal to or more than the number of bits transmitted, and they have different values. It is up to the data link layer to detect, and if necessary, correct value.

The data link layer breaks the bit stream into discrete frames and computes the checksum for each frame. When a frame arrives at the destination, the checksum is recomputed. If the newly computed checksum happens to be different from the one contained in the frame, the data link layer knows that an error has occurred and takes the necessary steps to deal with it.

Breaking up the bit stream into frames can be done by inserting time gaps between frames. But networks rarely makes nay guarantees about timing, so the gaps might be squeezed out or extra gaps might be inserted during transmission. Thus it is better to look at other methods:

i) *Character count:* Here, field is used in the header to specify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow, and hence where is the end of the frame. Here, the trouble is, the count can be violated by a transmission error. The destination may get out of synchronization and will be unable to locate the start of the next frame. Though the destination knows that the frame is bad, it still has now way to indicate where the next frame starts. Sending a frame back to the source asking for a retransmission does not help either, since the destination does not know how many characters to skip over to get the start of the retransmission. Thus this method is rarely used.

ii) *Starting and ending characters with character stuffing:* In order to overcome the problem of resynchronization by having each frame short the ASCII character sequence DLE (Data Link Escape) STX (Start of Text) and end with the sequence DLE ETX (End of Text). By this method, if the destination loses track of the frame boundaries, the layer just for DLE STX or DLE ETX in the data may easily interfere with the framing. It can be solved in the sender's data link layer by inserting on ASCII DLE character just before each "accidental" DLE character in the data. The data link on the receiving end removes the DLE before the data are given to the network. This technique is called character stuffing. Thus a framing DLE STX or DLE ETX can be distinguished from one in the data by the absence or presence of a single DLE.

a) Data sent by the network layer
DLE STX A DLE B DLE ETX
b) Data after being character stuffed by the data link layer
DLE STX A DLE DLE B DLE ETX
c) Data passed to the network layer on the receiving side
DLE STX A DLE B DLE ETX

A major disadvantage of using this framing method is that it is closely tied to 8-bit characters in general and ASCII character code in particular.

iii) *Starting and ending flags with bit stuffing:* In this method, the data frames contain an arbitrary number of bits and allows character codes with an arbitrary number of bits per character. For instance, say each frame begins and ends with special bit pattern 01111110, called a flag byte. Whenever the sender's data link layer encounters 5 consecutive ones in the data, it automatically stuffs (places) a 0 bit into the outgoing bit stream. This bit stuffing is analogous to character stuffing where DLE is stuffed in the outgoing character stream. When the receiver notices five consecutive incoming 1 bits, followed by a 0 bit, it automatically destuffs (deletes) 0 bit. Thus bit stuffing is completely transparent to the network layer in both computers.

a) 0110111111110010 → original data
b) 0110111110111110010 → data as they appear on the line
c) 0110111111110010 → data as they are stored in the receiver's memory after destuffing

With bit stuffing, the boundary between two frames can be unambiguously recognized by the flag pattern. Thus if the receiver loses track of where it is, it simply scans the input for flag sequences, since they can only occur at frame boundaries and never within the data.

iv) *Physical coding violations:* This method is applicable only to the networks in which a surplus (overflow) encoding on the physical medium is present. For instance, some LANs encode 1bit of data by using 2 physical bits. Normally a 1bit is high-low pair and a 0bit is low-high pair. The combinations high-high and low-low are not used for data, but here every data bit has a transition in the middle, making it easy for the receiver to locate the bit boundaries. This use of invalid physical codes is part of the 802 LAN standards.

Note: Many data link protocols use a combination of a character count with one of the other methods for extra safety. When a frame arrives, the count field is used to locate the end of the frame.

Error Control:

The next design objective is to make sure that all frames are eventually delivered to the network layer at the destination, and in the proper order. If the sender just sends frames disregarding whether they were arriving properly or not. It may be suitable for unacknowledged connectionless service, but not for reliable connection-oriented service. Hence for ensuring reliable delivery, the sender has to provide some feedback about what is happening at the other end of the line. Typically the protocols for the receiver to send back special control frames bear either positive or negative acknowledgements about the incoming frames. If the sender receives a positive acknowledgement about a frame, it is determined that the frame has arrived safely, but if it is negative, the frame must be transmitted again.

Another possibility is that hardware trouble may cause a frame to vanish completely. In such cases, the receiver will not react at all; just waits for the acknowledgement. It can be dealt with by introducing timers into the data link layer. When the sender transmits a frame, it usually also starts a timer. The timer is set to go off after an interval long enough for the frame to reach the destination, be processed and have the acknowledgement propagate back to the sender. Thus if either frame or the acknowledgement is lost, the timer goes off, and the frame has to be transmitted again. But again, the problem is the frames may be transmitted multiple times and the receiver will keep on accepting them. To prevent this, sequence numbers are to be assigned to outgoing frames, so that the receiver can distinguish retransmissions from originals.

Flow Control:

Another objective is to deal with the sender who systematically wants to transmit frames faster than the receiver can accept them. Such case is possible when the sender is running on a fast machine and the receiver on a slow one. The sender keeps pumping the frames out at a high rate unit, whereas the receiver is completely confused. Even if the transmission is error free at a certain point, the receiver will simply not be able to handle the frames as they arrive and will start to lose some.

Hence flow control is to be introduced to throttle the sender into sending frames not faster than the receiver can handle the traffic. The flow control requires some kind of a feedback mechanism, so the sender can be acquainted whether or not the receiver is able to keep up (store and receive).

The protocol contains well-defined rules about when a sender may transmit the next frame. These rules often prohibit frames from being sent until the receiver has granted permission, either implicitly or explicitly.

Error Detection and Correction:

The telephone system has 3 parts: the switches, interoffice trunks and the local loops. The first two are almost digital while the local lops are still analog twisted copper wires due to

excess expense for replacing them. Though errors are rare on the digital part, but still are common on the local loops.

The two basic strategies have been developed to deal with errors. One is tot include enough redundant information along with each block of data sent to enable the receiver to deduce what the transmitted character must have been and the other is to include only enough redundancy to allow the receiver to deduce that an error occurred, but not which error, and have it request a retransmission. The former strategy uses error-correcting codes and latter uses error-detecting codes.

Elementary Data Link Protocols:

The physical layer, data link layer and network layer are independent processes that communicate by passing messages back and forth. When a frame arrives at the receiver, the hardware computes the checksum. If the checksum is incorrect the data link layer is so informed (event=cksum_err), while if the frame arrived is undamaged, the data link layer is also informed (event=frame_arrival) and so the frame can be acquired for more inspection using from_physical_layer. As soon as the receiving data layer acquires an undamaged frame, it checks the control information in the header and if everything is right, the packet portion is passed to the network layer. Under no circumstances, a frame header is even given to a network layer.

Hence in order to keep the network and data link protocols completely separate, the network layer must never be given any part of the frame header. As long as the network layer knows nothing at all about the data link protocol or the frame format, these things can be changed without requiring changes to the network layer's software.

1) **An Unrestricted Simplex Protocol:** It is the most simplest form of protocol, where the data are transmitted only in one direction, both the transmitting and receiving layers are always ready, processing time being ignored and infinite buffer space is available. And finally the communication channel between the data link layer never damages or loses frames. It is also known as "utopia". Thus it has an absence of error and flow control field since on the data field is being used.
2) **A Simple Stop-and-Wait Protocol:** Here, it is similar to an unrestricted simplex protocol; only the additional feature is making it possible for the sender to simply insert a delay in protocol to slow it down, sufficiently to slow it down, and sufficiently to keep from swamping the receiver. For this, the receiver has to provide feedback to the sender. After the packet has passed to its network layer, the receiver sends a little dummy frame (blank) back to the sender, which in effect, gives the sender permission to transmit the next frame. Protocols in which the sender sends one frame and then waits for an acknowledgment before proceeding are called stop-and-wait.
3) **A Simple Protocol for Noisy Channel:** It is related with a communication channel that makes errors. Frames may either be damaged or lost completely. Here, if a damaged frame is transmitted, the receiver will detect this when it computes a checksum. Here, the sender waits for a positive acknowledgment before advancing to the next data item, known as PAR (Positive Acknowledgment with Retransmission) or ARQ (Automatic Repeat reuQuest), also it has the timeout interval to be long enough to prevent premature timeouts.

SLIDING WINDOW PROTOCOLS:

It is related with full duplex data transmission where the same circuit is used for data transmission in both directions. The data frames are intermixed with the acknowledgment frames; and by looking at the type of field in the header of an incoming frame, the receiver can differentiate whether the frame is data or acknowledgment. When a data frame arrives, instead of immediately sending a separate control frame, the receiver restrains itself and waits until the network layer gives the next packet. The acknowledgment is attached to the outgoing frame. This technique of temporarily delaying outgoing acknowledgments so that they can be hooked on to the next outgoing data frame is known as piggybacking.

Piggybacking is a better use of the available channel bandwidth. Piggybacking introduces a communication not present with separate acknowledgments. The acknowledgment field in the

frame header costs only a few bits, whereas a separate frame would need a header, acknowledgment and a checksum.

HDLC:

High-level Data Link Control are bit oriented, an d all use bit stuffing for data transparency. All the bit-oriented protocols use the frame structure as shown:

8	8	8	>0	16	8
01111110	Address	Control	Data	Checksum	01111110

Frame format for bit-oriented protocols

The address field is the main one on lines with multiple terminals, where it is used to identify one of the terminals. The control field is used for sequence number, acknowledgments and other purposes. The data field may contain arbitrary information. It may be long, although the efficiency of the checksum falls off with increasing frame length due to the greater probability of multiple burst errors. The checksum is a minor variation on the well-known cyclic redundancy code, using CRT-CCIT the generator polynomial. It allows lost flag bytes to be detected.

The frame is delimited with another flag sequence (01111110). On idle point-to-point lines, flag sequences are transmitted continuously. The minimum frame contains 3 fields and totals 32 bits, excluding the flags on either end.

The control field uses 3 kinds of frames: Information Supervisory and Unnumbered. The seq is the frame sequence number. The next field is piggybacked acknowledgment. The P/F i.e. Poll/Final field is used when a computer is inviting the terminal to send data. All the frame sent by the terminal, except the final one, have the P/F bit set to P and the final one is set to F. In some of the protocols, the P/F bit is used to force other machine to send a supervisory frame immediately rather than writing for reverse traffic onto which to piggyback the window information.

The various types of supervisory frames are distinguished by the TYPE field.

Type 0 is an acknowledgement frame, used to indicate the next frame to be expected. This frame is used when there is no reverse traffic to use for piggybacking. (RECEIVE READY)

Type 1 is a negative acknowledgement frame (officially called REJECT), which is used to indicate that a transmission error has been detected. The next field indicates the first frame in sequence not received correctly. The sender is required to retransmit all outstanding frames starting at Next.

Type 2 is RECEIVE NOT READY. It acknowledges all frames up to but not including "Next", instead it tells the sender to stop sending. It is intended to signal certain temporary problems with the receiver like shortage of buffers, etc. when the condition has been repaired, the receiver sends a RECEIVE READY, REJECT or certain control frames.

Type 3 is the SELECTIVE REJECT. It asks for retransmission of only the frame specified. It is applicable when the sender's window size is half the sequence space size or less. Hence if a receiver wishes to buffer out sequence frames for potential use, it can force the retransmission of any specific frame using Selective Reject.

Unnumbered frame is used for control purposes but can also be used to carry data when unreliable connectionless service is required. Here, the various bit-oriented protocols differ considerably with other two kinds. 5 bits are available to indicate the frame type, but not all 32 bits.

1	3	1	3	
0	Seq	P/F	Next	Information Frame

1	0	Type	P/F	Next	Supervisory Frame

1	1	Type	P/F	Modifier	Unnumbered Frame

The Data Link Layer in the Internet:

Point-to-point lines play a major role in the Internet, when the millions of individuals connect to the Internet using modems and dial-up telephone lines. The two widely used protocols used in the Internet are SLIP and PPP.

SLIP – Serial Line IP

SLIP is the older protocol actually sued to connect Sun workstations to the Internet over a dial-up line using a modem. Here, the workstation sends raw IP packets over the line with a flag byte (0XC0) at the end for framing. If the flag byte occurs inside the IP packet, a form of character stuffing is sued, and the two byte sequence (0XDB, 0XDC) is sent in its place. Some SLIP attaches a flag byte to both the front and back of each IP packet sent. More recent versions of SLIP do some TCP and IP header compression because the consecutive packets often have many header fields in common. These are compressed by omitting those fields that are the same as the corresponding fields in the previous IP packet.

Demerits:

a) SLIP does not do any error detection or correction, so it is up to higher layers to detect and recover from lost, damaged or merged frames.
b) SLIP supports only IP. With the growth of the Internet to accumulate networks which don't use IP as their native language.
c) Each side must know the other's IP address in advance; neither address can be dynamically assigned during setup. The shortage of IP address is the major issue as it is impossible to give each Internet user a unique IP address.
d) SLIP doesn't provide any form of authentication, so neither party knows whom it is really communicating to (especially in dial-up lines).
e) SLIP is not an approved Internet standard, so various (incomplete) versions exist. Thus the internetworking is not much easier.

PPP – Point-to-Point Protocol

It is an official Internet standard data link protocol for point-to-point lines. PPP handles error detection, supports multiple protocols, allows IP addresses to be negotiated at connection time, permits authentication and has many other improvements over SLIP. PPP provides three major things:

a) A framing method that clearly specifies the end of one frame and the start of the next one. The frame format also handles the error detection.
b) A link control protocol for bringing lines up, testing them, negotiating options and bringing them down properly when they are no longer needed. This protocol is called LCP (Link Control Protocol).
c) Process of negotiating network layer option, independent of the network layer protocol to be used. This method is to have a different NCP (Network Control Protocol) for each network layer supported.

Example: The PC first calls the ISP's router via a modem and telephone line. After the router's modem receives the phone and establishes a physical connection, the PC sends a series of LCP packets to the router in the payload field of one or more PPP frames. These packets, and their responses, select the PPP parameters to be used. Once these have been agreed upon, a series of NCP packets are sent to configure the network layers. Typically, the PC wants to run a TCP/IP protocol stack, so it needs an IP address. Since there is no enough IP address to go around, normally each IP gets a block of them and then dynamically assigns one to each PC for the duration of its login session. If an ISP owns "n" IP address, it can have up to "n" machines logged in simultaneously. Here NCP is used to do the IP address assignment. At this instant, PC is now an Internet host and can send and receive IP packets, just as hardwired hosts can. When the user is finished, NCP is used to breakdown the network layer connection and free up the IP address. Then LCP is used to shutdown the data link layer connection. Finally the computer tells modem to hang up the phone, releasing the physical layer connection.

PPP Frame Format: It is very much similar to HDLC frame format, where only the major difference is that PPP is character oriented while HDLC is bit oriented. PPP, like SLIP uses a character stuffing on dial-up modem lines, so all frames are on integral number of bytes. PPP

frames cannot only be sent over dial-up telephone lines, but also over SONET or true bit-oriented HDLC line (example: for router-router connections).

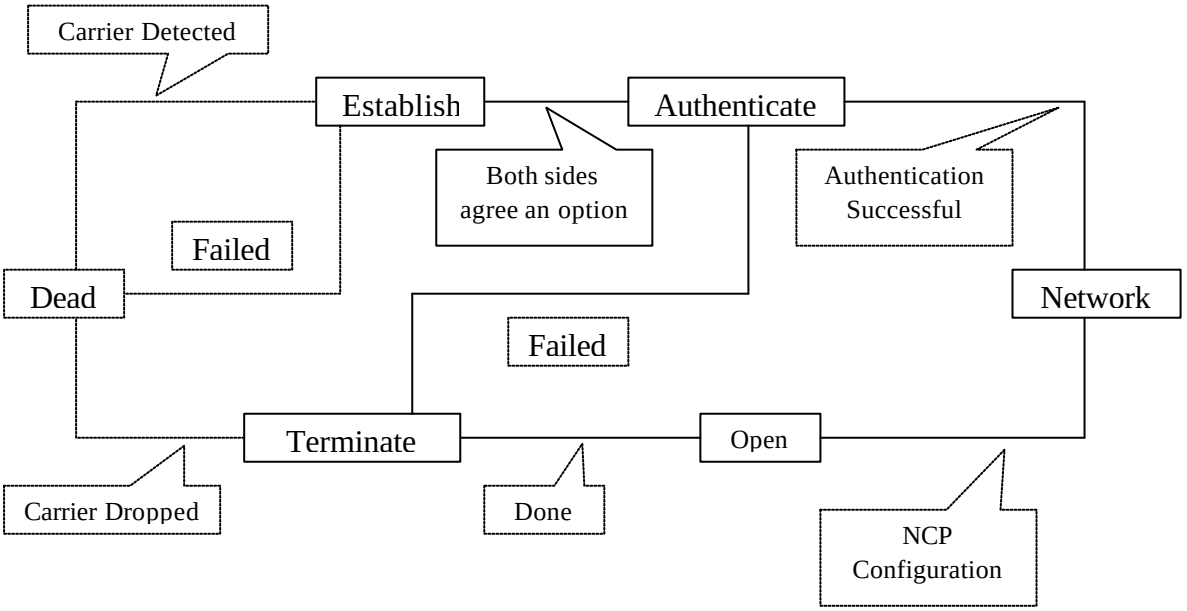
1	1	1	1 or 2	Variable	2 or 4	1	Bytes
Flag 01111110	Address 11111111	Control 00000011	Protocol	Payload	Checksum	Flag 01111110	

All PPP frames begin with the standard HDLC flag byte (01111110), which is character stuffed. Next is the Address field, which is always set to the binary value 11111111 to indicate that all stations are to accept the frame. The Control field's default value is 00000011, which indicates the unnumbered frame.

