UNIT 4 NETWORK, TRANSPORT AND APPLICATION LAYER

Structure **Page Nos.**

4.0 INTRODUCTION

This unit covers three important layers in networking, namely: the network, transport and the application layers. Important algorithms and mechanisms related to each layer are introduced. We will also discuss certain types of network devices.

4.1 OBJECTIVES

After going through this unit, you should be able:

- to explain the difference between the network, transport and application layers, and
- to make use of the routing and congestion algorithms and explain transport control mechanisms.

4.2 NETWORK LAYER

The network layer provides services to the transport layer. It can be based on either virtual circuits or datagrams. In both cases, its main job is routing packets from the source to the destination. In virtual circuit subnets, a routing decision is made when the virtual circuit is set up. In datagram subnets, it is made on every packet.

The network layer services have been designed with the following goals:

- The services should be independent of the subnet technology.

- The transport layer should be shielded form the number type and topology of the subnets present.
- The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

The discussion centers on the question of whether the network layer should provide connection – oriented service or connectionless service.

4.2.1 Routing Algorithms

Many routing algorithms are used in computer networks. Static algorithms include shortest path routing, flooding, and flow – based routing. Dynamic algorithms include distance vector routing and link state routing. Most actual networks use one of these. Other important routing topics are hierarchical routing, routing for mobile hosts, broadcast routing, and multicast routing.

The function of the network layer is routing packets from the source machine to the destination machine. In most subnets, packets will require multiple hops to reach the destination. The only notable exception is for broadcast networks, but even here routing is an issue if the source and destination are not on the same network. The algorithms that choose the routes and the data structures that they use are a major area of network layer design.

The routing algorithm is that part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on.

Regardless of whether routes are chosen independently for each packet or only when new connection are established, there are certain properties that are desirable in a routing algorithm: correctness, simplicity, robustness, stability, fairness and optimality.

Stability is an important goal for the routing algorithm. Routing algorithms can be grouped into two major classes: *non-adaptive* and *adaptive*. Non-adaptive algorithms do not base their routing decisions on measurements or estimates of the current traffic and topology. Instead, the choice of the route to use to get form I to J is computed in advance, off-line, and downloaded to the routers when the network is booted. This procedureis sometimes called static routing.

Adaptive algorithms, in contrast, change their routing decisions to reflect changes in the topology, and usually the traffic as well. Adaptive algorithms differ in where they get their information (e. g., every sec, when the load changes, or when the topology changes), and what metric (measure) is used for optimisation (e.g., distance, number of hops, or estimated transit time). In the following sections we will introduce a variety of routing algorithms, both static and dynamic.

Shortest Path Routing

Let us begin our study of routing algorithms with a technique that is widely used in many forms because it is simple and easy to understand. The idea is to build a graph of the subnet, with each node of the graph representing a router and each arc of the 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. One-way of measuring path length is the number of hops.

Flooding

Another static algorithm is flooding, in which every incoming packet is sent out on every outgoing line except the one it arrived on. Flooding obviously generates vast numbers of duplicate packets, in fact an infinite number unless some measure are taken, but it is one of the simplest routing algorithm.

Routers need to communicate with other routers so they can exchange routing information. Most network operating systems have associated routing protocols which support the transfer of routing information. Typical routing protocols and their associated network protocol stakes are:

- BGP (Border Gateway Protocol) – TCP/IP.
- EGP(Exterior Gateway Protocol)-TCP/IP.
- IS-IS (Immediate System to Intermediate Systems)-DECnet, OSI.
- NLSP (NetWare Link state Protocol) – Net Ware 4.1.
- OSPF (Open Shortest Path First) –TCP/IP.
- RIP (Routing Information Protocols) – XNS, Net Ware, TCP/IP.
- RTMP (Routing Table Maintenance Protocol) – Apple Talk.

4.2.2 Congestion Control Algorithms

When too many packets are present in the subnet, the performance degrades. This situation is called congestion.

