Subject: Computer Networks
Subject Code: 17MCA31
Year: Dec.2018/Jan.2019

Q1(a) **Define Computer Networks. Explain the uses of Computer Networks in business application an
home applications.**

A computer network is a group of computer systems and other computing hardware devices that ar
linked together through communication channels to facilitate communication and resource-sharin
among a wide range of users. Networks are commonly categorized based on their characteristics.

Business Applications:-

For smaller companies, all the computers are likely to be in a single office or perhaps a single buildin
but for larger ones, the computers and employees may be scattered over dozens of offices and plants i
many countries. Nevertheless, a sales person in New York might sometimes need access to a produc
inventory.

database in Singapore. Networks called VPNs (Virtual Private Networks) may be used to join th
individual networks at different sites into one extended network. In other words, the mere fact that a use
happens to be 15,000 km away from his data should not prevent him from using the data as though the
were local. This goal may be summarized by saying that it is an attempt to end the ‘‘tyranny o
geography.’’

In the simplest of terms, one can imagine a company’s information system as consisting of one or mor
databases with company information and some number of employees who need to access them remotel
In this model, the data are stored on powerful computers called servers. Often these are centrally house
and maintained by a system administrator. In contrast, the employees have simpler machines, calle
clients, on their desks, with which they access remote data, for example, to include in spreadsheets the
are constructing. (Sometimes we will refer to the human user of the client machine as the ‘‘client,’’ but
should be clear from the context whether we mean the computer or its user.)

A second goal of setting up a computer network has to do with people rather than information or eve
computers. A computer network can provide a powerful communication medium among employee
Virtually every company that has two or more computers now has email (electronic mail), whic
employees generally use for a great deal of daily communication. In fact, a common gripe around th
water cooler is how much email everyone has to deal with, much of it quite meaningless because bosse
have discovered that they can send the same (often content-free) message to all their subordinates at th
push of a button. Telephone calls between employees may be carried by the computer network instead o
by the phone company. This technology is called IP telephony or Voice over IP (VoIP) when Interne
technology is used. The microphone and speaker at each end may belong to a VoIP-enabled phone or th
employee’s computer. Companies find this a wonderful way to save on their telephone bills.

A third goal for many companies is doing business electronically, especially with customers an
suppliers. This new model is called e-commerce (electronic commerce) and it has grown rapidly i
recent years. Airlines, bookstores, and other retailers have discovered that many customers like th
convenience of shopping from home. Consequently, many companies provide catalogs of their good
and services online and take orders online. Manufacturers of automobiles, aircraft, and computer
among others, buy subsystems from a variety of suppliers and then assemble the parts. Using compute
networks, manufacturers can place orders electronically as needed. This reduces the need for larg
inventories and enhances efficiency.

Home Applications:-

Internet access provides home users with connectivity to remote computers. As with companies, hom
users can access information, communicate with other people, and buy products and services with e
commerce. The main benefit now comes from connecting outside of the home. Bob Metcalfe, th
inventor of Ethernet, hypothesized that the value of a network is proportional to the square of the numbe
of users because this is roughly the number of different connections that may be made (Gilder, 1993
This hypothesis is known as ‘‘Metcalfe’s law.’’ It helps to explain how the tremendous popularity of th
Internet comes from its size.

Peer-to-peer communication is often used to share music and videos. It really hit the big time aroun

2000 with a music sharing service called Napster that was shut down after what was probably the bigges copyright infringement case in all of recorded history (Lam and Tan, 2001; and Macedonia, 2000). Lega applications for peer-to-peer communication also exist. These include fans sharing public domain musi families sharing photos and movies, and users downloading public software packages. In fact, one of th most popular Internet applications of all, email, is inherently peer-to-peer. This form of communicatio is likely to grow considerably in the future.

The Internet can be used by applications to carry audio (e.g., Internet radio stations) and video (e.g YouTube). Besides being a cheap way to call to distant friends, these applications can provide ric experiences such as telelearning, meaning attending 8 A.M. classes without the inconvenience of havin to get out of bed first. In the long run, the use of networks to enhance human-to-human communicatio may prove more important than any of the others. It may become hugely important to people who ar geographically challenged, giving them the same access to services as people living in the middle of big city.

Between person-to-person communications and accessing information are social network application Here, the flow of information is driven by the relationships that people declare between each other. On of the most popular social networking sites is Facebook. It lets people update their personal profiles an shares the updates with other people who they have declared to be their friends. Other social networkin applications can make introductions via friends of friends, send news messages to friends such as Twitte above, and much more

Q1(b) **Explain Local Area Network in detail.**

The next step up is the LAN (Local Area Network). A LAN is a privately owned network that operate within and nearby a single building like a home, office or factory. LANs are widely used to conne personal computers and consumer electronics to let them share resources (e.g., printers) and exchang information. When LANs are used by companies, they are called enterprise networks. Wireless LAN are very popular these days, especially in homes, older office buildings, cafeterias, and other plac where it is too much trouble to install cables. In these systems, every computer has a radio modem an an antenna that it uses to communicate with other computers. In most cases, each computer talks to device in the ceiling as shown in Fig. 1-8(a). This device, called an AP (Access Point), wireless route or base station, relays packets between the wireless computers and also between them and the Interne Being the AP is like being the popular kid as school because everyone wants to talk to you. However, other computers are close enough, they can communicate directly with one another in a peer-to-pee configuration.

There is a standard for wireless LANs called IEEE 802.11, popularly known as WiFi, which has becom very widespread. It runs at speeds anywhere from 11 to hundreds of Mbps. (In this book we will adhe to tradition and measure line speeds in megabits/sec, where 1 Mbps is 1,000,000 bits/sec, an gigabits/sec, where 1 Gbps is 1,000,000,000 bits/sec.)

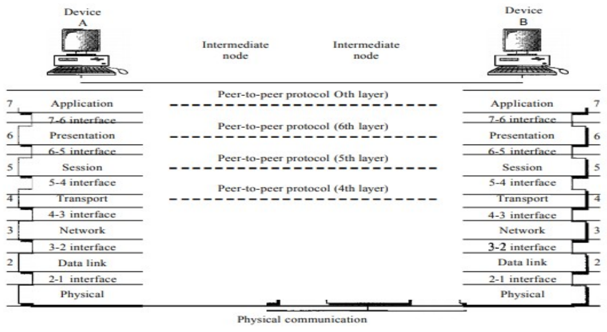
It is also possible to divide one large physical LAN into two smaller logical LANs. You might wonde why this would be useful. Sometimes, the layout of the network equipment does not match th organization's structure. For example, the engineering and finance departments of a company might hav computers on the same physical LAN because they are in the same wing of the building but it might b easier to manage the system if engineering and finance logically each had its own network Virtual LA or VLAN. In this design each port is tagged with a ''color,'' say green for engineering and red fo finance. The switch then forwards packets so that computers attached to the green ports are separate from the computers attached to the red ports. Broadcast packets sent on a red port, for example, will no be received on a green port, just as though there were two different LANs.

Both wireless and wired broadcast networks can be divided into static and dynamic designs, dependin on how the channel is allocated. A typical static allocation would be to divide time into discrete interva and use a round-robin algorithm, allowing each machine to broadcast only when its time slot comes u Static allocation wastes channel capacity when a machine has nothing to say during its allocated slot, s most systems attempt to allocate the channel dynamically (i.e., on demand).