Since the Address and Control field are always constant in the default configuration, LCP provides the necessary mechanism for the two parties to negotiate. The fourth is Protocol field, whose job is to tell what kind of packet is in the Payload field. Codes are defined for LCP, NCP, IP, IPX, Apple Talk and other protocols. Protocols starting with a O bit are network layer protocols such as IP, IPX, OSI, CLNP, XNS. While those starting with a 1 bit are sued to negotiate other protocols like LCP and NCP. The size of the Protocol field is 2 bytes, but it can be negotiated down to 1 byte using LCP.

The Payload field is variable length. If the length is not negotiated using LCP during line setup, a default length of 1500 bytes is used. After the Payload is the Checksum field, which is normally of 2 bytes, but a 4 bytes checksum can be negotiated.

In summary, PPP is a multiprotocol framing mechanism suitable for use over modems, HDLC bit-serial lines, SONET and other physical layers. It supports error detection, option negotiation, header compression and reliable transmission using HDLC framing.



When the line is DEAD, no physical layer carrier is present and no physical layer connection exists. As the physical connection is established, the line moves to ESTABLISHED. Then LCP begins to negotiate, which if successful, leads to AUTHENTICATE. Now the two parties can check on each other's identities, if desired. When the NETWORK phase is entered, the appropriate NCP protocol is invoked to configure the network layer. If the configuration is successful, OPEN is reached and data transport can take place. When data transport is finished, the line moves to TERMINATE phase and from there, back to DEAD when the carrier is dropped.

Medium Access Sub layer

Networks can be divided into two categories: those using point-to-point connections and those using broadcast channels. In any broadcast network, the key issue is how to determine who gets to use the channel when there is competition for it. Broadcast channels are also known as multi-access channels or random access channel.

The protocols used to determine who goes next on a multi-access channel belong to a sublayer of the data link layer called the MAC (Medium Access Control) sublayer. MAC is especially important in LANs, where mostly multi-access channel is the basics of their communication. WANs, in contrast use point-to-point links except for satellite networks.

1. Static Channel Allocation in LANs and MANs: The traditional way of allocating a single channel is Frequency Division Multiplexing (FDM). If there are N users, the bandwidth is divided into N equal sized positions, where each bandwidth is divided into N equal sized positions, where each user is assigned one portion. Since each user has a private frequency, there is no interference between the users. When there is a small and fixed number of users, FDM is a simple and efficient allocation mechanism.

In case of large number of users, the spectrum in cut up into N regions and if only few of them are interested in communicating, a large piece of valuable spectrum will be wasted. And even if more than N users are interested, some of them will be denied permission due to lack of bandwidth and even if they acquire, they can hardly acquire or receive anything.

2. Dynamic Channel Allocation in LANs and MANs:

Its five main fields of assumptions are:

i) *Static Model:* The model consists of N independent stations (computers, telephones, etc), each with a program or user that generates frames for transmission. Once a frame has been generated, the station is blocked and does nothing until the frame has been successfully transmitted.

ii) *Single Channel Assumption:* A single channel is available for all communications. All stations can transmit on it and can receive from it.

iii) *Collision Assumption:* If two frames are transmitted simultaneously, they overlap in time and the resulting signal is garbled. This event is called collision. A collided frame must be transmitted later.

iv) *a. Continuous timer:* Frame transmission can begin at any instant. There is no master clock dividing time into discrete intervals.
b. Slotted time: Time is divided into discrete intervals (slots). Frame transmission always begins at the start of a slot. A slot may contain 0, 1 or more frames referring to an idle slot, a successful transmission or a collision, respectively.

v) *a. Carrier sense:* Station can predict if the channel is in use before trying to use it. If the channel is sensed busy, no station will attempt to use it until it goes idle.
b. No carrier sense: Stations can't sense the channel before trying to use it. They just go ahead and transmit. Only later they can know the result.

MULTIPLE ACCESS PROTOCOLS

ALOHA:

This approach was introduced by Norman Abramson in 1970s at the University of Hawaii; a technique for uncoordinated users to effectively compete for a channel. It is a new method to solve the channel allocation problem. It is classified into 2 forms as per the time division.

Pure ALOHA:

Here, the users transmit data whenever they need to send. During the collisions, the frames will be damaged. But due to the feedback property of broadcasting, a sender can know whether the frame is destroyed or not. If the frame is destroyed, the user waits for the random period of time and sends it again. The waiting time must be random; otherwise the frames will keep on colliding. Systems in which multiple users share a common channel that lead to conflicts are known as *contention systems*. Generally the frames in the ALOHA system are of same size.

Slotted ALOHA:

In this approach, the time is divided into discrete intervals, each interval corresponding to one frame. Its basic characteristics are as follows:

- All frames consist of exactly N bits.

- Time is divided into N/L seconds (i.e. a slot equals the time to transmit one frame)
- Stations start transmitting only at the start of slots.
- Stations are synchronized so that each node knows when the slots begin
- If two or more frames collide in a slot, then all the stations detect the collision even before the slot ends.

Carrier Sense Multiple Access Protocol

Protocols in which stations listen for a carrier and act accordingly are called carrier sense protocols.

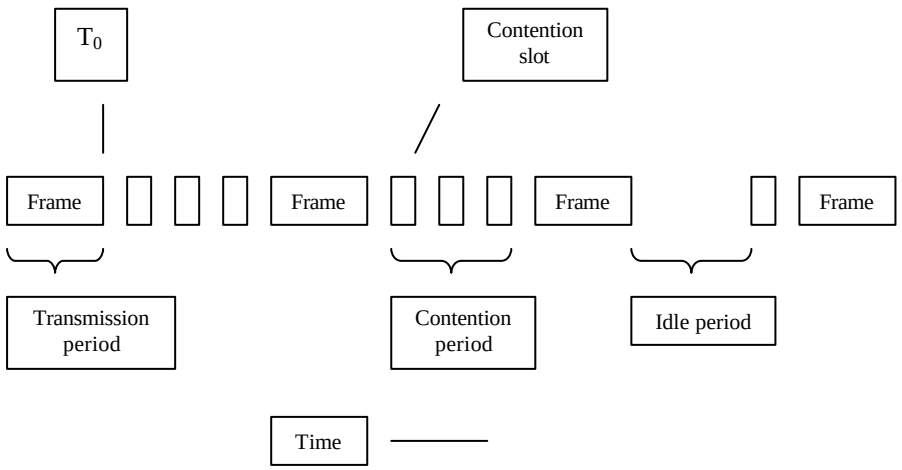
Persistent and Non Persistent CSMA

1-Persistent CSMA: Here, when a station has to send data, it first listens to the channel to see if anyone else is transmitting at that moment. If the channel is busy, the station waits until it becomes idle. When the station detects an idle channel, it transmits a frame. If a collision occurs, the station waits random amount of time and starts all over again. The protocol is called 1-persistent because the station transmits with a probability of 1 whenever if finds the channel idle.

Non-Persistent CSMA: Here, before sending, a station senses the channel. If no one else is sending, the station begins doing so itself. But if the channel is already in use, the station does not continually sense it in order to sense immediately upon detecting the end of the previous transmission. Instead, it waits a random period of time and then starts all over again.

P-Persistent CSMA: It is applicable to slotted channels. When a station becomes ready to send, it senses the channel. If it is idle, it transmits with a probability P. With a probability q = 1-p, it defers until the next slot. If that slot is also idle, it either transmits or defers again with probabilities p and q. This process is repeated unless either the frame has been transmitted or another station has begun transmitting.

CSMA with Collision Detection: Persistent and non persistent CSMA protocols ensure that no stations begin to transmit when it senses the channel busy. But if two stations sense the channel to be idle and begin transmitting simultaneously, they will both detect the collision almost immediately. The transmission should be stopped as soon as the collision is detected before the frame transmission is completed. Quickly terminating damaged frames saves time and bandwidth. This protocol is known as CSMA/CD (Carrier Sense Multiple Access with Collision Detection), which is widely used on LANs in the MAC sublayer. The conceptual model of CSMA/CD is given below:



At the point marked t_0, a station has finished transmitting a frame. After that any station will try to send a frame. If two or more stations try to transmit simultaneously, there will be a collision. Collisions can be detected by looking at the power or pulse width of the received signal

and comparing it to the transmitted signal. After a station detects a collision, it aborts its transmission, waits a random period of time, and then tries again, assuming that no other station has started transmitting at that time. Hence, CSMA/CD consists of alternating contention and transmission periods, with idle periods occurring when all stations are quiet.

It should also be noted that a sending station must continuously monitor the channel, listening for noise bursts that might indicate a collision. For this reason, CSMA/CD with a single channel is inherently a half-duplex system. It is impossible for a station to transmit and receive frames at the same time because the receiver is looking for collisions during every transmission.

Collision Free Protocols: Although collisions do not occur with CSMA/CD, once a station has seized the channel, but they can still occur during the contention period. The collisions adversely affect the system performance, especially when the cable is long and the frames are short. Moreover, very long, high bandwidth fiber optic networks use the combination of large and short frames. Thus protocols which resolve the contention for the channel without any collisions at all are called collision-free protocols.

Wireless LAN Protocols: A system of portable computers that communicate by radio can be regarded as a wireless LAN. These have different properties than conventional LAN and require special MAC sublayer protocols. A common configuration for a wireless LAN is an office building with base stations strategically placed around the building. All the base stations are wired together using copper or fiber. If the transmission power of the base stations and portables is adjusted to have a range of 3 or 4 meters, then each room becomes a single cell, and the entire building becomes a large cellular system. But when a receiver is within range of two active transmitters, the resulting signal will generally be garbled and useless.

Chapter – 6:

The TCP/IP Reference Model

ARPANET was a research network sponsored by US Department of Defense. It eventually connected hundreds of universities and government installations through leased telephone lines. When satellite and radio networks were added later, the existing protocols had trouble internetworking with them. So new reference architecture was needed, which would be able to connect multiple networks together in a seamless way. This was known as TCP/IP Reference Model.

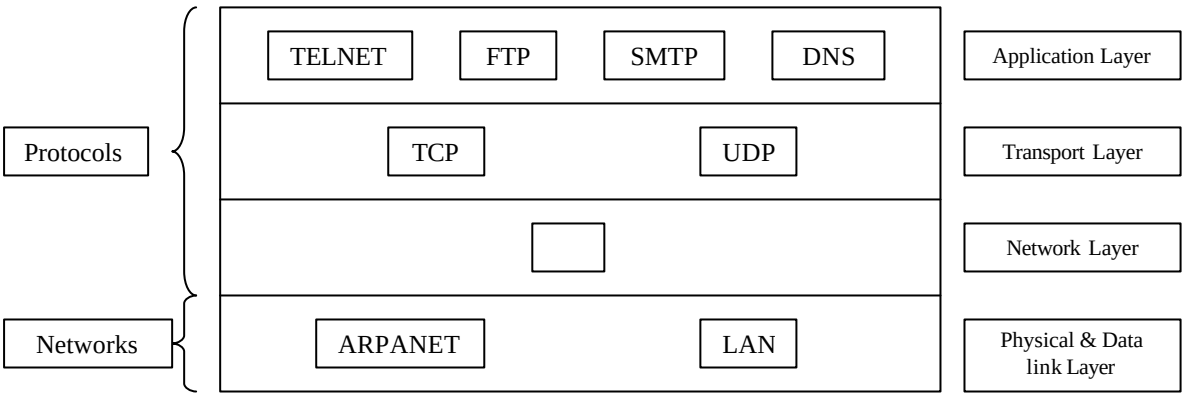
1. **The Internet Layer:** The Internet Layer holds the whole architecture together. It is a packet-switching network based on a connectionless inter network layer. Its function is to permit hosts to inject packets into nay network and let them to travel independently to their destinations. The Internet Layer defines an official packet format and protocol called Internet Protocol (IP). Thus this layer's function is to deliver IP packets to their proper destination.

2. **The Transport Layer:** The Transport Layer allows peer entities on the source and destination hosts to carry on a conversation. Here, two end-to-end protocols have been defined. The first one TCP (Transport Control Protocol) is a reliable connection-oriented protocol that allows a byte stream originating on one machine to be delivered on any other machine in the internet without error. It fragments the incoming byte stream into discrete messages and passes each one on to the Internet Layer. At the destination, the receiving TCP/IP process reassembles the received messages into the output stream. TCP/IP also handles the flow control.

   The second is UDP (User Datagram Protocol), an unreliable connectionless protocol for applications that do not want TCP'S sequencing or flow control. It is widely used for one-shot, client-server type request-reply queries and applications in which prompt delivery is important than accurate delivery like transmitting speech or voice.

3. **The Application Layer:** The TCP/IP model does not have session or presentation layers. The Application layer is directly on top of transport layer. It contains all the higher-level protocols. The early one included virtual terminal (Telnet), file transfer (FTP) and electronic mail (SMTIP). Telnet protocol allows a user on one machine to log into a distant machine and work there. FTP provides a way to move data efficiently from one machine to another. Email was originally just a kind of file transfer, but later, a specialized protocol was developed for it. There some other new protocols like Domain Name Service for mapping host names to their network address or vice-versa, NNTP for moving news articles around and HTTP for fetching pages on the World Wide Web, etc.

4. **The Host-to-Network Layer:** The TCP/IP reference model does not really say much about what happen below the Internet Layer, except to port out that the host has to connect to the network using same protocol so it can send IP packets over it. This protocol is not defined and varies from host to host and network to network.



Protocols and networks in TCP/IP model

A comparison of OSI and TCP Reference Models:

The OSI and TCP/IP reference models have much in common. Both are based on the concept of stack independent protocols, also the layers are functionally similar. Say, in both models,, the layers up through and including the transport layer provide and end0to-end network independent transport service to processes wishing to communicate. These layers form the transport provider. The three major central concepts to OSI models are:

a) services
b) interfaces
c) protocols

The **service** definition tells what the layer does, not how entities above it access it or how the layer works inside.

A layer's **interface** tells the processes above it how to access it. It specifies what the parameter are and at results to expect. It is not concerned about how the layer works inside.

The **protocols** used in a layer are for their own function. The layer can use nay protocols it wants to, as long as it gets the job done. It can also exchange them without affecting software in higher layers.

But in case of TCP/IP model, it did not originally clearly distinguish between service, interface and protocol. The only real service s offered by the Internet layer are SEND IP PACKET and RECEIVE IP PACKET.

Hence, the protocols in OSI model are better hidden than in the TCP/IP model and can be replaced relatively easily as the technology changes.

The OSI reference model was introduced before the protocols were invented. Thus the model was not focused toward one particular set of protocols. Thus a s demerit, the designers did not have much experience with the subject and did not have good idea of which functionality to put in which layer. For instance, the data link layer originally dealt only with point0to-porrint networks. When broadcast networks were introduced, a new sub layer had to be hacked into the model. While in case of TCP/IP, the protocols came first and the model was really just a description of the existing protocols. There was no problem with the protocol filling the model. They fit perfectly. The only trouble was that the model did not fit any other protocol stacks.