Subnet can become congested, increasing the delay and lowering the throughput for packets. Network designers attempt to avoid congestion by proper design. Techniques include traffic shaping, flow specifications, and bandwidth reservation. If congestion does occur, it must be dealt with. Choke packets can be sent back, load can be shed, and other methods applied.

When the number of packets dumped into the subnet by the hosts is within its carrying capacity, they are all delivered and it is proportional to the number sent. However, as traffic increases too far, the routers are no longer able to cope, and they begin losing packets. This tends to make matters worse. At very high traffic, performance collapses completely, and almost no packets are delivered.

Congestion can be brought about by several factors.

- If all of a sudden, streams of packets begin arriving on three or four input lines and all need the same output line, a queue will build up. If there is insufficient memory to hold all of them, packets will be lost. If router have an infinite amount of memory, congestion gets worse, not better, because by the time packets get to the front of the queue, they have already timed out and duplicates have been sent. All these packets will be dutifully forwarded to the next router, increasing the load all the way to the destination.
- Slow processors can also cause congestion. If the routers; CPUs are slow at performing the bookkeeping tastes required of them, queues can build up, even though there is excess line capacity. Similarly, low – bandwidth (equivalents low data rate) lines can also cause congestion.
- Congestion control has to do with making sure the subnet is able to carry the offered traffic. It is a global issue, involving the behaviour of all the hosts, all the routers the store and forwarding processing within the routers, and all the other factors that tend to diminish the carrying capacity of the subnet.

4.2.3 Comparison of Virtual Circuit and Datagram Subnets

Here, first explain virtual circuit and datagram concepts using figures.
Inside the subnet, several trade-offs exist between virtual circuits and datagrams.

- One trade-off is between router memory space and bandwidth. Virtual circuits allow packets to contains circuit numbers instead of full destination addresses. If the packets tend to be fairly short, a full destination address in every packet may represent a significant amount of overhead, and hence wasted bandwidth. The price paid for using virtual circuits internally is the table space within the routers. Depending upon the relative cost of communication circuits versus router memory, one or the other may be cheaper.
- Another trade-off is setup time versus address parsing time. Using virtual circuits requires a setup phase, which takes time and consumes resources. However, figuring out what to do with a data packet in a virtual circuit subnet is easy: the router just uses the circuit number to for a table lookup to find out w here the packet goes. In datagram subnet, a more complicated procedure is required to determine where the packet goes.
- Virtual circuits have some advantages in avoiding congestion within the subnet because resources can be reserved in advance, at the time of connection establishment. Once the packets start arriving, the necessary bandwidth and router capacity will be there. With a datagram subnet, congestion avoidance is more difficult.
- For transaction processing systems, the overhead is required to set up and clear a virtual circuit. If the majority of the traffic is expected to be of this kind, the use of switched virtual circuits inside the subnet makes little sense. On the other hand, permanent virtual circuits, which are set up manually and last for months or years, may be useful here.
- Virtual circuits also have a vulnerability problem. If a router crashes and loses its memory, even if it comes back up a second later, all the virtual circuits passing through it will have to be aborted. In contrast, if a datagram router goes down, only those users whose packets were queued up in the router at the time will suffer, and may be not even all those, depending upon whether they have already been acknowledged or not. The loss of communication line is fatal to virtual circuits using it but can be easily compensated for if datagrams are used.
- Datagrams also allow the routers to balance the traffic throughout the subnet, since routes can be changed during the transaction halfway through a connection.
- It is worth explicitly pointing out that the service offered is a separate issue from the subnet structure. In theory, all four combinations are possible. Obviously, a virtual circuits implementation of a connection-oriented service and datagram implementation of a conn ectionless service are reasonable. Implementing connections using datagrams also makes sense when the subnet is trying to provide a highly adaptive service.