While we could think of the home network as just another LAN, it is more likely to have differe properties than other networks. First, the networked devices have to be very easy to install. Wirele

	routers are the most returned consumer electronic item. People buy one because they want a wireles network at home, find that it does not work ‘‘out of the box,’’ and then return it rather than listen t elevator music while on hold on the technical helpline.
Q2(a)	**Describe the architecture of internet with diagram.** The architecture of the Internet has also changed a great deal as it has grown explosively. In this sectio we will attempt to give a brief overview of what it looks like today. The picture is complicated b continuous upheavals in the businesses of telephone companies (telcos), cable companies and ISPs tha often make it hard to tell who is doing what. One driver of these upheavals is telecommunication convergence, in which one network is used for previously different uses. For example, in a ‘‘triple play’ one company sells you telephony, TV, and Internet service over the same network connection on th assumption that this will save you money. Consequently, the description given here will be of necessit somewhat simpler than reality. And what is true today may not be true tomorrow. To join the Internet, the computer is connected to an Internet Service Provider, or simply ISP, from wh the user purchases Internet access or connectivity. This lets the computer exchange packets with all o the other accessible hosts on the Internet. The user might send packets to surf the Web or for any of thousand other uses, it does not matter. There are many kinds of Internet access, and they are usuall distinguished by how much bandwidth they provide and how much they cost, but the most importan attribute is connectivity. A common way to connect to an ISP is to use the phone line to your house, in which case your phon company is your ISP. DSL, short for Digital Subscriber Line, reuses the telephone line that connects t your house for digital data transmission. The computer is connected to a device called a DSL mode that converts between digital packets and analog signals that can pass unhindered over the telephone lin At the other end, a device called a DSLAM (Digital Subscriber Line Access Multiplexer) convert between signals and packets. Another method is to send signals over the cable TV system. Like DSL, this is a way to reuse existin infrastructure, in this case otherwise unused cable TV channels. The device at the home end is called cable modem and the device at the cable headend is called the CMTS (Cable Modem Terminatio System). DSL and cable provide Internet access at rates from a small fraction of a megabit/sec t multiple megabit/sec, depending on the system. These rates are much greater than dial-up rates, whic are limited to 56 kbps because of the narrow bandwidth used for voice calls. Internet access at muc greater than dial-up speeds is called broadband. The name refers to the broader bandwidth that is use for faster networks, rather than any particular speed. The access methods mentioned so far are limited by the bandwidth of the ‘‘last mile’’ or last leg o transmission. By running optical fiber to residences, faster Internet access can be provided at rates on th order of 10 to 100 Mbps. This design is called FTTH (Fiber to the Home). For businesses in commercia areas, it may make sense to lease a high-speed transmission line from the offices to the nearest ISP. Fo example, in North America, a T3 line runs at roughly 45 Mbps. Wireless is used for Internet access too. An example we will explore shortly is that of 3G mobile phon networks. They can provide data delivery at rates of 1 Mbps or higher to mobile phones and fixe subscribers in the coverage area. We can now move packets between the home and the ISP. We call th location at which customer packets enter the ISP network for service the ISP’s POP (Point of Presence We will next explain how packets are moved between the POPs of different ISPs. From this point on, th system is fully digital and packet switched. ISP networks may be regional, national, or international i scope. We have already seen that their architecture is made up of long-distance transmission lines tha interconnect routers at POPs in the different cities that the ISPs serve. This equipment is called th backbone of the ISP. If a packet is destined for a host served directly by the ISP, that packet is route over the backbone and delivered to the host. Otherwise, it must be handed over to another ISP.
Q2(b)	**Explain OSI Reference model with diagram.** the International Standards Organization (ISO) is a multinational body dedicated to worldwid agreement on international standards. An ISO standard that covers all aspects of networ communications is the Open Systems Interconnection model. It was first introduced in the late 1970s. The OSI model is a layered framework for the design of network systems that allows communicatio between all types of computer systems. It consists of seven separate but related layers, each of whic

defines a part of the process of moving information across a network



LAYERS IN THE OSI MODEL

1) Physical Layer

The physical layer coordinates the functions required to carry a bit stream over a physical medium. I deals with the mechanical and electrical specifications of the interface and transmission medium.

It also defines the procedures and functions that physical devices and interfaces have to perform fo transmission to Occur.

Representation of bits. The physical layer data consists of a stream of bits (sequence of Os or 1s) with n interpretation. To be transmitted, bits must be encoded into signals--electrical or optical. The physica layer defines the type of encoding (how Os and Is are changed to signals).

Data rate. The transmission rate-the number of bits sent each second-is also defined by the physica layer. In other words, the physical layer defines the duration of a bit, which is how long it lasts.

Synchronization of bits. The sender and receiver not only must use the same bit rate but also must b synchronized at the bit level. In other words, the sender and the receiver clocks must be synchronized.

Physical topology. The physical topology defines how devices are connected to make a networl Devices can be connected by using a mesh topology (every device is connected to every other device), star topology (devices are connected through a central device), a ring topology (each device is connecte to the next, forming a ring), a bus topology (every device is on a common link), or a hybrid topolog (this is a combination of two or more topologies).

Transmission mode. The physical layer also defines the direction of transmission between two devices simplex, half-duplex, or full-duplex. In simplex mode, only one device can send; the other can onl receive. The simplex mode is a one-way communication. In the half-duplex mode, two devices can sen and receive, but not at the same time. In a full-duplex (or simply duplex) mode, two devices can sen and receive at the same time.

2) Data Link Layer

The data link layer transforms the physical layer, a raw transmission facility, to a reliable link. It make the physical layer appear error-free to the upper layer (network layer).

Other responsibilities of the data link layer include the following:

Framing. The data link layer divides the stream of bits received from the network layer into manageabl data units called frames. The data link layer is responsible for moving frames from one hop (node) to th next.

Physical addressing. If frames are to be distributed to different systems on the network, the data lin layer adds a header to the frame to define the sender and/or receiver of the frame. If the frame is intende for a system outside the sender's network, the receiver address is the address of the device that connect the network to the next one.

Flow control. If the rate at which the data are absorbed by the receiver is less than the rate at which dat are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelmin the receiver.

Error control. The data link layer adds reliability to the physical layer by adding mechanisms to detec and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Erro control is normally achieved through a trailer added to the end of the frame.

Access control. When two or more devices are connected to the same link, data link layer protocols ar