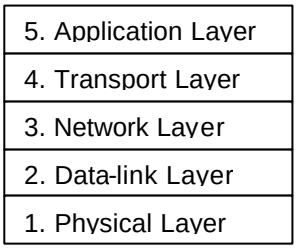
Also an obvious difference between the two models in the number of layer is: the OSI model had seven layers and the TCP/IP has four layers. Both have different (inter) network, transport and application layers. And the final difference is of being connectionless versus connection-oriented communication. The OSI model supports both connectionless and connection-oriented communication in the network layer, but only connection oriented communication in the transport layer. The TCP/IP model has only one mode in the network layer but supports both modes in the transport layer, giving user a choice.

Demerits of OSI Model and Protocols:

1. Bad timing
2. Bad technology
3. Bad implementations
4. Bad politics

Demerits of TCP/IP Reference Model:

a) The TCP/OP model doesn't clearly distinguish the concepts of service, interface and protocol. Good software engineering requires differentiating between the speciation and the implement, something that OSI does very carefully, not by TCP/IP. Hence TCP/IP model is not much of guide for designing new networks using new technologies.
b) TCP/IP model poorly describes any protocol stack.
c) Host-to-network layer is not really a layer; the term is used in the context of layer protocols. It is simply an interface between the network and data link layers.
d) TCP/IP model doesn't distinguish (or even mention) the physical and data link layers, which are totally different.
e) TCP/IP protocols though well implemented, but generally produced by a couple of graduate students hack away until they got tired. The protocol implementations were then distributed free, resulting widely spread and thus hard to replace.



The hybrid reference model

At the network layer, the Internet can be viewed as a collection of subnetworks or Autonomous Systems (ASes) that are connected together. There is no real structure, but several backbones exist which are constructed from high-bandwidth lines and fast routers. Attached to the backbones are regional (mid level) networks, and attached to them are the LANs at many universities, companies and ISPs.

The component that holds the Internet together is the network layer protocol, IP (Internet Protocol). Its job is to provide a best-efforts way to transport datagram from source to destination, without regard to whether or not these machines are on the same network.

The true process of communication in the Internet is the transport layer takes the data streams and breaks up into fragments. Each datagram is transmitted through the Internet, possibly being fragmented into smaller units as it goes. When all the pieces finally get to the destination machine, they are reassembled by the network layer into the original datagram. This datagram is then handled to the transport layer, which inserts it into the receiving process input stream.

a) The IPv4 Protocol:

32 Bits

Version	IHL	Type of Service		Total Length
Identification			DF MF	Fragment Offset
Time To Live	Protocol	Header Checksum		
Source Address				
Destination Address				
Options (0 or more words)				

An IP datagram consists of a header part and a text part. The header has a 20-byte fixed part and a variable length optimal part.

The version field keeps track of which version of the protocol the datagram belongs to.

Since the header length is not constant, a field in the header IHL, is provided to tell how long the header is, in 32-bit words. The minimum value is 5, which applies when no options are present.

The type of service field allows the host to tell subnet what kind of service it wants. There are various combinations of reliability and speed is possible. Like for digitized voice, fast delivery is required while for file transfer, error-free transmission is more important than fast transmission.

The total length includes everything in the datagram – both header and data. The maximum length is 65,535 bytes.

The identification field is needed to allow the destination host to determine a newly arrived fragment belongs to which datagram. All the fragments of a datagram contain the same identification value.

Then there is unused bit, followed by two 1-nit fields. DF stands for Don't Fragment. It is an order to the routers not to fragment the datagram because the destination is incapable of putting the pieces back together again.

MF stands for More Fragments. All fragments except the last one have this bit set so that when all fragments of a datagram have arrived.

The fragment offset indicates the location of the fragment in the current datagram. All fragments except the last are in a datagram must be a multiple of 8 bytes, the elementary fragment unit. Since 13 bits are provided, there is a maximum of 8192 fragments per datagram i.e. a maximum datagram length of 65536 bytes, one more than the total field length.

The time to live field is a counter used to limit packet lifetimes. It counts time in seconds, allowing a maximum lifetime of 255 seconds. It must be decremented in each hop and also multiple times when queued for a long time in a router. When it hits zero, the packet is discarded and a warning packet is sent back to the source host.

The protocol field tells which transport process has to be followed. For instance, TCP and UDP.

The header checksum verifies header only. These are useful for detecting errors generated by bad memory words inside a router. The header checksum is recomputed at each hop, because at least one field always changes.

The source address and destination address indicate the network number and host number.

The options field was designed in order to avoid allowing subsequent versions of the protocol to include information not present in the original design permitting experiments to try out new ideas. The options are variable length.

b) IP Addresses:

Every host and router on the Internet has an IP address, which encodes its network number and host number. Since the combination is unique, no two host have the same IP address. All IP addresses are 32-bits long and are used in the source address and destination address fields of IP packets. Those machines connected to multiple networks have a different IP address on each network.

Class	32 bits				Range
A	0	Network	Host		1.0.0.0 to 127.255.255.255
B	10	Network	Host		128.0.0.0 to 191.255.255.255
C	110	Network		Host	192.0.0.0 to 223.255.255.255
D	1110	Multicast Address			224.0.0.0 to 239.255.255.255
E	11110	Reserved for future use			240.0.0.0 to 247.255.255.255

The class A, B, C and D formats allow for up to 126 networks with 16 million hosts each, 16,382 networks with up to 64k hosts, 2 million networks with up to 254 hosts each and multicast in which a datagram is directed to multiple hosts. Network numbers are assigned by NIC (Network Information Center) to avoid conflicts.

Network addresses are usually written in dotted decimal notation. In this format, each of the 4 byte is written in decimal from 0 to 255. The lowest IP address is 0.0.0.0 and the highest is 255.255.255.255.

c) Subnets:

Since all the hosts in a network must have the same network number, this can cause problems as networks grow. For its solution, a network is allowed to split into several parts for internal use but still at like a single network to the outside world. These parts are known as subnets.

Subnet mask indicates the split between network + subnet number and host. Subnet masks are also written in dotted decimal notation, with the addition of a slash followed by the number of bits in the network + subnet part. Let there be a company having a class B address that could start out numbering the hosts from 1 to 254. When the second LAN arrives, the 16-bit host number is split into a 6-bit subnet number and a 10-bit host number. This split allows 62 LANs, each with up to 1022 hosts. This is due to the fact that a class B IP address has 16 – bit network number and 16 – bit host number. When 6 – bit of the host number is borrowed to network number, the 6 – bit number is said to the subnet number. The new total network number happens to be 16 + 6 = 22. Then the subnet mask is calculated as 11111111.11111111.11111100.00000000 i.e. 255.255.252.0/22 where /22 indicates that the subnet mask is 22 bits long.

32 Bits			
10	Network	Subnet	Host
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		1 1 1 1 1 1	0 0 0 0 0 0 0 0 0 0

Class B network subnetted into 64 subnets

Outside the network, the subnetting is not visible, so allocating a new subnet does not require contacting NIC or changing any external databases.

To observe how subnets work, it is necessary to know how IP packets are processed at a router. Each router has a table listing some number of IP addresses which are of two types. The first kind tells how to get distant networks and the second kind tells how to get to local hosts. The network interface is associated with each table to reach destination and other reasons.

When a IP packet arrives, its destination address is looked up in the routing table. If the packet is for a distant network, it is forwarded to the next router on the interface given in the table. If it is a local host, it is sent directly to the destination. But if the network is not present, the packet is forwarded to the default router with more extensive tables. Hence the router on the subnet knows how to get to all other subnets and also how to get to all the hosts on subnet. But it does not require knowing the details about hosts on other subnets.

d) Internet Control Protocols:

ICMP: When some error occurs in the Internet, the event is reported by the ICMP (Internet Control Message Protocol). Each ICMP message type is encapsulated in an IP packet.

Message Type	**Description**
Destination Unreachable	Packet could not be delivered
Time Exceed	Time to live field hit 0
Parameter Problem	Invalid header field
Source Quench	Choke packet (Slow down the packet flow)
Redirect	Teach a router about geography (wrong routing)
Echo Request	Ask a machine if it is alive
Echo Reply	Yes, I am alive (reachable)
Time Stamp Request	Same as Echo request, but with timestamp
Time Stamp Reply	Same as Echo reply, but with timestamp

ARP: Address Resolution Protocol is the protocol for querying the identification of host for the given IP address in the network (either the Internet or the Ethernet).

Here, when a sender has to send a data, its IP software builds an Ethernet frame addressed to the receiver and then puts the IP packet in the payload field and dumps it onto the Ethernet. The Ethernet board of the receiver detects this frame, recognizes it as a frame for itself. The Ethernet driver extracts the IP packet from the payload and passes it to IP software, which sees that it is correctly addressed and processes it.

In many cases, receiver needs to send back a reply, forcing it to run ARP to determine sender's Ethernet address. This ARP broadcast can be avoided by including tan IP address of the sender to Ethernet mapping in the ARP packet.

When a packet has to be delivered to the host in another network, there are two types of solutions. First, the router could be configured to ARP requests for the network connecting t o the destination host. In this case, the sender will make an ARP cache entry the destination IP address and send all traffic for the receiver to the local router. This solution is called proxy ARP. The second solution is have sender immediately see that the destination is on a remote network and just send al such traffic to a default Ethernet address that handles all remote traffic. Here, the router is not required to know which remote network is serving.

RARP: ARP returns the hardware address to a given IP address, while RARP returns the IP address for a specific hardware address. Reverse Address Control Protocol allows a newly-booted diskless workstation to broadcast its 48-bit Ethernet address, which is recorded by the RARP server that looks up the Ethernet address in its configuration fields and sends back the corresponding IP address.

In diskless workstation, when they are booted up, they get a binary image of its operating system from a remote file server thus they know only an Ethernet address but not IP address.

CIDR – Classless Inter Domain Routing:

IP has been in heavy use for over a decade. But unfortunately, it is running out of addresses. For its solution, CIDR was put forward. The basic idea of CIDR is to allocate the remaining class C networks of which there are almost two million, in variable-sized blocks. If an ISP needs say 2000 addresses, it is given a block of 2048 address.

In addition to use blocks of contiguous class C networks as units, the allocation rules for the class C address were also changed. The world was partitioned into four zones, and each one given a portion of class C address space.

Addresses 194.0.0.0 to 195.255.255.255 are for Europe
Addresses 198.0.0.0 to 199.255.255.255 are for North America
Addresses 200.0.0.0 to 201.255.255.255 are for Central and South America
Addresses 202.0.0.0 to 203.255.255.255 are for Asia and the Pacific

In this way, each region was given about 32 million addresses to allocate. While another 320 million class C addresses from 204.0.0.0 to 223.255.255.255 are held for the future.

IPv6:

Though CIDR may prove to be prompt solution at the instant, but the Internet is growing exponentially, being applicable almost all the parts of the daily life. Hence new version of IP is required that can fulfill the following objectives:

i) Support billions of hosts, even efficient with inefficient address space allocation.
ii) Reduce the size of the routing tables.
iii) Simplify the protocol, to allow routers to process packets faster.
iv) Provide better security.
v) More concentrated towards the type of service, especially real-time data.
vi) Aid multicasting by allowing scopes to be specified.
vii) Make it possible for a host to mobile without changing its address.
viii) Allow the protocol to evolve in future.
ix) Permit the old and new protocols to coexist for years.

And finally, a new version was invented called SIPP (Simple Internet Protocol Plus) and its designation was given as IPv6. Though IPv6 is not compatible with present IPv4, but it is

compatible with all other Internet protocols including TCP, UDP, ICMP, IGMP, OSPF, BGP & DNS. Its characteristics are as follows:

i) It has longer addresses than IPv4 i.e. of 16 bytes. Thus it can effectively provide unlimited supply of Internet addresses.

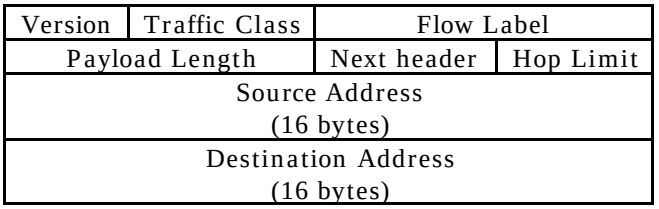
ii) It has simplified header containing only 7 fields (3 in IPv4). This results routers to process packets faster.

iii) Provides better support for options. Since the fields that previously were required are now optional, making simple for routers to skip for options not intended for them.

iv) It has improved security. Authentication and privacy are its key features.

v) More attention has been given in the type of service with a view to cope with multimedia traffic growth.

IPv6 Header



The version field is of 6 bytes and keeps track of which version of the protocol the datagram belongs to.

The priority field is sued to distinguish between packets whether their sources can be flow controlled or not. The distinction allows router to deal with packets better in the event of congestion. The priorities of packets are in ascending order, such as 1 for news, 4 for FTP and 6 for telnet.

Though flow label is not used practically, it might be used to allow a source and destination to set up an exceptional connection with particular properties and requirements.

The payload length field tells how many bytes the packet acquires.

The next header field is placed so that there can be additional extension headers.

The hop limit field is used to keep packets alive.

The source and destination address fields are of 16 bytes, so that they never run out.

Chapter -7

Network Layer and Internet Layer

The network layer is concerned with getting packets from the source all the way to the destination. There may require many hops at intermediate routers along the way while getting to the destination. The network layer is the lowest layer that deals with end-to-end transmission.

In order to get its objective, the network layer must know about the topology of the communication subnet (i.e. the set of all routers) and choose appropriate paths through it. It must also look forward to choose routes in order to avoid overloading of some communication, lines and routers while leaving others idle. Finally, when the source and destination are in different networks, it is up to the network layer to deal with these differences and solve the problem.

Design Issues of Network Layer:

a) Service Provided to the Transport Layer: The network layer provides services to the transport layer at the network layer/transport layer interface. Its important characteristic is that it frequently is the interface between the carrier and the customer i.e. the boundary of the subnet. The carrier often has control of the protocols of the protocols and interfaces up to and including the network layer. Its job is to deliver packets given to it by customers. It is designed with a view to attain the following objectives:

i) The services should be independent of the subnet technology.
ii) The transport layer should be shielded from the number, type and topology of the subnets present.
iii) The network address provided to the transport layer should use a uniform numbering plan, even across LANs and WANs.

But there are two factions in the design process i.e. whether the network layer should provide connection oriented or connectionless service.

Connectionless Service: It refers to the Internet Community. According to them, the subnet's function is simply moving bits around and is unreliable, no matter how it is designed. Thus the hosts should accept the fact that it is unreliable and do error control and flow control themselves. Thus network service should be connectionless, with primitives SEND PACKET and RECEIVE PACKET and each packet must carry the full destination packet.

Connection-oriented Service: It refers to the telephone companies which say the subnet should provide a reliable, connection-oriented service with following properties:

* Before sending data, a network layer on the sending side must setup a connection to its peer on the receiving side. For it, a special identifier is given that is to be used until all the data have been sent.
* When a connection is setup, the two processes can negotiate about the parameters, quality of cost of service to be provided.
* Communication is in both directions and packets are delivered in sequence.
* Flow control is to be provided to prevent a fast sender form dumping packets into the pipe at a higher rate than the receiver can accept, thus leading to overflow.

In terms of complexity, it is in the network layer (subnet) in connection-oriented service, while it is connectionless service it is in the transport layer (hosts). Supporters of connectionless service say that the computing power has become cheap, so there is no reason to put the complexity in the subnet. While the connection-oriented users argue that most users are not interested in running complex transport layer protocols in their machines. They simply want reliable and trouble-free service which the network layer can provide.

b) Internal Organization of the Network Layer:

There are basically two different schemes for organizing the subnet; one using connections and the other working connectionless. In the context of the internal operation of the subnet, a connection is usually called a virtual circuit (similar telephone system). On the other hand, the independent packets of the connectionless organization are called datagram (similar to telegrams).

The key objective of virtual circuit is to avoid choosing a new route for every packet or cell sent. Instead, when a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup, which is used for all traffic flowing over the connection. And when the connection is released, the virtual circuit is also terminated.

In contrast, with a datagram subnet, no routes are worked out in advance, even if the service is connection-oriented. Each packet sent is rerouted independent of its predecessors. Successive packets may follow different routes.

Since each router must remember where to forward packets for area of the currently open virtual circuits passing through it, hence each packet traveling thought the subnet mush contain a virtual circuit number field in its header, along with sequence numbers, checksums, etc. So that when a packet arrives at the route, the router knows which line it arrived and only can forward on the correct output line.