4.2.4 Internetworking

In internetworking modes are connected together through different network equipments. Networks connect to other networks through repeaters, bridges or routers. A repeater corresponds to the physical layer and always routes signals form one networks segment to another. Bridges route using the data link layer and routers route using the network layer. Networks differ in various ways, so when multiple networks are connected together problems can occur. Sometimes the problems can be overcome by tunnelling a packet (where original packet is put inside another packet) through a hostile network, but if the source and destination networks are different, this approach fails. When different networks have different maximum packet sizes, fragmentation may be called for.

Repeaters

All types of network connections suffer form attenuation an d pulse distortion. For a given cable specification and bit rate, each has a maximum length of cable. Repeaters can be used to increase the maximum interconnection length and will do the following:

- Clean signal pulses.
- Pass all signals between attached segments.
- Boost signal power.
- Possibly translate between two different media types (e.g., fiber – optic to twisted – pair cable).

Bridges

Bridges filter input and output traffic so that only packets intended for a network are actually routed into the network and only packet intended for the outside are allowed out of the network.

Routers

Routers examine the network address field and determine the best route for the packet. They have the great advantage that they normally support several different types of network layer protocol.

Routers, which only read one type of protocol, will normally have high filtering and forwarding rates. If they support multiple protocols then there is normally an overhead in that the router must detect the protocol and look in the correct place for the destination address.

4.3 TRANSPORT LAYER

Transport layer provides reliable cost effective, data transport from the source machine to destination machine.

4.3.1 Transport Service and Mechanism

Transport layer provides various services, the most important of which is an end to end, reliable, connection-oriented byte stream from sender to receiver. It is accessed through service primitives that permit the establishment, use and release of connections.

Transport protocols must be able to do connection management over unreliable networks. Connection establishment is complicated by the existence of delayed duplicate packets that can reappear at inopportune moments. To deal with them, three-way handshakes are needed to establish connections. Releasing a connection is easier than establishing one but is still far from trivial.

4.3.2 Types of Service (ToS)/Quality of Service (QoS)

The need to define quality of service arises form the realisation that users require different quality presentations at different times. The different quality presentations map onto different parameter values. When a multimedia presentation is transmitted via a network, it translates into different requirements of network performance.

To be able to specify QoS aspects concisely and to request them of a network, QoS must be specified as a set of parameters that can be assigned numerical values. In a multimedia presentation, the ultimate user of the systems is a human being. Thus, the quality of the presentation is a matter of the user's perception, which is limited by the response of the human vision and auditory senses. This perceptual nature of QoS makes it subjective and difficult to quantify precisely. Thus, it is easier to specify a range of values rather than a single value.

The concept of quality of service has evolved over the years. It was originally developed by ISO in the 1980s. The QoS concept developed by ISO had a rather restricted perspective. It focused mainly on the SAS factors for the application and the NPPs for the network. More recently, a wider view has been developed to describe QoS for multimedia applications.

4.3.3 Transport Control Mechanism

The transport control service is implemented by a transport protocol used between the two transport entities. It is similar to the Data Link Protocol, but with some differences:

- Environments in which they operate. At the data link layer two router communicate directly via a physical channel, whereas at the transport layer, this physical channel is replaced by the entire subnet.
- In Data Link Protocol, it is not necessary for a router to specify which router to talk to. Each outgoing line uniquely specifies a particular route.

Addressing

When an application process wishes to set up a connection to a remote application process, it must specify which one to connect to. In Internet, these end points are (IP address + Local Port) pairs. The end point in this context are:

- **TSAP** (Transport Service Access Point)
- **NSAP** (Network Service Access Point)

A transport entity support multiple TSAPs.

Flow Control and Buffering

Flow Control of transport layer is similar to that of Data Link Layer, but in Transport layers the number of connections open is numerous as compared to Data Link Layer.

If the subnet provides datagram service, the sending transport entity must also be buffered, for re-transmitting in the case of loss. If the receiver knows that the sender buffer all TPDUs (Transport Protocol Data Unit) until they are acknowledged, the receiver may or may not dedicate specific buffers, to specific connections.

In summary, if the network service is unreliable, the sender must buffer all TPDUs. However, with reliable network services, other trade-off becomes possible.