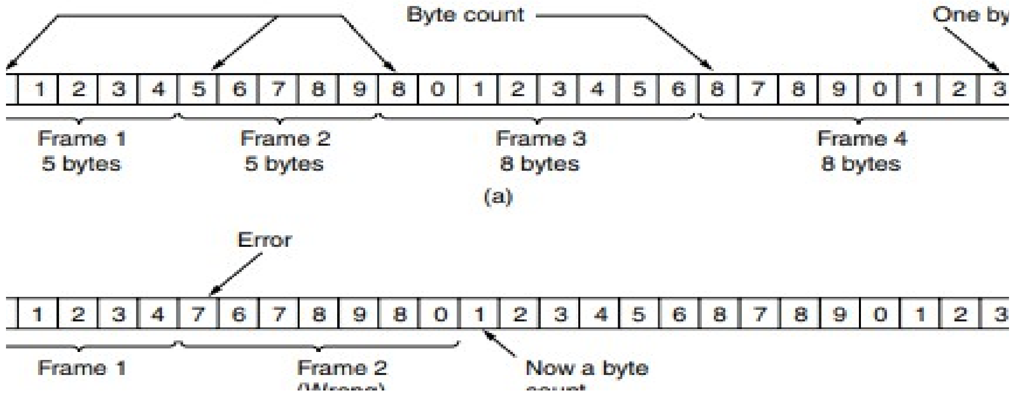
necessary to determine which device has control over the link at any given time
3)Network Layer
The network layer is responsible for the source-to-destination delivery of a packet, possibly acros multiple networks (links). Whereas the data link layer oversees the delivery of the packet between tw systems on the same network (links), the network layer ensures that each packet gets from its point c origin to its final destination.
If two systems are connected to the same link, there is usually no need for a network layer. However, the two systems are attached to different networks (links) with connecting devices between the network (links), there is often a need for the network layer to accomplish source-to-destination delivery. Figur shows the relationship of the network layer to the data link and transport layers.
Logical addressing. The physical addressing implemented by the data link layer handles the addressin problem locally. If a packet passes the network boundary, we need another addressing system to hel distinguish the source and destination systems. The network layer adds a header to the packet comin from the upper layer that, among other things, includes the logical addresses of the sender and receiver.
Routing. When independent networks or links are connected to create internetworks (network c networks) or a large network, the connecting devices (called routers or switches) route or switch th packets to their final destination. One of the functions of the network layer is to provide this mechanism
4)Transport Layer
The transport layer is responsible for process-to-process delivery of the entire message. A process is a application program running on a host. Whereas the network layer oversees source-to-destinatio delivery of individual packets, it does not recognize any relationship between those packets. It treat each one independently, as though each piece belonged to a separate message, whether or not it does The transport layer, on the other hand, ensures that the whole message arrives intact and in orde overseeing both error control and flow control at the source-to-destination level.
Service-point addressing.
Computers often run several programs at the same time. For this reason, source-to-destination deliver means delivery not only from one computer to the next but also from a specific process (runnin program) on one computer to a specific process (running program) on the other. The transport laye header must therefore include a type of address called a service-point address (or port address).
The network layer gets each packet to the correct computer; the transport layer gets the entire message t the correct process on that computer.
Segmentation and reassembly.
A message is divided into transmittable segments, with each segment containing a sequence numbe These numbers enable the transport layer to reassemble the message correctly upon arriving at th destination and to identify and replace packets that were lost in transmission.
Connection control.
The transport layer can be either connectionless or connection oriented. A connectionless transport laye treats each segment as an independent packet and delivers it to the transport layer at the destinatio machine. A connection oriented transport layer makes a connection with the transport layer at th destination machine first before delivering the packets. After all the data are transferred, the connectio is terminated.
flow control.
at this layer is performed end to end rather than across a single link.
error control
at this layer is performed process-to process rather than across a single link. The sending transport laye makes sure that the entire message arrives at the receiving transport layer without error (damage, loss, c duplication). Error correction is usually achieved through retransmission.
5)Session Layer
The services provided by the first three layers (physical, data link, and network) are not sufficient fo some processes. The session layer is the network dialog controller. It establishes, maintains, an synchronizes the interaction among communicating systems. The session layer is responsible for dialo control and synchronization.
Specific responsibilities of the session layer include the following:

Dialog control. The session layer allows two systems to enter into a dialog. It allows the communicatio
between two processes to take place in either half duplex (one way at a time) or full-duplex (two ways a
a time) mode.
Synchronization. The session layer allows a process to add checkpoints, or synchronization points, to
stream of data. For example, if a system is sending a file of 2000 pages, it is advisable to inse
checkpoints after every 100 pages to ensure that each 100-page unit is received and acknowledge
independently. In this case, if a crash happens during the transmission of page 523, the only pages tha
need to be resent after system recovery are pages 501 to 523. Pages previous to 501 need not be resent.
6) Presentation Layer
The presentation layer is concerned with the syntax and semantics of the information exchanged betwee
two systems.
Translation. The processes (running programs) in two systems are usually exchanging information in th
form of character strings, numbers, and so on. The information must be changed to bit streams befor
being transmitted. Because different computers use different encoding systems, the presentation layer i
responsible for interoperability between these different encoding methods.
Encryption. To carry sensitive information, a system must be able to ensure privacy. Encryption mean
that the sender transforms the original information to another form and sends the resulting message ou
over the network. Decryption reverses the original process to transform the message back to its origina
form.
Compression. Data compression reduces the number of bits contained in the information. Dat
compression becomes particularly important in the transmission of multimedia such as text, audio, an
video.
7). Application Layer
The application layer enables the user, whether human or software, to access the network. It provide
user interfaces and support for services such as electronic mail, remote file access and transfer, share
database management, and other types of distributed information services.
In Figure shows the relationship of the application layer to the user and the presentation layer. Of th
many application services available, the figure shows only three: XAOO (message-handling services
X.500 (directory services), and file transfer, access, and management (FTAM). The user in this exampl
employs XAOO to send an e-mail message.
Specific services provided by the application layer include the following:
Network virtual terminal. A network virtual terminal is a software version of a physical terminal, and
allows a user to log on to a remote host. To do so, the application creates a software emulation of
terminal at the remote host. The user's computer talks to the software terminal which, in turn, talks to th
host, and vice versa. The remote host believes it is communicating with one of its own terminals an
allows the user to log on.
File transfer, access, and management. This application allows a user to access files in a remote host (t
make changes or read data), to retrieve files from a remote computer for use in the local computer, and t
manage or control files in a remote computer locally.
Mail services. This application provides the basis for e-mail forwarding and storage.
Directory services. This application provides distributed database sources and access for globa
information about various objects and services.

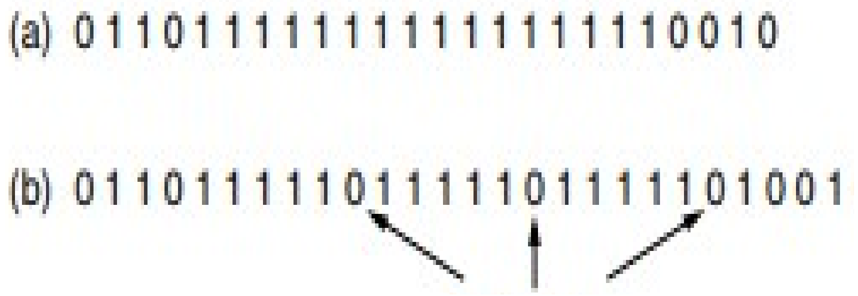
Q3(a) **What is framing? Explain the four farming methods**

The usual approach is for the data link layer to break up the bit stream into discrete frames, compute
short token called a checksum for each frame, and include the checksum in the frame when it i
transmitted. (Checksum algorithms will be discussed later in this chapter.) When a frame arrives at th
destination, the checksum is recomputed.
If the newly computed checksum is different from the one contained in the frame, the data link laye
knows that an error has occurred and takes steps to deal with it (e.g., discarding the bad frame an
possibly also sending back an error report). Breaking up the bit stream into frames is more difficult tha
it at first appears. A good design must make it easy for a receiver to find the start of new frames whil
using little of the channel bandwidth.
We will look at four methods:

1. Byte count.
2. Flag bytes with byte stuffing.
3. Flag bits with bit stuffing.
4. Physical layer coding violations.



The first framing method uses a field in the header to specify the number of bytes in the frame. When th data link layer at the destination sees the byte count, it knows how many bytes follow and hence wher the end of the frame is.
The second framing method gets around the problem of resynchronization after an error by having eac frame start and end with special bytes. Often the same byte, called a flag byte, is used as both the startin and ending delimiter.
The third method of delimiting the bit stream gets around a disadvantage of byte stuffing, which is that is tied to the use of 8-bit bytes. Framing can be also be done at the bit level, so frames can contain a arbitrary number of bits made up of units of any size. It was developed for the once very popular HDL (Highlevel Data Link Control) protocol.



When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically destuf (i.e., deletes) the 0 bit. Just as byte stuffing is completely transparent to the network layer in bot computers, so is bit stuffing. If the user data contain the flag pattern, 01111110, this flag is transmitted a 011111010 but stored in the receiver's memory as 01111110.