While each data must contain the full destination address. For a large network, these addresses can be quite long. When a packet comes in, the router looks up the outgoing line to use and sends the packet on its way. Also, the establishment and release of network or transport layer connections do not acquire any special work on the part of the routers.

The Optimality Principle: It refers to a general statement about optimal routes without regard to network topology or traffic. It states that *if router J is on the optimal path from router I to router K, then the optimal path from J to K also fall along the same route.*

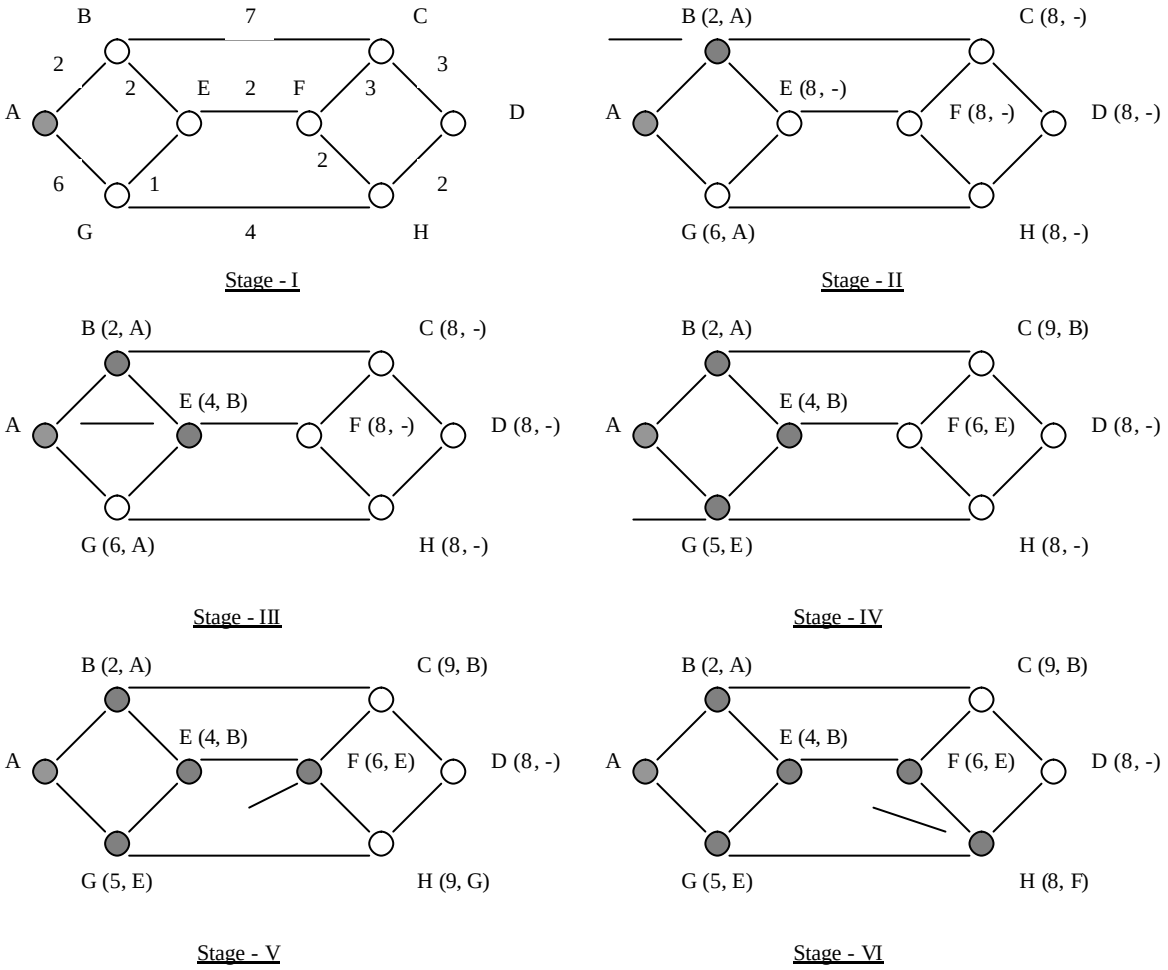
Consequently, the set of optimal routes from all sources to a given destination form a tree rooted at the destination, which is known as a sink tree. The goal of all routing algorithms is to discover and use the sink trees for all routers. A sink tree does not contain any loops; each packet will be delivered within a finite and bounded number of hops.

Shortest Path Routing:

The very first thing to be done is to build a graph of the subnet, with each node of graph representing a router and an arch representing a communication line. To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph. The shortest path can be defined in three ways: counting the number of hops, measuring geographic distance and the transmission delay between two nodes.

One of the shortest path routing algorithms was deduced by Dijkstra in 1959m where each node is labeled with its distance from the source node along the best known path. Initially, no paths are known and hence all nodes are labeled with infinity. As the algorithm proceeds and paths are found, the labels are changed, resulting better paths. A label is either tentative or permanent. At first, all labels are tentative; but when it is discovered that a label represents the shortest possible path from the source to that node, the label is made permanent.

For showing the operation of shortest path algorithm, let an undirected graph is taken, where distance is taken as a metric. Figure below shows the procedure to find a shortest distance from A to D.



Computing the shortest path from A to D

Flooding: It is a static algorithm in which every incoming packet is sent out on every outgoing line except the one it arrived on. Flooding obviously generates large number of packets, in fact, an infinite number of unless some measures are taken to damp the process. One such measure is to place a hop counter in the header of each packet that is decremented at each loop, with the packet begin discarded when the counter reaches zero. Ideally, the hop counters are to be placed in al the way from source to destination.

An alternative technique for damping the flood is to trace which packets have been flooded in order to avoid sending then again.

Selective Flooding: is more practical than the ideal one. In this algorithm, the routers do not send every incoming packet out on every line, but only on those lines that are going to approximately in the right direction. Flooding is not practical in most applications, but it does have some uses:

i) In military applications where large number of routers may be blown at any instant, flooding is highly desirable.

ii) In distributed database application, it is often necessary to update all the databases concurrently, in which flooding can be useful.

iii) Since flooding always chooses the shortest path as it chooses every possible path in parallel, consequently no other algorithm can produce a shorter delay.

Dynamic Routing Algorithm:

Generally, modern computer networks generally use dynamic routing algorithms rather than the static ones. There are two most popular dynamic algorithms: distance vector routing and link state routing.

a) Distance Vector Routing:

In distance vector routing algorithm, each router maintain a table (i.e. vector) giving the best known distance to each destination and which line to use to get there. These tables are updated by exchanging information with the neighbors.

The distance vector routing algorithm is also known as Bellman-Ford Routing Algorithm and Ford Fulkerson Algorithm. It was the original ARPANET routing algorithm and was also used in the Internet as RIP and in early versions of DECnet and Novell's IPX. AppleTalk and Cisco router also used improved distance vector protocols.

In distance vector routing, each router maintains an indexed routing table and contains one entry for each router in the subnet. This entry contains two parts: the preferred outgoing line to use for that destination and an estimated time or distance to that destination. The metric used might be number of hops, time delay in milliseconds and total number of packets queued along the path. The router is assumed to known the "distance" to each of its neighbors. If the metric is hop, the distance is one hop. If the metric is queue length, the router simply examines each queue. And if the metric is delay, the router can measure it directly with special ECHO packets that the receiver just timestamps and sends back as fast as it can.

b) Link State Routing: The distance vector routing had two major drawbacks; first, it was unable to take line bandwidth into accounts when changing routers and second, the algorithm often took too long to converge. Thus distance vector was completely replaced by a new algorithm called link state routing.

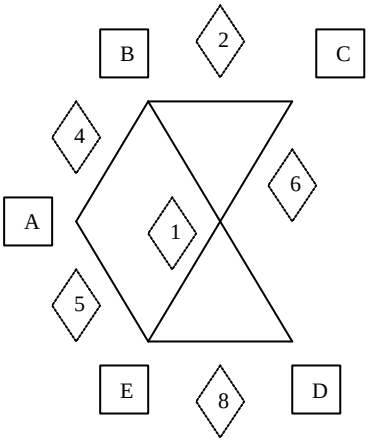
Link state routing can be stated into 5 parts, where each router must:

i) Discover its neighbors and learn their network addresses.
ii) Measure the delay cost to each of its neighbors.
iii) Construct a packet telling all it has just learned.
iv) Send this packet to all other routers.
v) Compute the shortest path to every other router.

i) Learning about the neighbors: When a router is booted, its first task is who its neighbors are. It is done by sending a special HELLO packet on each point-to-point line. The router on the other end is expected to send back a reply telling who it is.

ii) Measuring line cost: Each router should know or at least have a reasonable estimate of the delay to each of its neighbors. It can be done by sending a special ECHO packet over the line that the other side is required to send back immediately. My measuring the round trip twice and dividing it by two, the sender can get a reasonable estimate of the delay.

iii) Building link state packets: Once the information required for the exchange has been collected, the next step for each router is to build a packet containing all the data. Here, the packet starts with identity of the sender, followed by a sequence number, age and a list of neighbors. For each neighbor, the delay to that neighbor is given:



A	
Seq.	
Age	
B	4
E	5

B	
Seq.	
Age	
A	4
C	2
D	6

C	
Seq.	
Age	
B	2
E	1

D	
Seq.	
Age	
A	5
C	1
D	8

A subnet and the link state packet for it

iv) Distributing the link state packets: Distributing the link state packets reliably is the trickiest part of the algorithm. Here, flooding is used to distribute the link state packets. To keep the flood in check, each packet contains a sequence number that is incremented for each new packet sent. Routers keep track of all the (source router, sequence) they see. Whenever a new packet is received; it is checked against the list of packets already seen. If it is new, it is forwarded on all line except the one it arrived on, otherwise it is discarded. Similarly, if a packet with a sequence number lower than the highest one seen, then it rejected as being obsolete. *[Flooding: When a link state packet comes into a router for flooding, it is not queued for transmission immediately. Instead, it is kept in the holding area. If another link state packet from the same source comes before it is transmitted, their sequence numbers are compared and if they are equal, the duplicate is discarded. If they are different, the older one is expelled. To guard against errors on the router-router line, all link state packets are acknowledged.]*

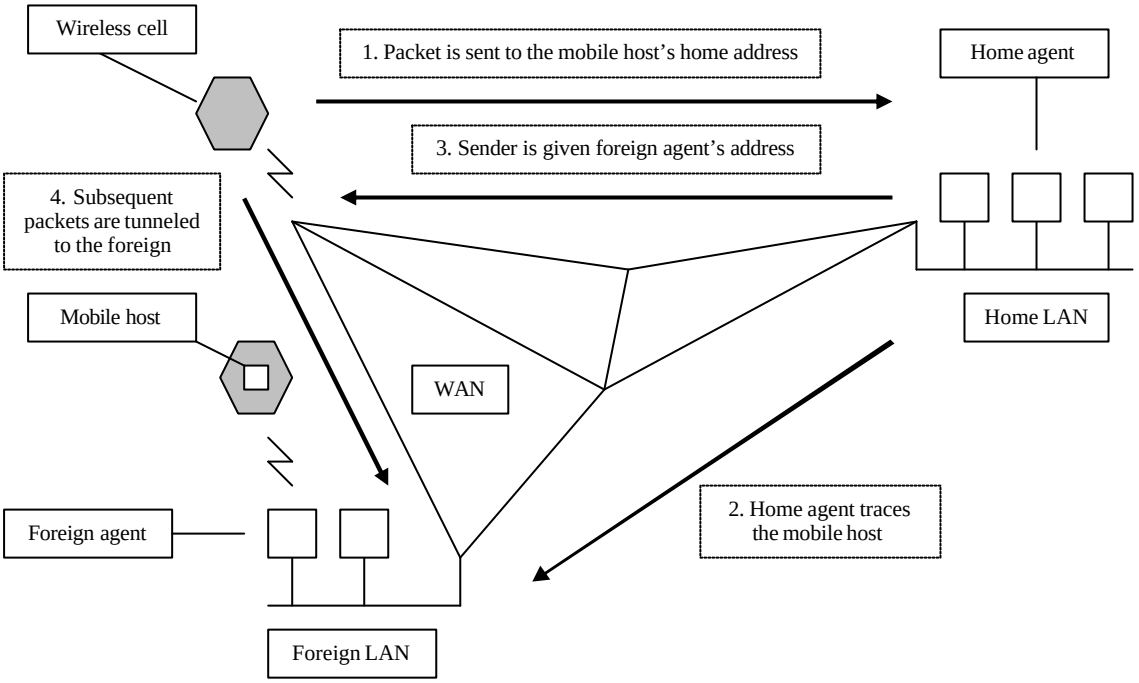
v) Computing the New Routes: Once a router has accumulated a full set of link state packets, it can construct the entire subnet graph because every link is represented. Link state routing is widely used in actual networks. IS-IS is the link state protocol being described. IS-IS (Intermediate System – Intermediate System) was designed for DECnet and later adopted by ISO for connectionless network layer protocol. Basically IS-IS distributes a picture of the router topology, from which the shortest paths are computed. Each router announces in its link state information, which network layer addresses it. These addresses can be IP, IPX, AppleTalk, etc. IS-IS can even support multiple network layer protocols at he same time.

Routing for Mobile Hosts:

WAN consists of router and hosts in which, LANs, MANs and wireless cells are connected to each other. Users who never move are said to be stationary and are connected to the network by copper wires or fibers. Migratory users are basically stationary users who move fro one fixed site to another but are the network only when they are physically connected to it. Roaming users compute on the run and want to maintain their connections as they move around. Migratory and roaming are jointly used known as mobile users.

All users are assumed to have a permanent frame location that never changes. Here, the main objective is to make it possible to send packets to mobile users using their home addresses and make reach their packets wherever they be. For it the world is divided up (geographically) into small units (say area). Each area has one or more foreign agents that keep track o fall mobile users visiting the area. They also have home agents that keep track of users whose home is in the area. When a new user enters an area, either by connecting to it or enters into the cell, his computer must register itself with the foreign agent there.

When a packet is sent to a mobile user, it is routed to the user's home agent. Then the home agent finds the mobile user's new (temporary) location and finds the address of the foreign agent handling the mobile user. The home agent sends the packet is the payload field of an outer packet to the mobile user as a data link frame. Then the home agent again asks the sender to send packets to the mobile host by addressing the packet to the foreign agent. Finally packet is routed to the mobile user via the foreign agent.



Packet routing for mobile hosts

Broadcasting Routing:

In some cases, hosts need to send messages to many or all other hosts such as distributing weather reports, stock market update, etc. Sending a packet to all destinations simultaneously is called broadcasting.

1) Method – I: Simply allowing the source to send a distinct packet to each destination. It is not only the waste of bandwidth, but also requires the source to have a list of all destinations.

2) Method –II: Flooding is another option, but is unsuitable for point-to-point communication broadcasting comparatively. Because it generates too many packets and consumes too much bandwidth.

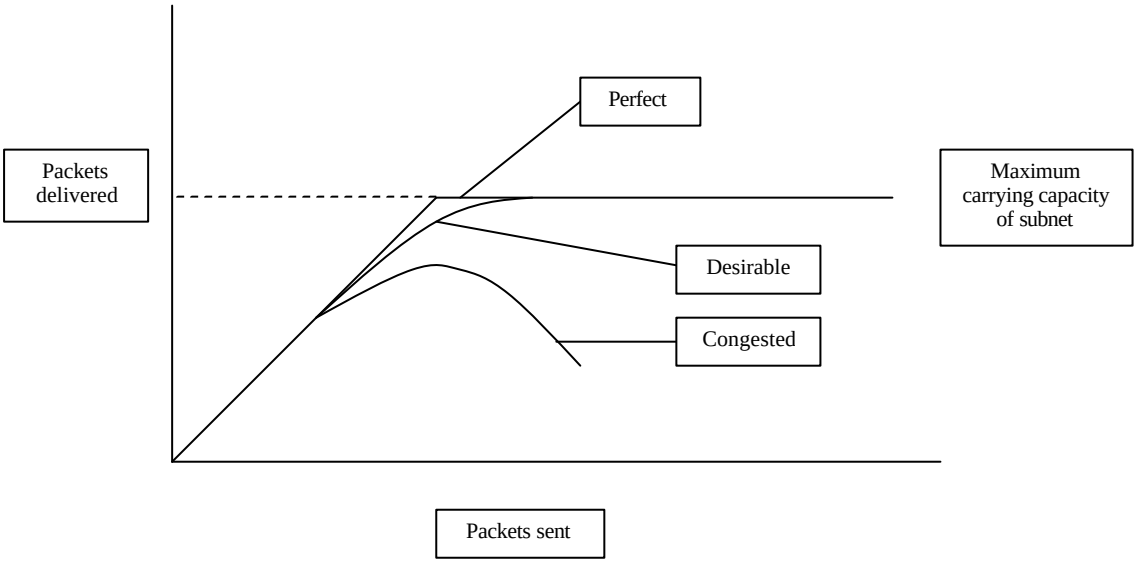
3) Method – II: In multidestination routing algorithm, each packet contains either a list of destination or a bit map indicating the desired destinations. When a packet arrives at a router, it checks all the destinations to determine the set of output line s that will be needed. Then the router generates a new copy of the packet for each output line to be used. Consequently, the destination set is partitioned among the output lines and after the sufficient number of hops, each packet will carry only one destination and can be treated as a normal destination that are to use the line.