The optimum trade-off between source buffering and destination buffering depends on the type of traffic carried by the connection. For low-bandwidth bursty traffic, such as that produced by an interactive terminal, it is better not to dedicate any buffer, but rather to acquire them dynamically at both ends.

Multiplexing

Multiplexing is putting multiple things on to one resource i.e., multiplexing several conversations onto connections, virtual circuits, and physical links plays a role in several layers of the network architecture.

Need for Multiplexing

- Number of virtual circuits are open by the users or one user opening more than one, which requires a lot of buffer in the router, this gives a solid reason for packet switched network.
- To bill the users based on the amount of data sent, not the connection time.

Upward Multiplexing: Multiplexing of different transport connections onto the same network connection attractive.

Downward Multiplexing: The transport layer opens multiple network connections and distributes the traffic among them on a round-robin basis.

Connection Establishment and Management

Connection Establishment is not easy as it sounds, but it is in fact a complicated task, we have to take care of the losses that occur during transmission. At first glance, it would seem sufficient for one transport entity to just send a CONNECTION REQUEST TPDU to the destination and wait for a CONNECTION ACCEPTANCE reply. The problem occurs when the network is not reliable.

Solution could be to give each connection a connection identifier, chosen by the initiating party, and put it in each TPDU, including the one requesting the connection. After each connection is released, each transport entity could update a table listing absolute connection as (peer transport entity, connection identifier) pair.

Unfortunately, this scheme has a basic flaw: it requires each transport entity to maintain a certain amount of history information indefinitely. If a machine crashes and looses its memory, it will no longer know which connection identifiers have already been used.

Crash Recovery

If host and routers are subject to crashes, recoveries from these crashes become an issue. The transport entity is entirely within the hosts and recovery from network and router crashes can be effectively implemented. If the network layer provides datagram services, the transport entity expects lost TPDUs all the time and knows how to cope with them. If the network layer provides connection-oriented service, then loss of virtual circuits is handled by establishing a new one and then probing the remote transport entity to ask it which TPDUs it has received and which one it has not received.

At first glance it would seen obvious: The client should retransmit only if it has an acknowledged TPDU outstanding when it learns of the crash. However, a closer inspection reveals difficulties with this naive approach.

4.3.4 TCP/UDP

In this section we will discuss two important transport layer protocols: TCP and UDP.

Transmission Control Protocol (TCP)

TCP provides a highly reliable, connection oriented, end-to-end transport service between processes in end systems connected to the subnet. TCP assumes that the layer below offers an unreliable datagram service. TCP provides the types of facility associated with the ISO Class 4 transport service, including error recovery, sequencing of packets, flow control by the windowing method, and the support of multiplexed connections from the layer above.

Format of TCP Header

The sender's TCP layer communicates with the receiver's TCP layer using the TCP protocol data unit. It defines parameters such as the source port, destination port, sequence number and so on. It is described below:

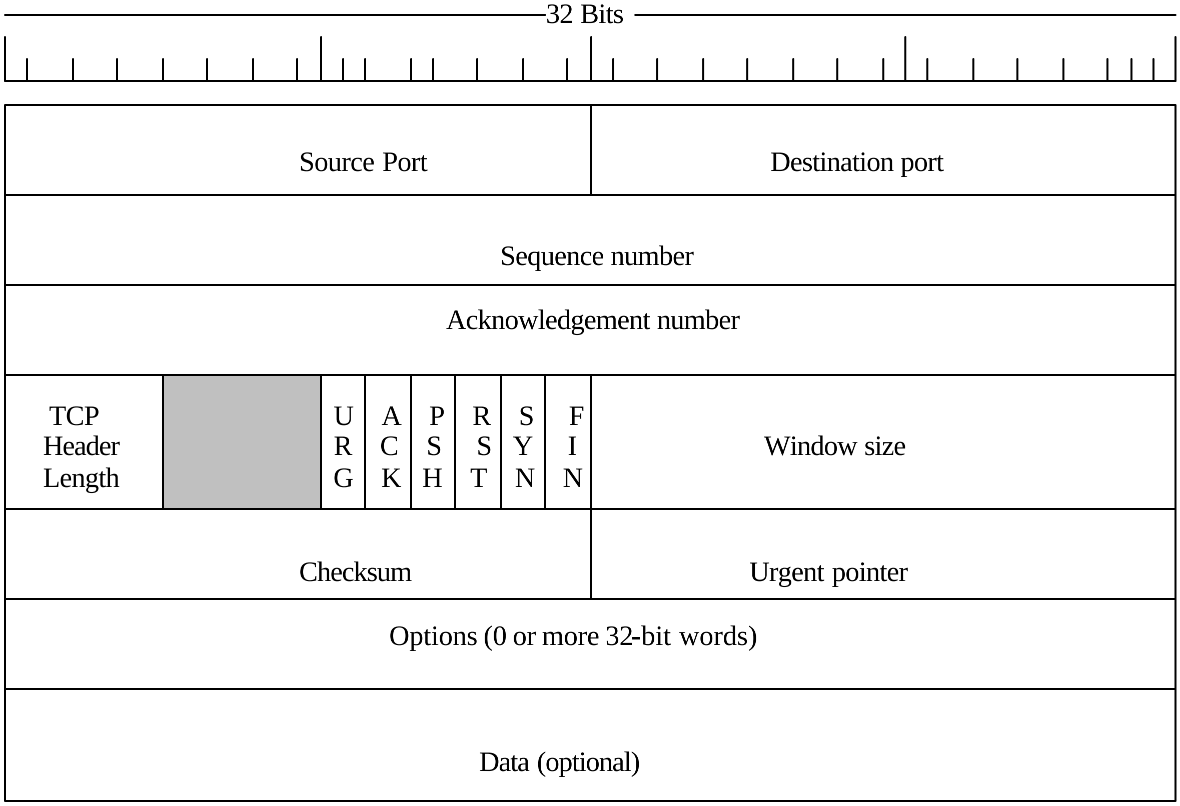

Figure 1: TCP Segment

- Source and destination port number – which are 16 bit values to identify the local port number.
- Sequence number – which identifies the current sequence number of the data segment. This allows the receiver to keep track of the data segments received. Any segments that are missing can be easily identified.
- Data offset – which is a 32-bit value and identifies the start of the data.
- Flags the flag field is defined as UAPRSE, where U is the urgent Flag, A the acknowledgment flag, P the push function, R the reset flag, S the sequence synchronise flag and F the end – of transmission flag.
- Windows – which is a 16 bit value and gives the number of data blocks that the receiving host can accept at a time.
- Checksum – which is a 16 bit checksum for the data and header.
  UrgPtr – which is the urgent pointer and is used to identify an important area of data.

User Data Protocol (UDP)

The Internet protocol suite also supports a connectionless transport protocol, UDP (User Data Protocol). UDP provides a way for applications to send encapsulated raw IP datagrams and send them without having to establish a connection. Many client-server applications that have one request and one response use UDP rather than go to the trouble of establishing and later releasing a connection.

A UDP segment consists of an 8-byte header followed by the data. The two ports serve the same function as they do in TCP: to identify the end points within the source and destination machines. The UDP length field includes the 8-byte header and the data. The UDP checksum includes the same format pseudo-header, the UDP header, and the UDP data, padded out to an even number of bytes if need be. It is optional and stored, as 0 if not computed.

4.4 APPLICATION LAYER

The application layer contains a variety of protocols that are commonly needed. For example, there are hundreds of incompatible terminal types in the world. To solve this problem there is a need to define a abstract network virtual terminal that editors and other programs can be written to deal with.

Another application layer function is file transfer. Different file systems have different file naming conventions.

4.4.1 The Domain Name System (DNS)

DNS is a scheme for assigning meaningful high - level name to a large set of machines, and discusses a mechanism that maps between high-level machine names and IP addresses. It considers both the translation from high-level name to IP addresses and the translation form IP addresses to high-level machine names. It has been used to assign machine names throughout the global Internet. It uses a geographically distributed set of servers to map names to addresses, the implementation of the name mapping mechanism provides a large scale example of the client server paradigm.