Q3(b) **Explain four different error-correcting codes.**
We will examine four different error-correcting codes:
1. Hamming codes.
2. Binary convolutional codes.
3. Reed-Solomon codes.
4. Low-Density Parity Check codes.
A frame consists of m data (i.e., message) bits and r redundant (i.e. check) bits. In a block code, the check bits are computed solely as a function of the m data bits with which they are associated, as thoug the m bits were looked up in a large table to find their corresponding r check bits. In a systematic code the m data bits are sent directly, along with the check bits, rather than being encoded themselves befor they are sent. In a linear code, the r check bits are computed as a linear function of the m data bits Exclusive OR (XOR) or modulo 2 addition is a popular choice. This means that encoding can be don with operations such as matrix multiplications or simple logic circuits. The codes we will look at in thi section are linear, systematic block codes unless otherwise noted.

To understand how errors can be handled, it is necessary to first look closely at what an error really is Given any two codewords that may be transmitted or received—say, 10001001 and 10110001—it i possible to determine how many corresponding bits differ. In this case, 3 bits differ. To determine how many bits differ, just XOR the two codewords and count the number of 1 bits in the result. For example:

10001001
10110001
00111000

The number of bit positions in which two codewords differ is called the Hamming distance (Hamming 1950). Its significance is that if two codewords are a Hamming distance d apart, it will require d single bit errors to convert one into the other.

The error-detecting and error-correcting properties of a block code depend on its Hamming distance. T reliably detect d errors, you need a distance d + 1 code because with such a code there is no way that single-bit errors can change a valid codeword into another valid codeword. When the receiver sees a illegal codeword, it can tell that a transmission error has occurred. Similarly, to correct d errors, yo need a distance 2d + 1 code because that way the legal codewords are so far apart that even with changes the original codeword is still closer than any other codeword. This means the original codewor can be uniquely determined based on the assumption that a larger number of errors are less likely. As simple example of an error-correcting code, consider a code with only four valid codewords:

0000000000, 0000011111, 1111100000, and 1111111111.

The third kind of error-correcting code we will describe is the Reed-Solomon code. Like Hammin codes, Reed-Solomon codes are linear block codes, and they are often systematic too. Unlike Hammin codes, which operate on individual bits, Reed-Solomon codes operate on m bit symbols. Naturally, th mathematics are more involved, so we will describe their operation by analogy. Reed-Solomon codes ar based on the fact that every n degree polynomial is uniquely determined by n + 1 points. For example, line having the form ax + b is determined by two points. Extra points on the same line are redundan which is helpful for error correction. Imagine that we have two data points that represent a line and w send those two data points plus two check points chosen to lie on the same line. If one of the points i received in error, we can still recover the data points by fitting a line to the received points. Three of th points will lie on the line, and one point, the one in error, will not. By finding the line we have correcte the error.

The final error-correcting code we will cover is the LDPC (Low-Density Parity Check) code. LDP codes are linear block codes that were invented by Robert Gallagher in his doctoral thesis (Gallaghe 1962). Like most theses, they were promptly forgotten, only to be reinvented in 1995 when advances i computing power had made them practical. In an LDPC code, each output bit is formed from only fraction of the input bits. This leads to a matrix representation of the code that has a low density of 1s hence the name for the code. The received codewords are decoded with an approximation algorithm tha iteratively improves on a best fit of the received data to a legal codeword. This corrects errors

Q4(a) **Explain the following:**

i) Simplex stop and wait protocol for error free channel.

The communication channel is still assumed to be error free, however, and the data traffic is sti simplex. One solution is to build the receiver to be powerful enough to process a continuous stream o back-to-back frames (or, equivalently, define the link layer to be slow enough that the receiver can kee up). It must have sufficient buffering and processing abilities to run at the line rate and must be able t pass the frames that are received to the network layer quickly enough. However, this is a worst-cas solution. It requires dedicated hardware and can be wasteful of resources if the utilization of the link i mostly low. Moreover, it just shifts the problem of dealing with a sender that is too fast elsewhere; in thi case to the network layer.

Although data traffic in this example is simplex, going only from the sender to the receiver, frames d travel in both directions. Consequently, the communication channel between the two data link layer needs to be capable of bidirectional information transfer. However, this protocol entails a stric alternation of flow: first the sender sends a frame, then the receiver sends a frame, then the sender send another frame, then the receiver sends another one, and so on. A halfduplex physical channel woul suffice here. As in protocol 1, the sender starts out by fetching a packet from the network layer, using i to construct a frame, and sending it on its way. But now, unlike in protocol 1, the sender must wait unt an acknowledgement frame arrives before looping back and fetching the next packet from the networ layer. The sending data link layer need not even inspect the incoming frame as there is only on possibility. The incoming frame is always an acknowledgement. The only difference between receiver and receiver2 is that after delivering a packet to the network layer, receiver2 sends an acknowledgemen frame back to the sender before entering the wait loop again. Because only the arrival of the frame bac at the sender is important, not its contents, the receiver need not put any particular information in it.

ii) Simplex stop and wait protocol for noisy channel.

Frames may be either damaged or lost completely. However, we assume that if a frame is damaged i transit, the receiver hardware will detect this when it computes the checksum. If the frame is damaged i such a way that the checksum is nevertheless correct—an unlikely occurrence—this protocol (and a other protocols) can fail (i.e., deliver an incorrect packet to the network layer). At first glance it migh seem that a variation of protocol 2 would work: adding a timer. The sender could send a frame, but th receiver would only send an acknowledgement frame if the data were correctly received. If a damage frame arrived at the receiver, it would be discarded. After a while the sender would time out and send th frame again. This process would be repeated until the frame finally arrived intact.

the network layer on B has no way of knowing that a packet has been lost or duplicated, so the data lin layer must guarantee that no combination of transmission errors, however unlikely, can cause a duplicat packet to be delivered to a network layer. Consider the following scenario:

1. The network layer on A gives packet 1 to its data link layer. The packet is correctly received at B an passed to the network layer on B. B sends an acknowledgement frame back to A.

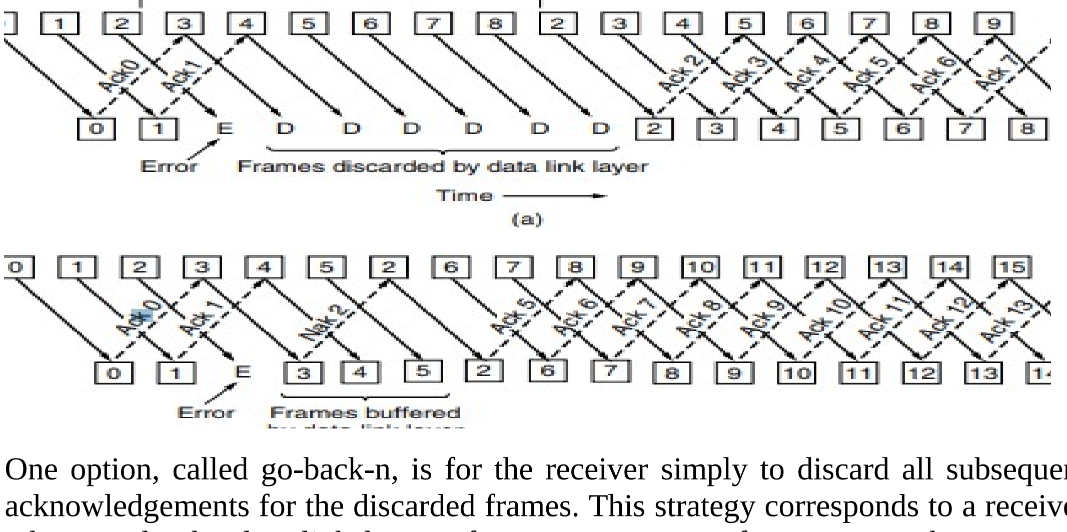
2. The acknowledgement frame gets lost completely. It just never arrives at all. Life would be a grea deal simpler if the channel mangled and lost only data frames and not control frames, but sad to say, th channel is not very discriminating.