4) Method – IV: Spanning tree is a subset of the subnet that includes all the routers but contains no loops. Here, each router copy an incoming broadcast packet on to all the spanning tree lines except the one it arrived on. By this method, there is an excellent use of bandwidth, generating the absolute minimum number of packets necessary to do the job. The only problem is each router must have knowledge of some spanning tree for it to be applicable.

5) Method – V: Reverse path forwarding is just the modification of spanning tree routing algorithm. Here, even when the routers do not know any information about the packet or spanning tree like distance vector routing, the packets can be handled efficiently. When a broadcast packet arrives at the router, the router checks to see if the packet arrived on the line that is normally used for sending packets to the source of broadcast. If so, then it makes a copy of the packet and forwards onto all lines except the one it arrived on. The principal advantage of this routing is that of being reasonably efficient and easy to implement. It does not require routers to know about spanning trees and does not require having destination list or biting map in each packet and also does not require stopping any special mechanism to stop the process, as flooding does.

Congestion Control Algorithms:

When too many packets are present in the subnet, performance degrades, this situation is called congestion. When the number of packets dumped into the subnet by the hosts within its carrying capacity, they all are delivered, but as traffic increases too far, the routers are no longer be able to cope and they begin to lose packets. At very high traffic, performance collapses completely and almost no packets are delivered. Ideally the number of packets delivered is proportional to the number sent.



Too much of traffic set the congestion, resulting sharp performance degrade

Factors for the condition:

- Insufficient memory: If streams of packets begin arriving on three or four input lines and al need the same output line simultaneously, a queue will build up. If there is insufficient memory to hold all of them, packets will be lost. But adding more memory also worsens the congestion because by the time packets get to the front of the queue, they have already timed out (repeatedly) and duplicates have been sent.

- Slow processors: If the router's CPU are slow at performing the tasks like queuing buffers and updating tables, queues can build up, even though there is excess line capacity.

- Absence of free buffers: If a router has no free buffers, it must ignore newly arriving packets. When a packet is discarded, the sending router may timeout and retransmits it (many times). Since it cannot discard the packet until it has been acknowledged, congestion at the receiver's end fores the sender to refrain from releasing a packet.

Traffic Shaping:

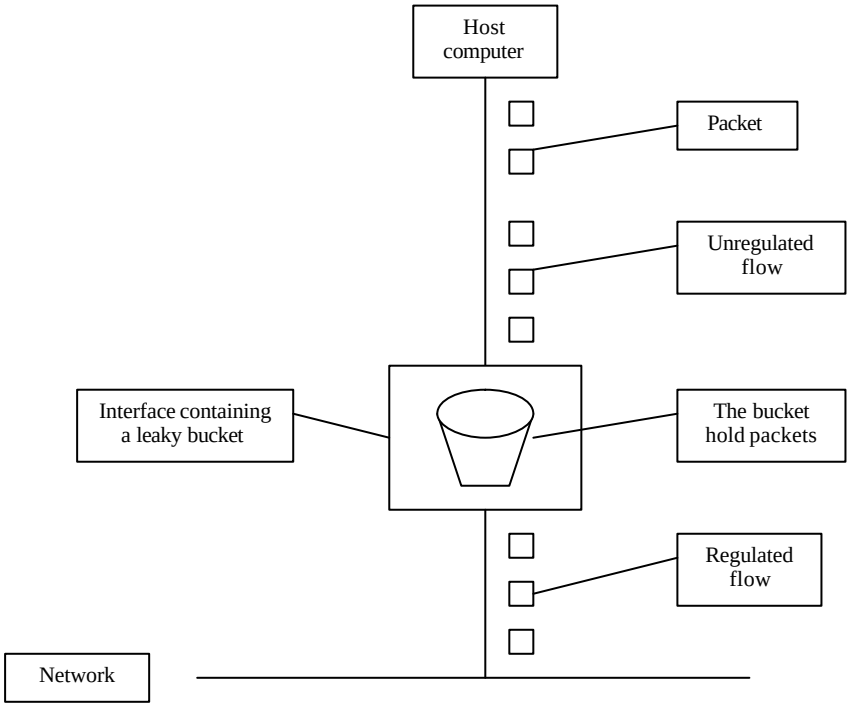
One of the main causes of congestion is that traffic is often burst. If hosts could be made to transmit at a uniform rate, congestion would be less common. This approach to congestion management is widely used in ATM networks and is called traffic shaping. Traffic shaping is about regulating the average rate of data transmission. When a virtual circuit is setup, the user and the subnet agree on a certain traffic pattern for that circuit. Traffic shaping thus reduces congestion. Such arrangements are not so important for file transfers but are of great importance for real-time data, such as audio and video connections.

Monitoring a traffic flow is called traffic policing. Agreeing to a traffic shape and policing it afterward are easier with virtual circuit subnet, than with datagram subnets.

The Leaky Bucket Algorithm:

Here, each host is connected to the network by an interface containing a leaky bucket, i.e. a finite internal queue. If a packet arrives at the queue when it is full, the packet is discarded. Basically this algorithm is nothing other than a single-server queuing system with constant service time.

The host is allowed to put one packet per clock tick on to the network. And at every clock tick, one packet is transmitted. When the packets are all the same sizes, it is convenient, but when variable-sized packets are being used, it is often better to allow a fixed number of bytes per tick, rather than just one packet.



A leaky bucket with packets

Congestion Control in Virtual Circuit Subnets:

Admission control is one of the widely used for congestion control, where once congestion has been signaled, no more virtual circuits are setup until the problem has gone away. Thus, attempts to set up new transport layer connections fail.

An alternative approach is to allow new virtual circuits but carefully route all new virtual circuits around problem areas such that omitting the congested routers and all of their lines.

Another strategy is to negotiate an agreement between the host and subnet when a virtual circuit is setup. This agreement normally specifies the volume and shape of the traffic, quality of service required and other parameters. For it, the subnet typically reserves resources along the path when the circuit is setup. Hence congestion is unlikely to occur on the new virtual circuits since all the necessary resources are guaranteed to be available. But its demerit is the wastage of resources.

Congestion Control for Multicasting:

Flow-based Algorithms:

During 1995-97, IETF devised architecture for streaming multimedia (RFCs 2205 – 2210) whose generic name was given as *flow-based algorithms* or *integrated services*. It was focused at both unicast and multicast applications, such as a single user streaming a video clip from a news site and a collection of digital television stations broadcasting their programs as streams of IP packets to many receivers at various locations.

a) RSVP – The Resource ReserVation Protocol: It is the main IETF protocol for the integrated services architecture. This protocol is used for making reservations, while other protocols are used for sending the data. RSVP allows multiple senders to transmit to multiple groups of receivers, permits individual receivers to switch channels freely, and optimizes bandwidth use while at the same time eliminating congestion.

RSVP uses multicast routing using spanning trees. Here, each group is assigned a group address. To send to a group, a sender put the group's address in its packets. Then the standard multicasting routing algorithm builds a spanning tree covering all group members. Extra information is added to the groups so that they periodically tell the routers along the tree to maintain certain data structure in their memories.

To get better reception and eliminate congestion, any of the receivers in a group can send reservation message up the tree to the sender. The message is propagated using reverse path forwarding. At each hop, the router notes the reservation and reserves the necessary bandwidth. If the available bandwidth is insufficient, it reports back failure. When making a reservation, a receiver can specify one or more source that it wants to receive from. It can also specify whether these choices are fixed for the duration of the reservation. The routers use this to optimize bandwidth planning.

Chapter – 8

Network Servers and Protocols

HTTP:

Hyper-Text Transfer Protocol is an ASCII protocol that is constantly evolving. The HTTP is classified into two parts: the set of requests from browsers to servers and the set of responses going back the other way (from server to browser).

All the newer versions of HTTP support two kinds of requests: simple requests and full requests. A simple request is just a single GET line naming the page desired, without the protocol version. In return, only the raw page is obtained with no headers, no MIME and no encoding. Full requests are indicated by the presence of the protocol version on the GET request line. Here, requests may consist of multiple lines followed by a blank line to indicate the end of the request. The first link of a full request contains the command, the page desired and the protocol version. Subsequent line contains RFC 822 headers.

Method	Description
GET	Request to read a web page
HEAD	Request to read a web page's header
PUT	Request to store a web page
POST	Append to a named resource
DELETE	Remove the web page
LINK	Connects two existing resources
UNLINK	Breaks an existing connection between two resources

The built-in HTTP request methods

Uniform Resource Locators:

URL effectively serves as the page's world wide name. URLs have three parts: the protocol, the DNS name of the machine on which the page is located and the file name of the specific page. The URLs have been designed not only to allow users to navigate the Web, but to deal with FTP, news, Gopher, email and Telnet as wel, thus making all the specialized interface programs for other unnecessary services and thus integrating nearly all Internet access into a single program, the Web browser.

Simple Mail Transfer Protocol:

Within the Internet, email is delivered by the source machine by establishing a TCP connection to port 25 of the destination machine. Listening to the port is done using SMTP by an email daemon. This daemon accepts incoming connections and copies messages from them into appropriate mailboxes. If a message cannot be delivered, an error report containing the first part of the undeliverable message is returned to the sender. Here, SMTP is a simple ASCII protocol.

After establishing a TCP connection to port 25, the sending machine waits for talking with the receiving machine. Then the receiving machine sends a line of text giving its identity and tells whether or not it is prepared to receive mail. If it is not, the sender releases the connection and tries again later.

If the receiver is willing to accept email, the client announces whom the email is coming from and whom it is going to. If such a recipient exists at the destination, the receiver gives the sender the go ahead to send the message. Then the message to is sent and the receiver acknowledges it. Here, normally no checksums are required because TCP provides a reliable byte stream. When all email has been sent, the connection is released.

Final Delivery: Generally, it is not necessary that all computers are on the Internet. Thus they are not capable of sending or receiving email from the outside world. Instead, the company or organization has one or more email servers that can send and receive emails. To send or receive messages, a PC must communicate to an email server using some kind of delivery protocol.

A simple protocol used for fetching email from a remote mailbox is POP3 i.e. Post Office Protocol. Its function is to fetch email from the remote mailbox and store it on the user's local machine. It has a command for authentication, log out, fetch messages and delete messages. POP3 consists of ASCII texts and some components of SMTP.

IMAP (Interactive Mail Access Protocol) is a more sophisticated delivery protocol, designed for the users who use multiple computers (in office, home or on the road). Its basic characteristic is to maintain a central repository for the email server, so that they can be accessed from any machine. IMAP also addresses mail by using attributes, not by arrival number. Using IMAP, a mailbox looks like a relational database system than a linear sequence of messages.

World Wide Web:

The WWW is an architectural framework for accessing linked documents spread over thousands of machines all over the Internet. From user's view, the web consists of a vast, worldwide collection of documents called pages, where each page may contain links (pointers) to other related pages anywhere in the world. Pages that point to other pages uses hypertext. Pages are viewed with a program called "browser", which fetches the requested page, interprets the text and formatting commands that it contains and displays the page, properly formatted on the screen. Strings of text that are links to other pages are called hyperlinks. These are highlighted, either by underlining, or displayed them in a special color (generally blue) or sometimes both. If the user even returns to the main page, the links that have already been followed may be shown with a dotted underline or a purple color to distinguish them from links that have not been followed. In addition to having ordinary text and hypertext, web pages can also contain icons, maps and images, where each of them can (optionally) be linked to another page. When hypertext pages are mixed with other media, the result is called multimedia. Some pages contain forms that request the user to enter information like searching a database for a user-supplied item, ordering a product or participating in a public opinion survey. Some browsers use the local disk to cache pages that they have fetched. Before a page is fetched, a check is made to see if it is in the local cache. If so, it is necessary to update.

To view a web page from the remote web server, a machine must be directly connected to the Internet, or at least have a SLIP or PPP connection to a router or other machine that is directly on the Internet. For it, the TCP connection has to be established between the web server and a machine asking for the page.

The Server Side View: Every web site has a server process listening to TCP port 80 for incoming connections from clients. After a connection has been established, the client sends a request and the server replies. Then the connection is released. The protocol that defines the legal requests and replies is called HTTP. The steps that followed during the web page displayed are as follows:

1. The browser determines the URL.
2. The browser asks DNS for the IP address of the given URL.
3. DNS replies with the IP address.
4. The browser makes a TCP connection to port 80 on given IP address.
5. It then sends a GET command.
6. The Web Server sends the file.
7. The TCP connection is released.
8. The browser displays all the contents.

Proxy Server: It is a kind of gateway that speaks HTTP to the browser but FTP, Gopher or some other protocols to the server. It accepts HTTP requests and translates them into say, FTP requests. The proxy server can be a program running on a same machine, but can also be on a free standing machine somewhere in the network serving many browsers.

Caching is another important characteristic of the proxy server that keeps all the pages that pass through it. It checks to see if the page is still current. In case the page is still current, it is passed to the user; otherwise a new copy is fetched.

Finally an organization can put a proxy server inside its firewall to allow users to access the web, but restricting them full Internet access. Here, users can talk to the proxy server and the proxy server contacts the remote sites and fetches pages on behalf of its clients.

Domain Name System (DNS)

It is a name service, whose prime naming database is used across the Internet. It was introduced by Mockapetris in 1987 to replace the original Internet naming scheme in which all host names and addresses were held in a single central master file and the computers requiring it had to download by FTP. The objects named by DNS are primarily computers, for which IP addresses were mainly stored as attributes. Same as in SNS, organizations and departments within them can manage their own naming data. The major difference between the requirements fulfilled by DNS and SNS model is that of scale. Hundreds of thousands names are bound by the Internet DNS, and lookups are made against it around the world. Any name can be resolved by any client. This is done by hierarchical partitioning of the name database, by replication of the naming data and by caching.

Domain names: DNS names are called *domain names*. A domain name consists of one or more strings called *labels* separated by the delimiter ".". There is no delimiter at the beginning or end of a domain name. Domains are collections of domain names; syntactically, a domain's name is the common suffix of the domain names within it. The Internet DNS name space is classified both organizationally as well as geographically. The names are written with the highest-level domain on the right, which is opposite of SNS names. Presently, the top level organizational domains across the Internet are as follows:

com - *Commercial Organization*
edu - *Educational Institutes*
gov - *Government Offices*
mil - *Military Services*
net - *Network Solution Providers*
org - *Organizations (excluding of the above)*

Apart from them, the countries also have their own domain names, such as "us" for USA, "uk" for United Kingdom, "np" for Nepal, "ru" for Russia, "in" for India, etc.

DNS Queries:

Host name resolution: Generally, applications use DNS to resolve the host names into IP addresses. For instance, the domain name *ncit.net.np* might be given to a web browser program - such as *internet explorer* – as the name of a computer which stores the web page for that domain. Internet explorer makes a DNS enquiry and obtains the IP address of *ncit.net.np*, in order to communicate with a http daemon (httpd) on that computer.

Mail host location: Email software like Outlook Express uses DNS to resolve domain names into IP addresses of mail hosts – computers that accept mail for those domains. For instance, when the email address *student@ncit.net.np* is to be resolved, DNS is queried with the address *ncit.net.np* and type the designation 'mail'. The DNS may return more than one domain name so that the mail software can try alternatives if the main mail host is unreachable. The DNS returns an integer preference value for each mail host, indicating the order in which the mail hosts should be tried.

Apart from them, some other types of queries are possible like:

Reverse resolution: In some cases, it is necessary to return a domain name on the submission of an IP address. It is just the reverse of the normal host name query. A special domain, *in-addr.arpa* is used to hold IP addresses for reverse lookups.

Host information: DNS can also store the machine architecture type and operating system against the domain name or an IP address of hosts. But the exposure of such information is vulnerable for unauthorized access.