In a TCP/IP internet, hierarchical machine names are assigned according to the structure of organisations that obtain authority for parts of the namespace, not necessarily according to the structure of the physical network interconnections.

4.4.2 TCP/IP Internet Domain Name

The mechanism that implements a machine name hierarchy for TCP/IP Internets is called the Domain Name Systems (DNS). DNS has two, conceptually independent aspects. The first is abstract; it specifies the name syntax and rules for delegating authority over names. The second is concrete; it specifies the implementation of a distributed computing system that efficiently maps names to addresses. This section considers the name syntax, and later sections examine the implementation.

The domain name system uses a hierarchical naming scheme as domain names. As in our earlier examples, a domain name consists of a sequence of sub names separated by a delimiter character, the period. In our examples we said that individual sections of the name might represent sites or groups, but the domain system simply calls each section a label. Thus, the domain name

cs.ignou.org

Contains three labels: CS, IGNOU, and ORG. Any suffix of a labels in a domain name is also called a domain. In the above example the lowest level domain is CS. ignou.org (the domain name for the Computer Science Department at ignou), the second level domain is ignou.org (the domain name for ignou) and the top level domain is org, as the example shows, domain names are written with the local label first and the top domain last. As we will see, writing them in this order makes it possible to compress messages that contain multiple domain names.

Domain Name	Meaning
com	Commercial Organisations
edu	Educational Institutes
gov	Government Institutions
mil	Military Groups
net	Major network Support centers
org	Organisations other than those above
arpa	Temporary ARPANET domain (obsolete)
int	International Organisations
country code e.g. “in”	Each country (geographic scheme)

4.4.3 Electronic Mail

This is the most widely used servic e facilitating users to send and receive messages electronically in a store and forward manner. Different E-mail standards, viz., SMTP, UUCP and X400 Message Handling system are supported on networks (e.g. ERNET).

Electronic Mail is a system whereby a computer user can exchange messages with other computer users or group of users via a communications network.
The backbone of an electronic mail system is a communication network that connects remote terminals to a central system or a local area network that interconnects personal computers. Users can send mail to a single recipient or they can broadcast it to any number of selected users on the systems. The method for receiving mail depends on the sophistication of the system. When multitasking personal computer and workstation are used mail can be delivered to users while they are working on something else. Otherwise, users have to interrogate their mailboxes in a central systems, or file server.

Many users first encounter computer networks when they send or receive electronic mail to or from a remote site. E-mail is the most widely used application service. Indeed, many computer users access networks only through electronic mail.

E-mail is popular because it offers a fast, convenient method of transferring information. E-mail can accommodate small notes or large voluminous memos with a single mechanism. It should not surprise you to learn that more users send files with electronic mail than with file transfer protocols.

Characteristics:

- Store and forward
- Delivery time ranging from few seconds to hours
- Largely textual
- Binary files may be appended or “uuencoded”
- Multimedia (“mime” Standard)
- Distribution lists, with “cc:”, “bcc:”, “fcc:”
- Mail forwarding
- Auto-processing
- Statistics collection
- Secure email
- Several mailers: smtp, uucp (smtp requires IP connectivity; uucp works with dial-up)

4.4.4 WWW (World Wide Web)

The world Wide Web is a system for linking up hypertext documents. Each document is a page written in HTML, possible with hyperlinks to other documents. A browser can display a document by establishing a TCP connection to its server, asking for the document, and then closing the connection. When a hyperlink is selected by the user, that document can also be fetched in the same way. This manner, documents all over the world are linked together in a giant web. Given below are some facts about WWW:

- Fastest growing discovery and retrieval system
- Presently 10,000 servers, growing at an astounding rate
- Retrieve "hypermedia" documents, with text, graphics, audio, video, and links to other hypermedia documents
- A navigational system based on "hyperlink"
- State-less interaction between client and server, conforming to "http" protocol.