3. The data link layer on A eventually times out. Not having received an acknowledgement, (incorrectly) assumes that its data frame was lost or damaged and sends the frame containing packet again.

4. The duplicate frame also arrives intact at the data link layer on B and is unwittingly passed to th network layer there. If A is sending a file to B, part of the file will be duplicated (i.e., the copy of the fil made by B will be incorrect and the error will not have been detected). In other words, the protocol wi fail.

Q4(b) **Explain the Go-back-N sliding window protocol.**

Two basic approaches are available for dealing with errors in the presence of pipelining, both of whic are shown in Fig.



One option, called go-back-n, is for the receiver simply to discard all subsequent frames, sending n acknowledgements for the discarded frames. This strategy corresponds to a receive window of size 1. I other words, the data link layer refuses to accept any frame except the next one it must give to th network layer. If the sender's window fills up before the timer runs out, the pipeline will begin to empty Eventually, the sender will time out and retransmit all unacknowledged frames in order, starting with th damaged or lost one. This approach can waste a lot of bandwidth if the error rate is high.

In Fig. we see go-back-n for the case in which the receiver's window is large. Frames 0 and 1 ar correctly received and acknowledged. Frame 2, however, is damaged or lost. The sender, unaware of thi problem, continues to send frames until the timer for frame 2 expires. Then it backs up to frame 2 an starts over with it, sending 2, 3, 4, etc. all over again.

In Fig. frames 0 and 1 are again correctly received and acknowledged and frame 2 is lost. When frame arrives at the receiver, the data link layer there notices that it has missed a frame, so it sends back a NAI for 2 but buffers 3. When frames 4 and 5 arrive, they, too, are buffered by the data link layer instead c being passed to the network layer. Eventually, the NAK 2 gets back to the sender, which immediatel resends frame 2. When that arrives, the data link layer now has 2, 3, 4, and 5 and can pass all of them t the network layer in the correct order. It can also acknowledge all frames up to and including 5, a shown in the figure. If the NAK should get lost, eventually the sender will time out for frame 2 and sen it (and only it) of its own accord, but that may be a quite a while later.

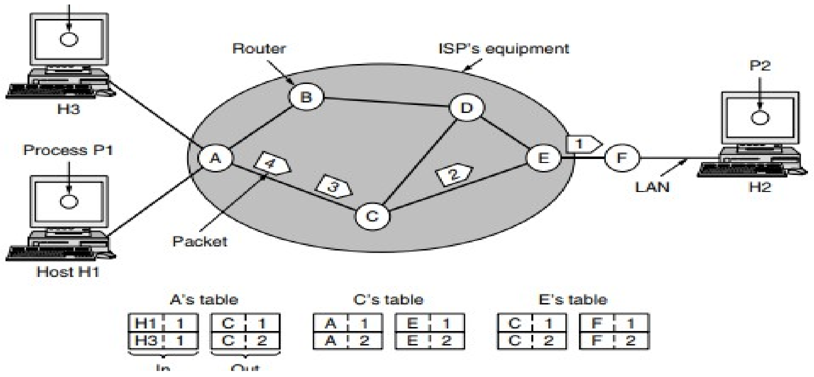
These two alternative approaches are trade-offs between efficient use of bandwidth and data link laye buffer space. Depending on which resource is scarcer, one or the other can be used. Fig. shows a go back-n protocol in which the receiving data link layer only accepts frames in order; frames following a error are discarded. In this protocol, for the first time we have dropped the assumption that the networ layer always has an infinite supply of packets to send. When the network layer has a packet it wants t send, it can cause a network layer ready event to happen. However, to enforce the flow control limit o the sender window or the number of unacknowledged frames that may be outstanding at any time, th data link layer must be able to keep the network layer from bothering it with more work. The librar procedures enable network layer and disable network layer do this job.

Q5(a) **How connection oriented services is implemented in network layer?**

For connection-oriented service, we need a virtual-circuit network. Let us see how that works. The ide behind virtual circuits is to avoid having to choose a new route for every packet sent.

when a connection is established, a route from the source machine to the destination machine is chose as part of the connection setup and stored in tables inside the routers. That route is used for all traffi flowing over the connection, exactly the same way that the telephone system works. When th connection is released, the virtual circuit is also terminated. With connection-oriented service, eac packet carries an identifier telling which virtual circuit it belongs to.

As an example, consider the situation shown in Fig. Here, host H1 has established connection 1 wit host H2. This connection is remembered as the first entry in each of the routing tables. The first line c A's table says that if a packet bearing connection identifier 1 comes in from H1, it is to be sent to route C and given connection identifier 1. Similarly, the first entry at C routes the packet to E, also wit connection identifier 1.



Now let us consider what happens if H3 also wants to establish a connection to H2. It choose connection identifier 1 (because it is initiating the connection and this is its only connection) and tells th network to establish the virtual circuit. This leads to the second row in the tables. Note that we have conflict here because although A can easily distinguish connection 1 packets from H1 from connection packets from H3, C cannot do this. For this reason, A assigns a different connection identifier to th outgoing traffic for the second connection. Avoiding conflicts of this kind is why routers need the abilit to replace connection identifiers in outgoing packets.

Q5(b) **Discuss about distance vector routing algorithm with suitable network.**

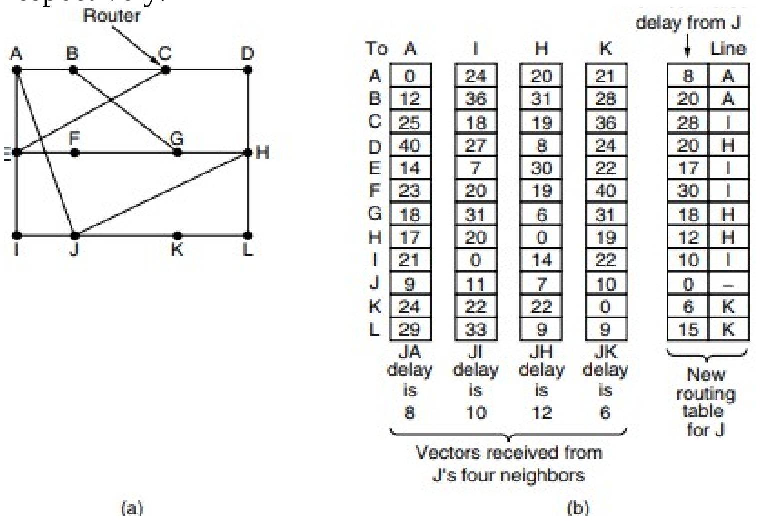
A distance vector routing algorithm operates by having each router maintain a table (i.e., a vector) givin the best known distance to each destination and which link to use to get there. These tables are update by exchanging information with the neighbors. Eventually, every router knows the best link to reac each destination.

The distance vector routing algorithm is sometimes called by other names, most commonly th distributed Bellman-Ford routing algorithm, after the researchers who developed it (Bellman, 1957; an Ford and Fulkerson, 1962). It was the original ARPANET routing algorithm and was also used in th Internet under the name RIP.

As an example, assume that delay is used as a metric and that the router knows the delay to each of it neighbors. Once every T msec, each router sends to each neighbor a list of its estimated delays to eac destination. It also receives a similar list from each neighbor.

Imagine that one of these tables has just come in from neighbor X, with Xi being X's estimate of hov long it takes to get to router i. If the router knows that the delay to X is m msec, it also knows that it ca reach router i via X in Xi + m msec. By performing this calculation for each neighbor, a router can fin out which estimate seems the best and use that estimate and the corresponding link in its new routin table. Note that the old routing table is not used in the calculation. This updating process is illustrated i Fig. 5-9. Part (a) shows a network.