Well known services: A list of services run by a computer (telnet, ftp, http) and the protocol used to obtain them (UDP, TCP) can be returned, given the host's domain name.

DNS name servers: The DNS database is distributed across a logical network of servers. Each server holds part of the naming database, primarily for the local domain. Most of the queries related to the host in the local domain are resolved by the servers within that domain. However, each server records the domain names and addresses of other (higher) name servers, so that the queries outside the domains can be resolved.

The DNS naming data are divided into zones. A zone contains the following data:

- Attribute data for names in a domain, less any sub-domains administered by lower-level authorities. For instance, a zone could contain data for NCIT College – *ncit.net.np* – less the Research and Consultancy Unit – *rcu.ncit.net.np.*
- The names and addresses of name servers that provide authoritative data for the zone.
- Zone management parameters, such as those governing the caching and replication of zone data.

There are two types of server that provide authoritative data for the zone. *Primary or master server,* that read zone data directly from a local master file. *Secondary or slave server* that downloads zone data from a primary server. They communicate periodically with the primary server, to check the updates in the master server. Moreover, any server is free to cache data from other servers for avoiding repeated query for the same data.

DNS resource records

Record type	Description
A	A computer (IP) address
NS	An authoritative name server
CNAME	The canonical name for an alias
SOA	State of Authority
PTR	Domain name pointer (for reverse lookups)
HINFO	Host information
MX	Mail Exchange

File Transfer Protocol (FTP):

FTP is used for uploading or downloading files between the local and remote machines. This is one of the very popular protocol in the Internet that is used for numerous purposes like uploading files in the concerned folder of the web server or downloading larger files (like images, audio, video, text, software, etc) from the remote server. For accessing FTP service, there are two types of users defined: anonymous and non-anonymous users. Anonymous users can be anyone with email address as their password. Anonymous users are allowed generally for downloading free documentations, free applications, games, audio and video files, etc. Anonymous users can also upload the files mostly in the user's group sites. Non-anonymous users are the specific users who have the authority to access into the server, especially for uploading and downloading the web contents of their respective web sites, university students for retrieving and submitting their assignments, etc. The basic ftp commands are as follows:

1. ftp> dir → list the contents of the current directory
2. ftp> cd → change the current directory
3. ftp> lcd → change the current local directory
4. ftp> rm → delete the content
5. ftp> mkdir → make directory
6. ftp> get → download a file
7. ftp> put → upload a file
8. ftp> mget → download multiple file
9. ftp> mput → upload multiple file
10. ftp> bye → exit FTP application

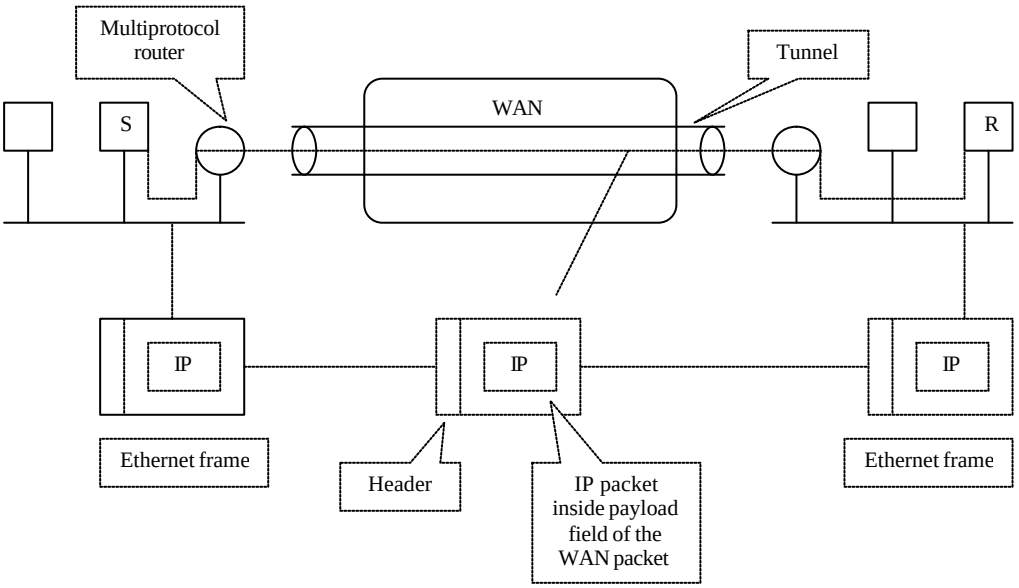
Dynamic Host Configuration Protocol (DHCP):

DHCP is one of the popular protocols in the networking environment. DHCP is responsible for assigning IP addresses to the various machines in the network. When a machine with a NIC card is booted up, it sends a DHCP request for obtaining an IP address. Then the DHCP server listens to it and grants the unique IP address to it so that the workstation/ machine can communicate other machines in the network. DHCP server assigns an IP address either randomly or as per the MAC address of the NIC card of the machine. The following specifications are defined by a DHCP server to the DHCP client:

1. IP address within its range.
2. Subnet Mask
3. Router Address
4. Domain Name
5. Internet Gateway

Tunneling:

It is the process of making two different networks internetwork. It is applicable when the source and destination hosts are on the same type of network, but different network in between. For instance, let there be two TCP/IP base Ethernet in the source and destination and a PTT WAN in between. To send an IP packet to the destination, sender constructs the packet containing the IP address of the receiver, inserts into an Ethernet frame addressed to the sender side's multiprotocol router. When the multiprotocol router gets the frame, it removes the IP packet, inserts it in the payload field of the WAN network layer packet and addresses the latter to the receiver side's multiprotocol router. When it gets there, the receiver side's router removes the IP packet and sends it to receiver inside an Ethernet frame.



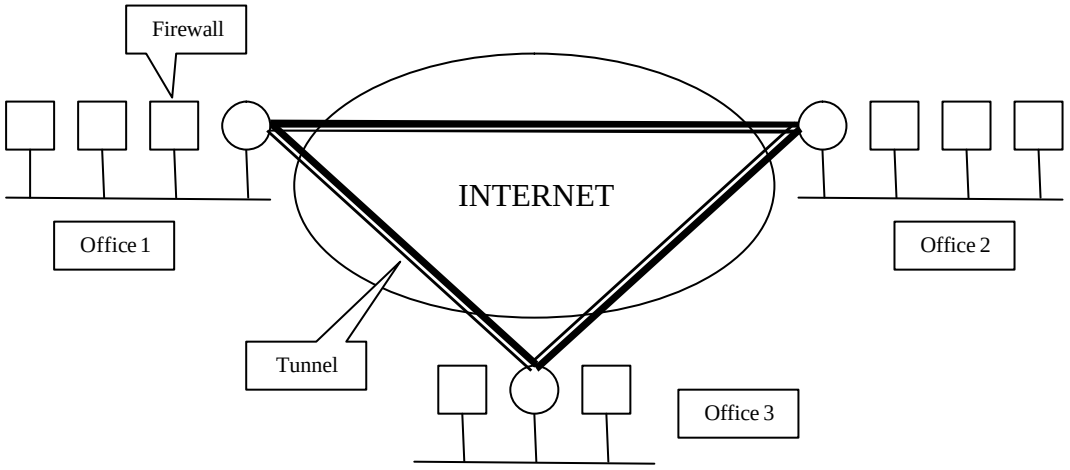
Tunneling a packet from sender to receiver

Virtual Private Network:

In the earlier days, the remote computers were interconnected via the leased telephone lines. A network built up from computers of the same organizations at various locations using leased telephone lines is called a private network. Though private network proved to be pretty secure and efficient, it proved to be quite costly. This led to the creation of VPN (Virtual Private Network), which run top of public networks but with most of the properties of private networks. They are called "virtual" because they are merely an illusion, just as virtual circuits and not real circuits.

Nowadays, VPNs being build up directly over the Internet are getting quite popular. The basic principle of VPN design is to equip each office with a firewall and create tunnels through the Internet between all pairs of offices.

For operating the system, each pair of firewalls has to negotiate the parameters of its Security Association, including the services, modes, algorithms and keys. Once the SAs have been established, traffic can begin flowing. To a router within the Internet, a packet traveling along a VPN is just an ordinary packet. The prime benefit of using a VPN is that it is completely transparent to all user software.



A virtual private network

Chapter - 9

Network Management and Security

In the present context, millions of hosts are using networks for banking, shopping, email, entertainment, etc and consequently arising more problems on network security. It is especially concerned with people trying to access remote services that they are not authorized to use and intended to gain some profit or harm some one. It should be note that making the network secure is more tedious than just keeping it free of programming errors. Normally, network security problems are of four types: secrecy, authentication, non-repudiation and integrity control. Secrecy has to do with keeping information out of the hands of unauthorized users. Authentication deals with determining whom you are talking to before revealing sensitive information. Non repudiation deals with signature that verifies the data has come from the correct user or source.

Introduction to Security Aspects:

To guard against various threats to the security of networking environment, *security policies* must be adopted, that are designed to ensure appropriate levels of security for the activities that are performed in the system and *security mechanisms* must be employed to implement security policies. For instance, the provision of a lock on a door does not ensure the security of a building, unless there is a policy for its use. When designing secure systems, the distinction between security policies and mechanisms is to be found out. But it is often difficult to be confident that a given set of security mechanisms fully implements the desired security policies. And it should be noted that the security policies are independent of the technology used.

In order to demonstrate the validity of the security mechanisms employed in a system, the system's designers must first make a list of threats, methods by which the security policies might be violated and show that each of them is prevented by the mechanisms employed. No list of threats is like to be comprehensive. Hence auditing methods must also be used in security-sensitive applications to detect violations, which are based on a log of security-sensitive system actions with details of the users performing the actions and their authority.

The term *principal* is used to refer the agents accessing the information or resources that are held in a distributed system. A principal is a person or a process (client, server or group member). Principals are able to obtain access to resources.

1) *Threats:* The purpose of a security system is to restrict access to information and resources to authorized principals. For designing a system which is secure against specific threats, it is necessary to classify the threats and the methods by which each of them may be achieved. Security threats to computers can be classified into 4 categories:

   Leakage: The acquisition of information by unauthorized recipients.

   Tampering: The unauthorized alteration of information.

   Resource stealing: The use of facilities without authorization.

   Vandalism: Interference with the normal operation of a system without authorization.

2) *Methods of attack:* To violate a system in any means require an access to the system. Virtually, all computers have communication channels for authorized access to their resources, and these are means through which an authorized access can be gained. In distributed systems, computers are attached to a network and their operating systems offer a standard communication interface which enables an establishment of virtual communication channels. The following are the methods by which security violations can be perpetrated in distributed systems:

   Eavesdropping: Obtaining copies of messages without authority. It can be done by obtaining messages directly form a network or by examining information that is inadequately protected

in storage. For instance, changing a workstation's network address to some other station on the network, enabling it to receive messages addressed to that station.

Masquerading: Sending or receiving messages using the identity of another principal without his authority. It can be done by obtaining and using other principal's identity and password.

Message tampering: Intercepting messages and altering their contents before passing them on to the intended recipient.

Replaying: Storing message and sending them at a later date. For instance manipulating data after authorization to use a resource has been revoked.

3) *Infiltration:* In order to launch attacks in distributed system, the attacker must have access to the system for running the program that implements the attack. Most attacks are launched by one of the legitimate users of a system. They abuse their authority by running programs that are designed to carry out one of the above forms of the attack. One of the simplest methods of infiltration is by guessing passwords or running 'password cracking' programs to obtain password of a known user. Such attacks can be prevented by the use of standard passwords. Apart from the direct forms of infiltration, there are various methods; some of them are as follows:

Virus: It is a program that is attached to a legitimate host program and installs itself in the target environment whenever the host program is run. Once it is installed, it performs its criminal actions, often as per a date. One of its other act is to replicate itself by attaching itself to all of the programs (as web requests, email attachments, etc) that it can find in the target environment. They travel between machines whenever a host program is moved, or by network communication or by the transport of physical storage.

Worm: It is a program that exploits facilities for running processes remotely in distributed system. Worms are either created accidentally or intentionally.

Trojan horse: It is a program that is offered to the users of a system as performing a useful function, but has a second malicious function hidden in it. The most common example is the 'spoof login', a program that prompts the user, very much similar to the regular login or password dialogue, but actually it stores the innocent user's input in a convenient file for later purpose.

Simple Network Management Protocol:

SNMP was designed with a view to provide a systematic way of monitoring and managing a computer network. It is widely implemented in commercial products and became the standard for network management. The SNMP model of a managed network consists of four components:

1. Managed nodes
2. Management stations
3. Management information
4. Management protocol

The managed nodes can be hosts, routers, bridges, printers or any other device capable of communicating status information to the outside world. A node must be capable of running an SNMP management process, called an SNMP agent. Each agent maintains a local database of variables that describe its state and history and affect its operation.

Management stations manages network, which are actually ordinary computers running special management software. The management stations contain one or more processes that communicate with the agents over the network, issuing command and getting responses. Its main objective is to keep the agents as simple as possible and minimize their impact on the devices they are running on.

It is obvious that the devices used in the network are normally from different manufacturers. Consequently, to allow a management station to communicate with those devices, all their information must be specified. Thus each device maintains one or more variables that describe its state, which are known as objects. But it should be noted that they are different from object-oriented system. The collection of all possible objects in a network is given in a data structure called the MIB (Management Information Base).

The management stations interact with the agents using the SNMP protocol. The management protocol (SNMP) allows the management station to query the state of an agent's local objects and change them if necessary.

Structure of Management Information (SMI): is used to define data structure. At the lowest level, SNMP variables are defined as individual objects. Related objects are collected together into groups that are further assembled into modules. It is designed so that the vendors supporting a group support all the objects in that group. But it is not necessary for a vendor supporting a module need not support all of its groups.

All MIB modules start with an invocation of the MODULE – IDENTITY macro. Its parameter provides the name and address of the implementer and other administrative information. It is followed by an invocation of the OBJECT – IDENTITY macro that tells where the module fits in the naming tree. The next one is the OBECT – TYPE macro, name the actual variables being managed and specify their properties. The OBJECT – TYPE macro has four required parameters and four optional ones. The first required parameter is SYNTAX that defines the variable's data type. The next one is MAX ACCESS which contains information about the variable's access. The most common values are read-write and read-only. The STATUS has three possible values. A current variable is conformant with an older version. A deprecated value is in between and is really obsolete. The last parameter is DESCRIPTION, which an ASCII string is telling what the variable does.

Management Information Base (MIB): The collection of objects managed by SNMP is defined in the MIB. For convenience, these objects are grouped into ten categories, which correspond to the 10 nodes under MIB-2. The ten categories are intended to provide a basis of what a management station should be understood.

Sr. no.	Group	Description
1	System	Name, location and description of the equipment
2	Interfaces	Network interfaces and their measured traffic
3	AT	Address translation
4	IP	IP packet statistics
5	ICMP	Statistics about ICMP messages received
6	TCP	TCP algorithms, parameters and statistics
7	UDP	UDP traffic statistics
8	EGP	Exterior Gateway Protocol traffic statistics
9	Transmission	Reserved for media-specific MIBS
10	SNMP	SNMP traffic statistics

The **system** group allows the manager to find out the description of the device. The time of the last boot and the name and address of the contact person are also provided.

The **interface** group deals with the network adapters. It keeps track of the number of packets and bytes sent and received from the network, the number of discards, the number of broadcasts and the current output queue size.

The **AT** group provides information about address mapping. (Ethernet to IP addresses).

The **IP** group deals with IP traffic into and out of the node. It keeps track of the number of packets discarded for each of a variety of reasons. Statistics about datagram fragmentation and reassembly are also available.

The **ICMP** group is related to IP error messages.

The **TCP** group monitors the current and cumulative number of connections opened, segments sent and received, and various error statistics.

The **UDP** group logs the number of UDP datagram sent and received, and how many of the latter were undelivered.

The **EGP** group is used for routers that support the exterior gateway protocol. It keeps track of how many packets of what kind went out, came in and forwarded correctly.

The **transmission** group is a place holder for media-specific MIBs. For example, Ethernet-specific statistics can be kept here.

The **SNMP** group is for collecting statistics about the operation of SNMP itself.

Cryptography:

Cryptography is a method to transform (encrypt) information in such a way that it cannot be understood by anyone except the intended recipient, who possesses the means to reverse the transformation (decrypt). The encryption of messages plays three major roles in the implementation of secure systems:

- It is used to conceal private information that is usually exposed in parts of the system such as physical communication channels, which are vulnerable to eavesdropping and message tampering. It was traditionally used in military and intelligence activities.
- It is used in support of mechanisms for authenticating communication between pairs of principal. A principal who decrypts a message successfully using a particular inverse key can assume that the message is authentic if it contains some expected value.