4.4.5 Mail-based Applications

Plain-old mail
Notices
Auto-save and processing
News dissemination through LISTSRV
Archival search and retrieval
Access to network-wide news (bulletin boards)

4.5 REMOTE PROCEDURE CALL (RPC)

The designers chose to build three independent pieces; the NFS protocol itself, a general purpose Remote Procedure Call (RPC) mechanism, and a general purpose External Data Representation (XDR). Their intent was to separate the three to make it possible to use RPC and XDR in other software, including application programs as well as other protocols. For example, a programmer can divide a program into a client side and a server side that use RPC as the chief communication mechanism. One of the client side, the programmer designates some procedures as remote, forcing the compiler to incorporate RPC code into those procedures. On the server side, the programmer implements the desired procedures and uses other RPC facilities to declare them to be part of a server. When the executing client program calls one of the remote procedures, RPC automatically collects values for arguments, from a message, sends the message to the remote server, awaits a response, and stores returned values in the designated arguments. In essence, communication with the remote server occurs automatically as a side-effect of a remote call. The RPC mechanism hides all the details of protocols, making it possible for programmers who know little about the underlying communication protocols to write distributed programs.

4.6 FILE TRANSFER PROTOCOL (FTP)

FTP (File Transfer Protocol) is the primary method of transferring files over Internet. "FTP" transfers files to and from a remote network site. Some sites maintain Anonymous accounts on the system for retrieval of public domain software's stored on the system.

The *ftp* protocol is used to access files by FTP, the Internet's file transfer protocol. FTP has been around more than two decades and is well entrenched Numerous FTP servers all over the world allow people anywhere on the internet to log in and download whatever files have been placed on the FTP server. The Web does not change this; it just makes obtaining files by FTP easier, as FTP has a somewhat arcane interface.

4.7 TELNET

Telnet is a program that allows you to establish a virtual terminal connection between two machines using TCP /IP. For this you must have its internet address or host name of computer.

☞ **Check Your Progress 1**

1) What is the difference in virtual circuit and Datagram subnets?

……………………………………………………………………………………..

……………………....…………………………………………………………………

……………………………………....……………………………………………………

………………………………………………………....…………………………………

2) Distinguish between non-adaptive and adaptive algorithms.

……………………………………………………………………………………..

……………………....…………………………………………………………………

……………………………………....……………………………………………………

………………………………………………………....…………………………………

3) What are the factors in network which could cause congestion?

……………………………………………………………………………………..

……………………....…………………………………………………………………

……………………………………....……………………………………………………

………………………………………………………....…………………………………

4) Explain functionality of
(i) Routers
(ii) Bridges
(iii) Repeaters

……………………………………………………………………………………..

……………………....…………………………………………………………………

……………………………………....……………………………………………………

………………………………………………………....…………………………………

5) Distinguish software upward and downward multiplexing.

……………………………………………………………………………………..

……………………....…………………………………………………………………

……………………………………....……………………………………………………

………………………………………………………....…………………………………

4.8 SUMMARY

In this unit you have been introduced to the network, transport and application layers, their features, services offered by them and the algorithms used by them. Other concepts covered include Internetworking, repeaters, routes, bridges, multiplexing, addressing and transport control mechanisms. Standards and definitions of commonly used terms in Application layer are covered briefly to familiarise you with current trends.

4.9 SOLUTIONS/ANSWERS

1) Refer to Section 4.2.3.

2) Non-adaptive algorithms do not base their routing decisions on measurements of current traffic. Instead choice of routes are computed in advance and downloaded to the routes when the network is booted. Adaptive algorithms change their routing decisions to reflect changes in the topology.

3) Many steam of packets arrive on more than input lines and need the same output line.

   Slow processes cause congestion.

   Low bandwidth lines.

   Subnet is unable to carry the offered load.

4) Please see sub-sections 4.4.1 to 4.4.3

5) Please see sub-section 4.3, 3.3