The first four columns of part (b) show the delay vectors received from the neighbors of router J. claims to have a 12-msec delay to B, a 25-msec delay to C, a 40- msec delay to D, etc. Suppose that has measured or estimated its delay to its neighbors, A, I, H, and K, as 8, 10, 12, and 6 msec respectively.



To	A	I	H	K	delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K

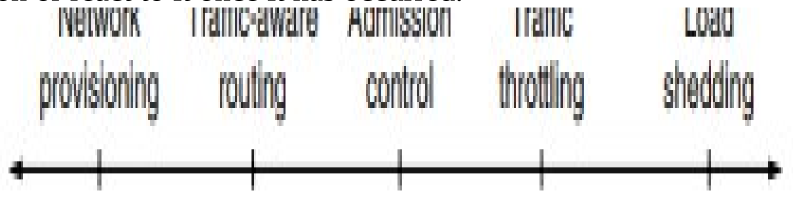
Consider how J computes its new route to router G. It knows that it can get to A in 8 msec, an furthermore A claims to be able to get to G in 18 msec, so J knows it can count on a delay of 26 msec t G if it forwards packets bound for G to A. Similarly, it computes the delay to G via I, H, and K as 41 (3 + 10), 18 (6 + 12), and 37 (31 + 6) msec, respectively. The best of these values is 18, so it makes a entry in its routing table that the delay to G is 18 msec and that the route to use is via H.

Q6(a) **What is congestion control? Discuss the approaches to congestion control.**

Too many packets present in (a part of) the network causes packet delay and loss that degrade performance. This situation is called congestion. The network and transport layers share th responsibility for handling congestion. Since congestion occurs within the network, it is the networ layer that directly experiences it and must ultimately determine what to do with the excess packets However, the most effective way to control congestion is to reduce the load that the transport layer i placing on the network.

Approaches to Congestion Control:-

The presence of congestion means that the load is (temporarily) greater than the resources (in a part c the network) can handle. Two solutions come to mind: increase the resources or decrease the load. A shown in Fig. 5-22, these solutions are usually applied on different time scales to either preven congestion or react to it once it has occurred.



The most basic way to avoid congestion is to build a network that is well matched to the traffic that carries. If there is a low-bandwidth link on the path along which most traffic is directed, congestion i likely. Sometimes resources can be added dynamically when there is serious congestion, for exampl turning on spare routers or enabling lines that are normally used only as backups (to make the syster fault tolerant) or purchasing bandwidth on the open market. More often, links and routers that ar regularly heavily utilized are upgraded at the earliest opportunity. This is called **provisioning** an happens on a time scale of months, driven by long-term traffic trends

routes can be tailored to traffic patterns that change during the day as network users wake and sleep i different time zones. For example, routes may be changed to shift traffic away from heavily used path by changing the shortest path weights. Some local radio stations have helicopters flying around thei cities to report on road congestion to make it possible for their mobile listeners to route their packet (cars) around hotspots. This is called **traffic-aware routing**. Splitting traffic across multiple paths is als helpful.

sometimes it is not possible to increase capacity. The only way then to beat back the congestion is t decrease the load. In a virtual-circuit network, new connections can be refused if they would cause th network to become congested. This is called **admission control**.

s, the network is forced to discard packets that it cannot deliver. The general name for this is **loa shedding**. A good policy for choosing which packets to discard can help to prevent congestion collapse.

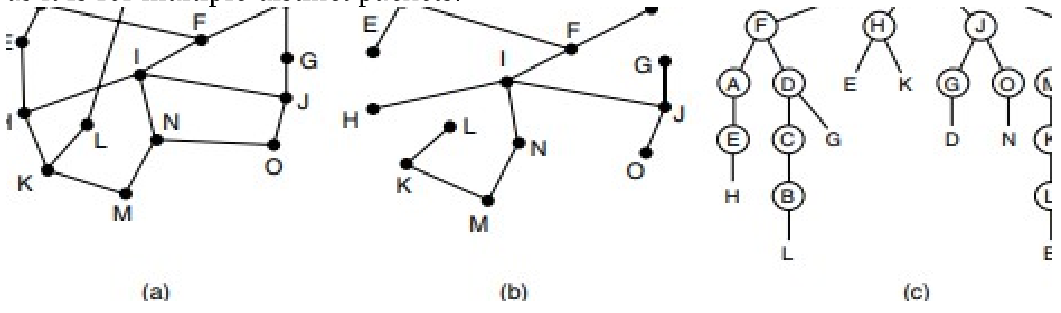
Q6(b) **Discuss about Broadcast routing in network layer**.

Sending a packet to all destinations simultaneously is called broadcasting. Various methods have bee proposed for doing it.

One broadcasting method that requires no special features from the network is for the source to simpl send a distinct packet to each destination. Not only is the method wasteful of bandwidth and slow, but also requires the source to have a complete list of all destinations. This method is not desirable i practice, even though it is widely applicable.

An improvement is multidestination routing, in which each packet contains either a list of destinations c a bit map indicating the desired destinations. When a packet arrives at a router, the router checks all th destinations to determine the set of output lines that will be needed. (An output line is needed if it is th best route to at least one of the destinations.) The router generates a new copy of the packet for eac output line to be used and includes in each packet only those destinations that are to use the line. I effect, the destination set is partitioned among the output lines. After a sufficient number of hops, eac packet will carry only one destination like a normal packet.

Multidestination routing is like using separately addressed packets, except that when several packet must follow the same route, one of them pays full fare and the rest ride free. The network bandwidth i therefore used more efficiently. However, this scheme still requires the source to know all th destinations, plus it is as much work for a router to determine where to send one multidestination packe as it is for multiple distinct packets.



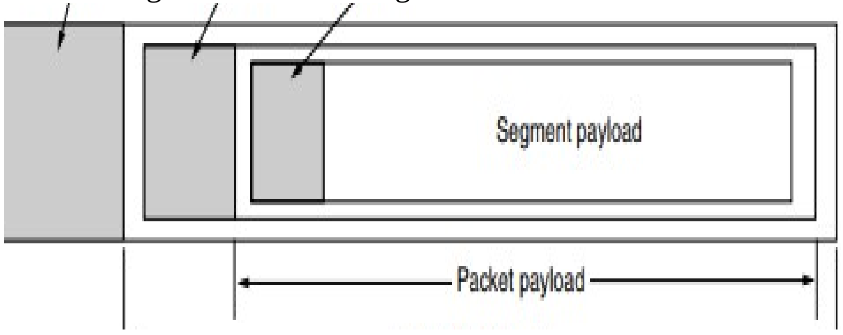
An example of reverse path forwarding is shown in Fig. . Part (a) shows a network, part (b) shows a sin

tree for router I of that network, and part (c) shows how the reverse path algorithm works. On the firs
hop, I sends packets to F, H, J, and N, as indicated by the second row of the tree. Each of these packet
arrives on the preferred path to I (assuming that the preferred path falls along the sink tree) and is s
indicated by a circle around the letter. On the second hop, eight packets are generated, two by each of th
routers that received a packet on the first hop. As it turns out, all eight of these arrive at previousl
unvisited routers, and five of these arrive along the preferred line. Of the six packets generated on th
third hop, only three arrive on the preferred path (at C, E, and K); the others are duplicates. After fiv
hops and 24 packets, the broadcasting terminates, compared with four hops and 14 packets had the sin
tree been followed exactly.

Q7(a) **Explain the following:**

i) Nesting of segments, packets and frames.