It is used to implement a mechanism known as a *digital signature.* This emulates the role of conventional signatures, verifying to a third part that a message is an unaltered copy of one produced by a specified principal.

Transformation and keys: A message is encrypted by the sender applying some rule to transform if from *plain text* to *cipher text.* The receiver must know the inverse rule in order to transform the cipher text received into the original plain text. Other principals are unable to decipher the message unless they know the inverse rule.

The encryption and decryption transformations are defined into two parts, a *function* and a *key*. The function defines an *encryption algorithm* that transforms data items in plain text into encrypted data items by combining them with the key and transposing them according to some operation whose results are heavily dependent on the value of the key. For instance, a text M is encrypted with an encrypted function and a key K as $\{M\}_K$.

Public-Key Algorithms:

Since in case of symmetric keys, encryption and decryption keys were same, there was a great risk of keys being reached into the wrong hands. Hence in 1976, researchers Diffie and Hellman introduced a new approach in which the encryption and decryption keys were different. Here, the encryption algorithm and the decryption algorithm has to meet three requirements:

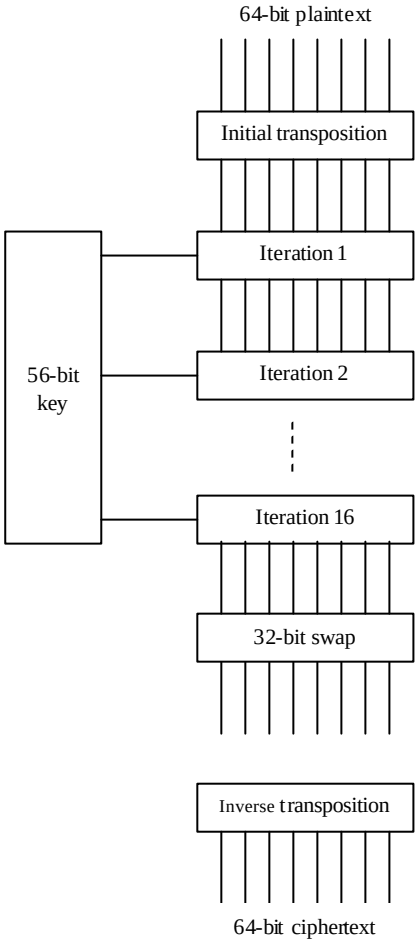
1. D(E(P)) = P
2. It is exceedingly difficult to deduce D from E
3. E cannot be broken by a chosen plaintext attack

Symmetric-Key Algorithms:

Symmetric-key algorithms use the same key for encryption and decryption. Here, block ciphers take an n-bit block of plain text as input and transform it using the key into n-bit of cipher text. *Block cipher* take an n-bit block of plaintext as input and transform it using the key into n-bit block of ciphertext.

DES – Data Encryption Standard:

DES was widely adopted for use in security products. Here, plain text is encrypted in blocks of 64 bits, yielding 64 bits of ciphertext. The algorithm, which is parameterized by a 56-bit key has 19 distinct stages. The first stage is a key-independent transposition on 64-bit plaintext, whereas the last stage is the exact inverse of this transposition. In the second last stage, the leftmost 32 bits are swapped with the rightmost 32 bits. The rest 16 stages are functionally identical but are parameterized by different functions of the key. The algorithm has been designed to decrypt with the same key to encrypt.



General outline of DES

Authentication Mechanism:

In distributed system, authentication is the means by which the identities of servers and clients are reliably established. Authentication mechanisms for distributed system take the form of an *authentication service,* which relies on the use of encryption to guarantee security. They require the provision of a secure means for generating, storing and distributing all of the encryption keys needed in a distributed system called *key distribution service.*

Needham and Schroeder first introduced a solution to authentication and key distribution based on an *authentication server* that supplies secret keys to clients. The job of the authentication server is to provide a secure way for pairs of processors to obtain keys. In order to do it, it must communicate with its clients using encrypted messages. They have described two protocols for a secure authentication server, the first using secret keys and the second using public keys.

Needham and Schroeder with secret keys: In this model, when a process on principal A wants to initialize secure communication with another process on principal B, it can obtain a key such that A acts like a client, initiating a sequence of request to B. The key is supplied to A in two forms, one that A can use to encrypt the message that it sends to B and next that it can transmit securely to B (this one is encrypted and is known to B but not to A, so that B can decrypt it).

The authentication server S maintains a table containing a *name* and a *secret key* for each principal known to the system. The secret key is used only to authenticate client processes to the authentication server and to transmit messages securely between client process and the authentication server. It is never disclosed to third parties and it is transmitted across the network at most once, when it is generated.

Needham and Schroeder with public keys: Public keys must be distributed by a trusted key distribution server in order to avoid intrusions by imposters. When obtaining a public key for use in communicating with B, A wants to be sure that the B's public key obtained is the real one and not some other public key sent to it by an imposter purporting to be B.

Case Study: Kerberos

Kerberos is an authentication protocol based on Needham and Schroeder secret-key protocol. It was developed at MIT in 1988 to provide a range of authentication and security facilities for use in Athena campus computing network and other open systems. It has gone multiple revisions and enhancements till the date. It is used to provide secure access to NFS as well as other services. Kerberos deals with three kinds of security objects:

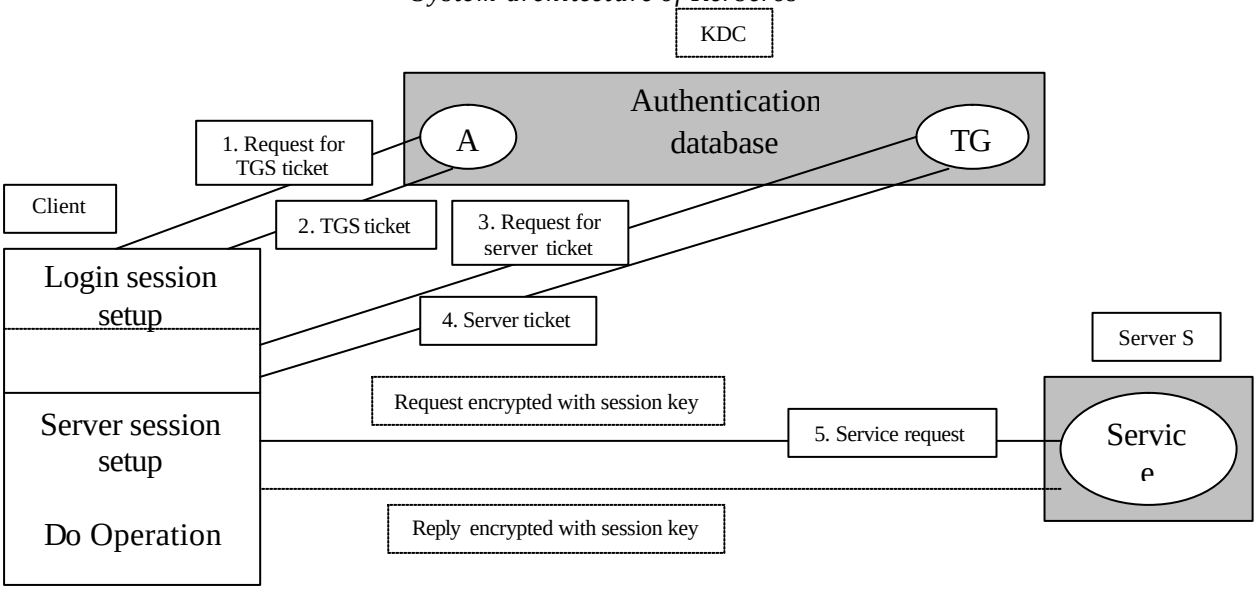
Ticket: It is a token issued to a client by Kerberos ticket-granting service, verifying that the sender has been recently authenticated by Kerberos. Tickets include an expiry time and a newly-generated session key for use by the client and the server.

Authenticator: It is a token constructed by a client and sent to a server to provide the identity of the user and the type of communication with a server. An authenticator can be used only once, which contains the client's name and a timestamp and is encrypted in the appropriate session key.

Session key: It is a secret key that is randomly generated by Kerberos and issued to a client for use when communicating with a particular server. Encryption is not mandatory for all communication with servers; the session key is used for encrypting communication with those sever that demand it as well as for encrypting all authenticators.

A Kerberos is known as a Key Distribution Centre (KDC), which offers an Authentication Service (AS) and a Ticket Granting Service (TGS). While logging in, users are authenticated by the AS, using a network-secure variation of the password method, and the client process acting on behalf of the user is supplied with a *ticket-granting ticket,,* and a session key for communicating with the TGS. Subsequently, the original client process and its descendants can use the ticket-granting ticket to obtain tickets and session keys for specific services from the TGS

System architecture of Kerberos



Logics of authentication:

A logical calculus based on an agreed set of deduction rules for formally reasoning about authentication protocols is called *logic of authentication*. The benefits can be derived from such logic:

Correctness: It should be possible to prove that a protocol either does or does not meet its security goals. If it does not achieve the stated goals, then the logic of authentication should show what it does in fact achieve.

Efficiency: If the security goals can be achieved without some of the messages, contents of messages or encryptions of message contents that are part of protocol, then the protocol can be made more efficient by eliminating them.

Applicability: In order to determine whether a protocol can be used in a practical situation, it helps to clarify the protocol's assumptions by formally stating them.

Digital Signatures:

Handwritten signatures are used to verify the document was originally produced by the signatory and that it has not subsequently been altered. It also commits a signatory – a person who has signed a document cannot subsequently disclaim it. But handwritten signatures have the following limitations:

- Forged signatures are hard to detect.
- A handwritten signature least prevent the alteration of document.
- The signatory may accidentally or by conspiracy sign a document and subsequently the signatory may be able to disclaim their signatures.
- Witness' signatures are often added in a document to authenticate the main signature.

Despite these limitations, handwritten signatures are widely used as an authentication technique for conventional documents. A handwritten signature testifies that the document was produced with the knowledge of the signatory and that it was not subsequently altered. Handwritten signatures are not applicable to computer-based documents.

In computer systems, documents or messages may be originated under the authority of one principal, transmitted to another and subsequently retransmitted to others. It is often necessary that each of the recipients should be able to verify that the claimed originator of a document is the real one and the document has not subsequently been altered, and that the originator will not be able to disclaim the document subsequently.

A digital signature should have the same authentication and legally binding functions as of handwritten signature. Once a signature is attached to an electronic document, it should be possible for any principal that receives a copy of the message from any source to verify that the document was originally sent by the signatory, and that it has not been altered.

An electronic document or message M can be signed by a principal A by encrypting a copy of M in a key K_A and attaching it to a plain-text copy of M and A's identifier. Then the signed document consists of $<M, A, \{M\}K_A>$. The purpose of appending a signature to a document is to enable any principal that subsequently receives the document to verify that it was originated by A and that its content M have not been altered subsequently.

In order to reduce the size of digital signatures for potentially large documents, a *digest function* D is used, which produces a characteristic value that uniquely identifies the message to be signed. Digest function are also called secure hash functions. They must be carefully designed to ensure that D(M) is different from D(M') for all likely pairs of messages M and M'. A message digest function called MD5 is used in secure mail and other applications in the Internet.

The verification of the signature proceeds differently depending on whether secret-key or public-key encryption is used to produce the signature.

Digital signatures with public keys: The originator A of a message M signs it by attaching a copy of D(M) encrypted in A's secret key $K_{A\ private}$ and sends to other principal. The recipient principal B may receive a copy of the message and can verify the signature by using A's public key $K_{A\ public}$ to decrypt the signature to get D(M) and comparing it with its own computation of D(M). Hence, in this case, no server is involved, except a key distribution server to supply a copy of A's public key to any principal. The protocol for A to send a signed document to Band for B to validate the signature would be:

S.No	*Header*	*Message*	*Description*
1	A → B:	M, A, $\{D(M)\}K_{A\ private}$	A sends the original message and the signature to B.
2	B → S:	A	B requests A's public key from S.
3	S → B:	A, $K_{A\ public}$	S supplies A's public key $K_{A\ public.}$ B uses it to decrypt the signature received with the message and compare it with a newly-computed value for D (M).

Digital signatures with secret keys: It is a digital signature service that is based on the authentication server. Here, A sends a signed message M to principal B using secret keys. When B receives the message from A that contains the name of A, the message M and the plain text certificate D (M), it uses the digest function to compute the value of D (M) form the message text and compares it with the value that it received from the server. The protocols used are as follows:

S.No	*Header*	*Message*	*Description*
1	A → S:	A, $\{D(M)\}K_A$	A computes D (M) and encrypts it with A's secret key and sends it to S.
2	S → A:	{A, D(M), t) K_S	S makes a signed and dated certificate of A's signature on the document by making a composite text containing A's name, D (M) and a timestamp t and encrypts it with its secret key. It sends a resulting certificate back to A.
3	A → B:	M, {A, D(M), t) K_S	A sends the message M and the certificate to B.
4	B → S:	B, {A, D(M), t) K_S	B saves a copy of the message and the certificate and then sends the certificate to S for decrypting.
5	S → B:	{A, D(M), t) K_S	S decrypts the certificate and uses B's secret key to encrypt the resulting plain text and sends it to B, where it is decrypted.

Gateway: is the entrance to the outer network to a fixed network (mostly LAN).

Router: is the device that enables to different network to communicate each other.

Firewalls:

Firewalls refer to the mechanism of checking each and every network packet while entering and departing a fixed network (mostly LAN). The two basis components of firewall are:

1. Packet filter
2. Application gateway

A packet filter is a standard router that inspects every incoming and outgoing packet. The packets which meet some specific criteria as per the routing tables are forwarded normally. But those that fail are dropped. Generally a network packet contains a source and destination IP address and their ports. Depending upon the network address or IP address or ports, the packet can be filtered.

An application gateway operates at an application level, that inspects the each specific service being properly served or not. For instance, a mail gateway examines each incoming and outgoing emails and decides whether to transmit or discard the message as per their header fields, message size, attachment, subject, etc.

Chapter - 10

Introduction to Socket Programming

Ports and Sockets:

During communication, each application layer process that uses TCP/IP protocol must identify itself with a **port** number. The port number is used between the two host computers to identify which application program is to receive the incoming traffic. In addition to it, TCP/IP based protocols also use an abstract identifier known as **socket.** Sockets can be classified into two topics: from the process to the kernel, and from the kernel to the process. The socket is derived from the network input/output operations of BSD 4.3 UNIX. The socket is termed as the concatenation of a port number and the network address (IP address) of the host that supports the port service.

The socket consists of the following C library routines:

int socket(int addr_family, int type, int protocol);
*int bind(int s, struct sockaddr *address, int address_len);*
int listen(int s, int backlog);
*int connect(int s, struct sockaddr *address, int address_len);*
*int accept(int s, struct sockaddr * address, int *address_len);*
*int send(int s, char *msg, int len, int flags);*
*int sendto(int s, char *msg, int len, int flags, struct sockaddr *to, int tolen);*
*int recv(int s, char *buf, int len, int flags);*
*int recvfrom(int s, char *buf, int len, int flags, stuct sockaddr *from, int *fromlen);*
*int getsockopt(int s, int level, into name, char *ovalue, int *olen);*
*int set socketopt(int s, int level, int oname, char *ovalue, int *olen);*

All of these functions are based on the system call *socketcall.*in addition, the system call *ioctl* to socket file descriptors enables network-specific configurations to be changed.

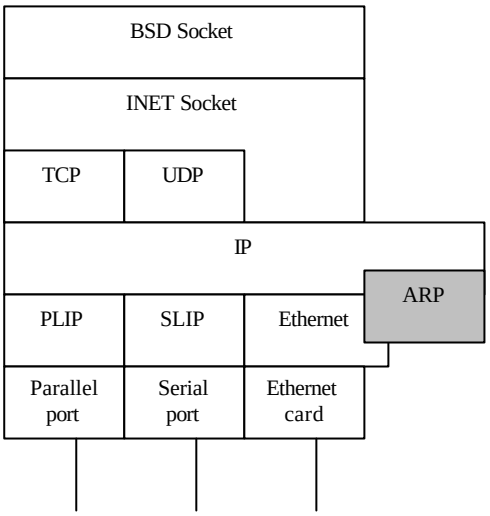
As the C library routine *socket()* returns a file descriptor, the usual I/O system calls, such as *read* and *write* are also applicable in the network. A computer is connected to a network via the variety of devices like Ethernet cards, modem, etc. Since there is no representation in the file system for the network devices, they cannot be setup in the /dev directory using *mknod* command like normal devices. Hence a network device can only be accessed if the system initialization function identifies the corresponding hardware.