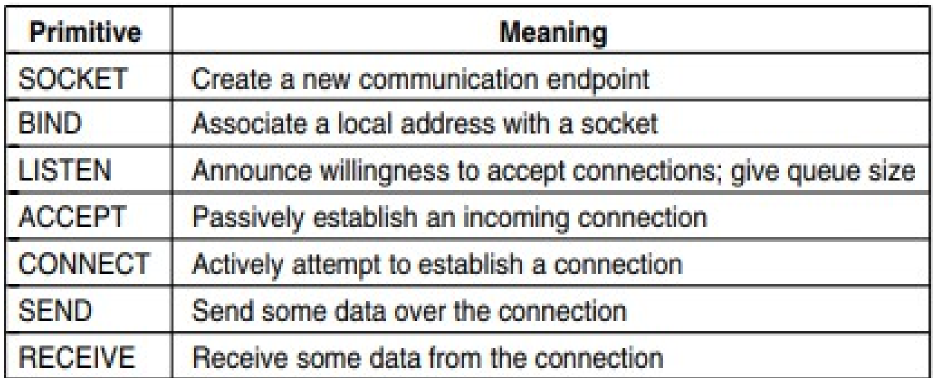
segments (exchanged by the transport layer) are contained in packets (exchanged by the network layer
In turn, these packets are contained in frames (exchanged by the data link layer). When a frame arrive
the data link layer processes the frame header and, if the destination address matches for local deliver
passes the contents of the frame payload field up to the network entity. The network entity similarl
processes the packet header and then passes the contents of the packet payload up to the transport entit
This nesting is illustrated in Fig.



Getting back to our client-server example, the client's CONNECT call causes a CONNECTIO
REQUEST segment to be sent to the server. When it arrives, the transport entity checks to see that th
server is blocked on a LISTEN (i.e., is interested in handling requests). If so, it then unblocks the serve
and sends a CONNECTION ACCEPTED segment back to the client. When this segment arrives, th
client is unblocked and the connection is established. Data can now be exchanged using the SEND an
RECEIVE primitives. In the simplest form, either party can do a (blocking) RECEIVE to wait for th
other party to do a SEND. When the segment arrives, the receiver is unblocked. It can then process th
segment and send a reply.

ii)Socket primitives in TCP

The primitives are listed in Fig. Roughly speaking, they follow the model of our first example but offe
more features and flexibility. We will not look at the corresponding segments here. That discussion wi
come later.



Primitive	Meaning
SOCKET	Create a new communication endpoint
BIND	Associate a local address with a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Passively establish an incoming connection
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection

The first four primitives in the list are executed in that order by servers. The SOCKET primitive create
a new endpoint and allocates table space for it within the transport entity. The parameters of the ca
specify the addressing format to be used, the type of service desired (e.g., reliable byte stream), and th
protocol.

	A successful SOCKET call returns an ordinary file descriptor for use in succeeding calls, the same wa an OPEN call on a file does. Newly created sockets do not have network addresses. These are assigne using the BIND primitive. Once a server has bound an address to a socket, remote clients can connect t it. The reason for not having the SOCKET call create an address directly is that some processes care abou their addresses (e.g., they have been using the same address for years and everyone knows this address whereas others do not. Next comes the LISTEN call, which allocates space to queue incoming calls fo the case that several clients try to connect at the same time. In contrast to LISTEN in our first exampl in the socket model LISTEN is not a blocking call.
Q7(b)	**Write socket programming with respect to internet file server.** The code has many limitations (discussed below), but in principle the server code can be compiled an run on any UNIX system connected to the Internet. The client code can be compiled and run on an other UNIX machine on the Internet, anywhere in the world. The client code can be executed wit appropriate parameters to fetch any file to which the server has access on its machine. The file is written to standard output, which, of course, can be redirected to a file or pipe. Let us look the server code first. It starts out by including some standard headers, the last three of which contain th main Internet-related definitions and data structures. Next comes a definition of SERVER PORT 12345. This number was chosen arbitrarily. Any number between 1024 and 65535 will work just as well, long as it is not in use by some other process; ports below 1023 are reserved for privileged users. Th next two lines in the server define two constants needed. The first one determines the chunk size in byte used for the file transfer. The second one determines how many pending connections can be held befor additional ones are discarded upon arrival. the server creates a socket and checks for errors (indicated by s < 0). In a production version of the cod the error message could be a trifle more explanatory. The call to setsockopt is needed to allow the port t be reused so the server can run indefinitely, fielding request after request. Now the IP address is boun to the socket and a check is made to see if the call to bind succeeded. The final step in the initialization i the call to listen to announce the server's willingness to accept incoming calls and tell the system to hol up to QUEUE SIZE of them in case new requests arrive while the server is still processing the curren one. If the queue is full and additional requests arrive, they are quietly discarded. At this point, the server enters its main loop, which it never leaves. The only way to stop it is to kill from outside. The call to accept blocks the server until some client tries to establish a connection with i If the accept call succeeds, it returns a socket descriptor that can be used for reading and writing analogous to how file descriptors can be used to read from and write to pipes. However, unlike pipe which are unidirectional, sockets are bidirectional, so sa (the accepted socket) can be used for readin from the connection and also for writing to it. A pipe file descriptor is for reading or writing but n both. After the connection is established, the server reads the file name from it. If the name is not yet availabl the server blocks waiting for it. After getting the file name, the server opens the file and enters a loo that alternately reads blocks from the file and writes them to the socket until the entire file has bee copied. Then the server closes the file and the connection and waits for the next connection to show u It repeats this loop forever.

Q8(a)

Write short notes on : I) TCP ii) UDP

i)TCP

A key feature of TCP, and one that dominates the protocol design, is that every byte on a TC connection has its own 32-bit sequence number. When the Internet began, the lines between routers wer mostly 56-kbps leased lines, so a host blasting away at full speed took over 1 week to cycle through th sequence numbers. At modern network speeds, the sequence numbers can be consumed at an alarmin rate, as we will see later. Separate 32-bit sequence numbers are carried on packets for the sliding windo position in one direction and for acknowledgements in the reverse direction, as discussed below. Th sending and receiving TCP entities exchange data in the form of segments. A TCP segment consists of fixed 20-byte header (plus an optional part) followed by zero or more data bytes. The TCP softwar decides how big segments should be. It can accumulate data from several writes into one segment or ca split data from one write over multiple segments. Two limits restrict the segment size.

First, each segment, including the TCP header, must fit in the 65,515- byte IP payload. Second, eac link has an MTU (Maximum Transfer Unit). Each segment must fit in the MTU at the sender an receiver so that it can be sent and received in a single, unfragmented packet. In practice, the MTU i generally 1500 bytes (the Ethernet payload size) and thus defines the upper bound on segment size.

When the segment arrives at the destination, the receiving TCP entity sends back a segment (with data any exist, and otherwise without) bearing an acknowledgement number equal to the next sequenc number it expects to receive and the remaining window size. If the sender's timer goes off before th acknowledgement is received, the sender transmits the segment again.

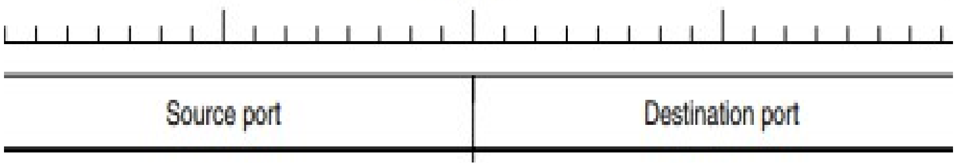
TCP must be prepared to deal with these problems and solve them in an efficient way. A considerabl amount of effort has gone into optimizing the performance of TCP streams, even in the face of networ problems. A number of the algorithms used by many TCP implementations will be discussed below.

ii)UDP:-

The Internet protocol suite supports a connectionless transport protocol called UDP (User Datagra Protocol). UDP provides a way for applications to send encapsulated IP datagrams without having t establish a connection. UDP is described in RFC 768.