The address conversion functions convert between a texts are presentation of an address and the binary value that goes into a socket address structure. Most IPv4 code uses *inet_addr* and *inet_ntoa,* but new functions *inet_pton* and *inet_ntop* is compatible with both IPv4 as well as IPv6.

The layer model of the network implementation:

When a process communicates via the network, it uses the functions provided by the BSD socket layer, which administers a general data structure for sockets, knows as BSD sockets. The BSD socket interface simplifies the porting of network applications which are pretty complex. INET socket layer is below the BSD socket layer that manages the communication end points for the IP-based protocols TCP and UDP. These are represented by the data structure *sock,* which is known as INET sockets. The layer below INET socket layer is dependent of the type of the socket which can be either TCP or UDP layer or the IP layer directly. The UDP layer implements the *User Datagram Protocol* on the basis of IP, and the TCP layer implements *Transmission Control Protocol* for reliable communication links. The IP layer contains the code for the *Internet Protocol.* Below the IP layer are the network devices, to which the IP passes the final packets. These are responsible for the physical transport of the information.

True communication occurs between two sides, producing a two-way flow of information. Hence the various layers are also connected together in the opposite direction, i.e. when IP packets are received; they are passed to the IP layer by the network devices and processed.



The Layer structure of a network

Socket Address Structures:

Most of the socket functions require a pointer to a socket address structure as an argument. Each supported protocol suite defines its own socket address structure. The names of these structures begin with *sockaddr_* with a distinct suffix for each protocol suite.

IPv4 Socket Address Structure:

An IPv4 socket address structure is also known as "Internet socket address structure", which is named as *sockaddr_in* and is defined by including the <*netinet/in.h*> header. The structure is as follows:

```
struct in_addr {
        in_addr_t       s_addr;         /* 32-bit IPv4 address */
                                        /* network byte ordered */
};

struct sockaddr_in      {
        uint8_t         sin_len;        /* length of structure (16) */
        sa_family_t     sin_family;     /* AF_INET */
        in_port_t       sin_port;       /* 16-bit TCP or UDP port number */
                                        /* network byte ordered */
        struct in_addr  sin_addr;       /* 32-bit IPv4 address */
                                        /* network byte ordered */
        char            sin_zero;       /* unused */
};
```

Generic Socket Address Structure:

When socket address structures are passed as an argument to any of the socket functions, they are always passed by reference. But the socket functions that take one of the pointers as an argument must deal with socket address structures from any of the supported protocol. Here the problem is to declare the type of pointer that is passed. The socket functions ANSI C is defined as *generic* socket address structure in the <*sys/socket.h*> header.

```
struct  sockaddr        {
        uint8_t         sa_len;
        sa_family_t     sa_family;      /* address family: AF_xxx value */
        char            sa_data[14];    /* protocol-specific address */
};
```

The generic socket address structure *sockaddr*

Then the socket functions are defined as taking a pointer to the generic socket address structure in the ANSI C function prototype for the *bind* function:

*int bind(int, struct sockaddr *, socklen_t);*

Any calls to these functions must direct the pointer to the protocol specific socket address structure to e a generic socket address structure. For instance,

```
struct   sockaddr_in     serv;     /* IPv4 socket address structure */
/* fill in serv{} */
bind(sockfd, (struct sockaddr *) &serv, sizeof(serv));
```

In short, the generic socket address structures are only used to direct pointers to protocol-specific structures.

Value-Result Arguments:

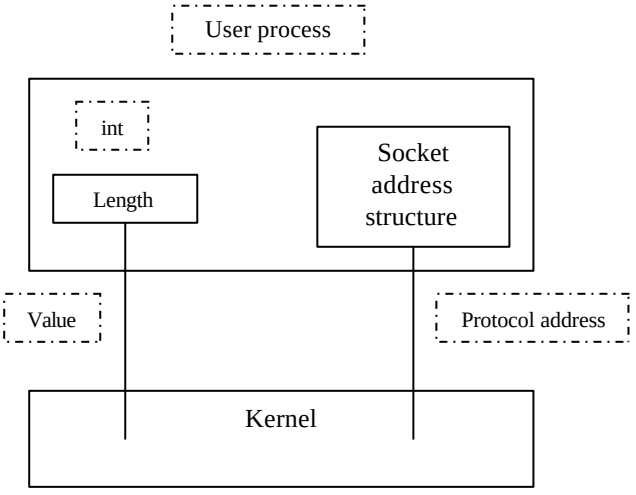
When a socket address structure is passed to any of the socket functions, it is always passed by reference, i.e. a pointer to the structure is passed. In addition to it, the length of the structure is also passed as an argument. But the way of passing the length depends on the direction in which the structure is passed; either fro the process to the kernel or vice-versa.

1. The three functions *bind, connect, & sendto* pass a socket address structure from the process to the kernel. One of the arguments to these functions is "the pointer to the socket address structure" and another is "the integer size of the structure":

```
struct sockaddr_in serv;

/* fill in serv{} */
connect (sockfd, (SA *) &serv, sizeof(serv));
```

Since both the pointer and the size of the structure is passed to the pointer, the kernel knows exactly how much of data is to be copied from the process into the kernel.



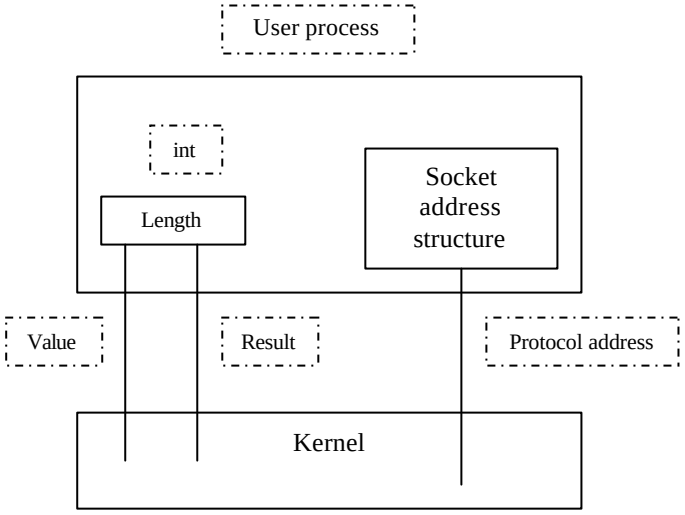
Socket address structure passed from process to kernel

2. The four functions *accept, recvfrom, getsockname, & getpeername* pass a socket address structure from the kernel to the process. The arguments to these functions are "the pointer to the socket address structure" and "the pointer to an integer containing the size of the structure":

```
struct sockaddr_un cli;          /* Unix domain */
socklen_t        len;

len = sizeof(cli);               /* len is a value */
getpeername(unixfd, (SA *) &cli, &len);
/* len may have changed */
```

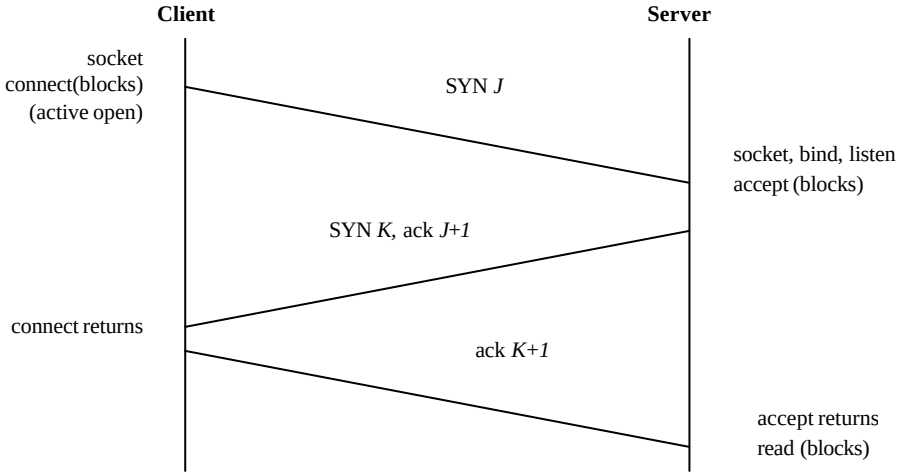
The size changes from an integer to be a pointer to an integer is because the size is both a *value* when the function is called, whereas it is a *result* when the function returns. Such type of argument is called a *value-result* argument.



Socket address structure passed from kernel to process

Three-Way Handshake: when a TCP connection is established, the following scenario occurs:

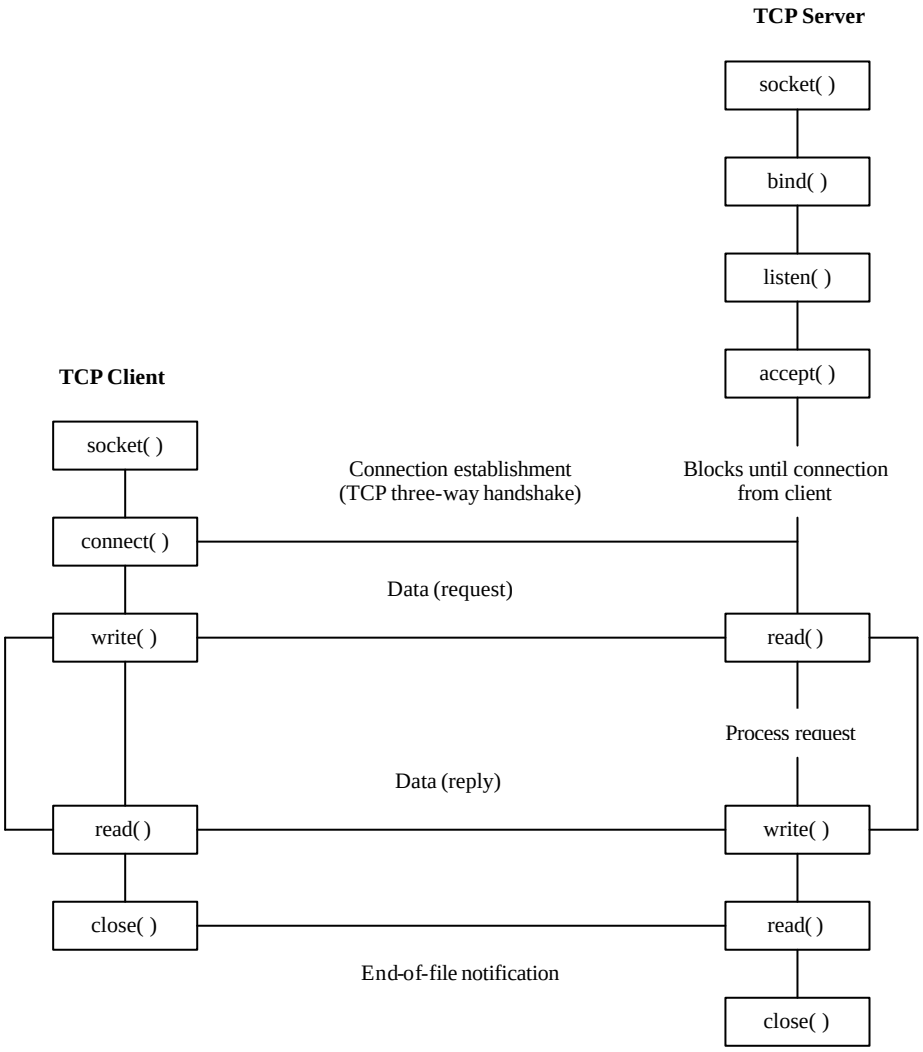
1. The server must be prepared to accept an incoming connection. This is normally done by called *socket, bind & listen*. It is known as a *passive open*.
2. The client issues an *active open* by calling *connect*. This causes the client TCP to send a SYN segment (synchronize) to tell the server about the client's initial sequence number for the data. Normally, there is no data sent with the SYN; it just contains an IP header, a TCP header, and possible TCP options.
3. The server must acknowledge the client's SYN and the server must also send its own SYN containing the initial sequence number. The sever sends its SYN and the ACK of the client's SYN in a sing segment.
4. The client must acknowledge the server's SYN.



TCP three-way handshake

Elementary TCP Sockets:

One of the basic component for writing a complete TCP client and server is the elementary socket function. The figure below shows the interaction between TCP client and server. At first, the server is started, and then sometime later a client is started that connects to the server. It is assumed that the client sends a request to the server, the server processes the request, and the server sends back a reply to the client. This continues until the client closes its end of the connection, which sends an end-of-file notification to the server. Finally the server closes its end of the connection and either terminates or waits for a new client connection.



Socket functions for elementary TCP client-server

socket function:

In order to perform I/O, the first thing a process must do is call the *socket* function, specifying required type of communication protocol. The *family* specifies the protocol family, where the constants are shown in the table. Similarly, the constants for socket *type* are also below. Normally, the *protocol* argument to the socket function is set to 0 except for raw sockets.

```
#include <sys/socket.h>
int socket(int family, int type, int protocol);
```

Family	*Description*
AF_INET	IPv4 protocols
AF_INET6	IPv6 protocols
AF_LOCAL	Unix domain protocols

Protocol family constants for socket function

Type	*Description*
SOCK_STREAM	Stream socket
SOCK_DGRAM	Datagram socket
SOCK_RAW	Raw socket

Type of socket for socket function

connect Function:

It is used by a TCP client to establish a connection with a TCP server.

```
#include <sys/socket.h>
int connect(int sockfd, const struct sockaddr *servaddr, socklen_t addrlen) ;
```

where, *sockfd* is a socket descriptor, which is returned by the socket function. The second and third arguments are a pointer to a socket address structure, and its size. The socket address structures must contain the IP address and port number of the server. The client does not have to call *bind* before calling *connect*. If required, the kernel will choose both temporary port and the source IP address. In TCP socket, the *connect* function initiates TCP's three-way handshake. The function returns only when the connection is established or an error occurs.

bind Function:

It assigns a local protocol address to a socket. The protocol address is the combination of either a 32-bit IPv4 address or a 128-bit IPv6 address with a 16-bit TCP or UDP port number.

```
#include <sys/socket.h>
int bind(int sockfd, const struct sockaddr *myaddr, socklen_t addrlen) ;
```

where, second argument is a pointer to a protocol-specific address and the third argument is the size of this address structure. With TCP, *bind* function enables to specify a port number, an IP address, both or neither.

- Serves bind their well-known port when they start. If a TCP client or server does not do this, the kernel chooses a temporary port for the socket when either *connect* or *listen* is called.
- A process can *bind* a specific IP address to its socket. The IP address must belong to an interface on the host. For a TCP client, it assigns the source IP address that will be used for IP datagrams sent on the socket. Whereas for a TCP server, the process restricts the socket to receive incoming client connections destined only to that IP address.

listen Function:

This function is normally called after both the *socket* and *bind* functions, but is necessary to be called before the *accept* function.

```
#include <sys/socket.h>
int listen(int sockfd, int backlog) ;
```

It is called only by a TCP server, which basically performs two actions:

1. When a socket is created by the *socket* function, it is assumed to be an active socket, i.e. a client socket that issues *connect*. The *listen* function converts an unconnected socket into a passive socket, indicating that the kernel should accept incoming connection requests directed to this socket.
2. The second argument specifies the maximum number of connections that the kernel should queue for this socket.

accept Function:

It is called by a TCP server to return the next completed connection from the front of the completed connection queue. It the completed connection queue is empty, the process is put to sleep.

```
#include <sys/socket.h>
int accept(int sockfd, struct sockaddr *cliaddr, socklen_t *addrlen);
```

The *cliaddr & addrlen* arguments are used to return the protocol address of the connected peer process (client). *addrlen* is a value-result argument.

fork and exec Functions:

The *fork* function is used to create a new process in UNIX. This function is called once but it returns twice. *fork* returns once in the calling process i.e. parent with a return value that is the process ID of the newly created process i.e. child. It also returns once in the child, with a return value of 0. Hence the return value tells the process whether it is the parent of the child. The *fork* returns 0 in the child instead of the parent's process IP is because a child has only one parent and it can always obtain the parent's process ID by calling *getppid*. Whereas, a parent can have any number of children, and it is not possible to obtain the process ID of its children. If the parent wants to keep track of the process IDs of its children, it must record the return values from *fork*.