UDP transmits segments consisting of an 8-byte header followed by the payload. The header is shown i Fig. 6-27. The two ports serve to identify the endpoints within the source and destination machine When a UDP packet arrives, its payload is handed to the process attached to the destination port.

UDP). Think of ports as mailboxes that applications can rent to receive packets. We will have more t say about them when we describe TCP, which also uses ports. In fact, the main value of UDP over jus using raw IP is the addition of the source and destination ports. Without the port fields, the transpo layer would not know what to do with each incoming packet. With them, it delivers the embedde segment to the correct application.



The source port is primarily needed when a reply must be sent back to the source. By copying the Sourc port field from the incoming segment into the Destination port field of the outgoing segment, the proces sending the reply can specify which process on the sending machine is to get it. The UDP length fiel includes the 8-byte header and the data. The minimum length is 8 bytes, to cover the header. Th maximum length is 65,515 bytes, which is lower than the largest number that will fit in 16 bits becaus of the size limit on IP packets.

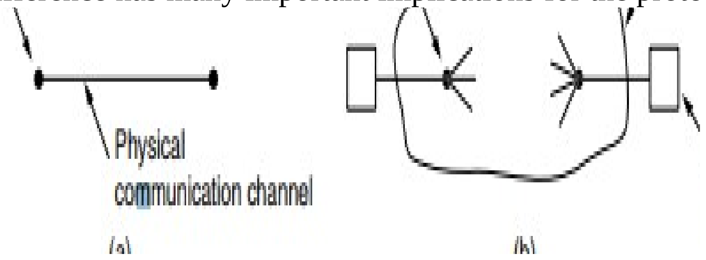
An optional Checksum is also provided for extra reliability. It checksums the header, the data, and conceptual IP pseudoheader. When performing this computation, the Checksum field is set to zero an the data field is padded out with an additional zero byte if its length is an odd number. The checksu algorithm is simply to add up all the 16-bit words in one's complement and to take the one' complement of the sum.

As a consequence, when the receiver performs the calculation on the entire segment, including th Checksum field, the result should be 0. If the checksum is not computed, it is stored as a 0, since by happy coincidence of one’s complement arithmetic a true computed 0 is stored as all 1s. Howeve turning it off is foolish unless the quality of the data does not matter (e.g., for digitized speech).

Q8(b) **Discuss about the elements of transport protocols.**

The transport service is implemented by a transport protocol used between the two transport entities.

At the data link layer, two routers communicate directly via a physical channel, whether wired c wireless, whereas at the transport layer, this physical channel is replaced by the entire network. Thi difference has many important implications for the protocols.



For one thing, over point-to-point links such as wires or optical fiber, it is usually not necessary for router to specify which router it wants to talk to—each outgoing line leads directly to a particular route In the transport layer, explicit addressing of destinations is required.

Another (exceedingly annoying) difference between the data link layer and the transport layer is th potential existence of storage capacity in the network. When a router sends a packet over a link, it ma arrive or be lost, but it cannot bounce around for a while, go into hiding in a far corner of the world, an suddenly emerge after other packets that were sent much later. If the network uses datagrams, which ar independently routed inside, there is a nonnegligible probability that a packet may take the scenic rout and arrive late and out of the expected order, or even that duplicates of the packet will arrive. Th consequences of the network’s ability to delay and duplicate packets can sometimes be disastrous an can require the use of special protocols to correctly transport information.

A final difference between the data link and transport layers is one of degree rather than of kinc Buffering and flow control are needed in both layers, but the presence in the transport layer of a larg and varying number of connections with bandwidth that fluctuates as the connections compete with eac other may require a different approach than we used in the data link layer.

Q9(a) **Explain the following i)Domain Name System ii) Electronic Mail.**

Domain Name System

Although programs theoretically could refer to Web pages, mailboxes, and other resources by using th network (e.g., IP) addresses of the computers on which they are stored, these addresses are hard fc people to remember. Also, browsing a company’s Web pages from 128.111.24.41 means that if th company moves the Web server to a different machine with a different IP address, everyone needs to b told the new IP address. Consequently, high-level, readable names were introduced in order to decoupl machine names from machine addresses.

this way, the company’s Web server might be known as www.cs.washington.edu regardless of its I address. Nevertheless, since the network itself understands only numerical addresses, some mechanisr is required to convert the names to network addresses. In the following sections, we will study how thi mapping is accomplished in the Internet.

However, well before many millions of PCs were connected to the Internet, everyone involved with realized that this approach could not continue to work forever. For one thing, the size of the file woul become too large. However, even more importantly, host name conflicts would occur constantly unles names were centrally managed, something unthinkable in a huge international network due to the loa and latency. To solve these problems, DNS (Domain Name System) was invented in 1983. It has been key part of the Internet ever since.

The essence of DNS is the invention of a hierarchical, domain-based naming scheme and a distribute database system for implementing this naming scheme. It is primarily used for mapping host names to I addresses but can also be used for other purposes. DNS is defined in RFCs 1034, 1035, 2181, and furthe elaborated in many others

Electronic Mail

	Email, like most other forms of communication, has developed its own conventions and styles. It is ver informal and has a low threshold of use. People who would never dream of calling up or even writing letter to a Very Important Person do not hesitate for a second to send a sloppily written email to him c her. By eliminating most cues associated with rank, age, and gender, email debates often focus o content, not status. With email, a brilliant idea from a summer student can have more impact than dumb one from an executive vice president. we will provide an overview of how email systems are organized and what they can do. The architectur of the email system is shown in Fig. 7-7. It consists of two kinds of subsystems: the user agents, whic allow people to read and send email, and the message transfer agents, which move the messages from th source to the destination. We will also refer to message transfer agents informally as mail servers. The message transfer agents are typically system processes. They run in the background on mail serve machines and are intended to be always available. Their job is to automatically move email through th system from the originator to the recipient with SMTP (Simple Mail Transfer Protocol). This is th message transfer step. Message transfer agents also implement mailing lists, in which an identical copy of a message i delivered to everyone on a list of email addresses. Other advanced features are carbon copies, blin carbon copies, high-priority email, secret (i.e., encrypted) email, alternative recipients if the primary on is not currently available, and the ability for assistants to read and answer their bosses' email.
Q9(b)	**Explain the DNS resource record types.** Every domain, whether it is a single host or a top-level domain, can have a set of resource record associated with it. These records are the DNS database. For a single host, the most common resourc record is just its IP address, but many other kinds of resource records also exist. When a resolver gives a domain name to DNS, what it gets back are the resource records associated wit that name. Thus, the primary function of DNS is to map domain names onto resource records. A resource record is a five-tuple. Although they are encoded in binary for efficiency, in most expositior resource records are presented as ASCII text, one line per resource record. The format we will use is a follows: Domain name Time to live Class Type Value The Domain name tells the domain to which this record applies. Normally, many records exist for eac domain and each copy of the database holds information about multiple domains. This field is thus th primary search key used to satisfy queries. The order of the records in the database is not significant. The Time to live field gives an indication of how stable the record is. Information that is highly stable i assigned a large value, such as 86400 (the number of seconds in 1 day). Information that is highl volatile is assigned a small value, such as 60 (1 minute). We will come back to this point later when w have discussed caching. The third field of every resource record is the Class. For Internet information, it is always IN. For nor Internet information, other codes can be used, but in practice these are rarely seen. The Type field tells what kind of record this is. There are many kinds of DNS records. The importar types are listed in Fig. 7-3. An SOA record provides the name of the primary source of informatio about the name server's zone (described below), the email address of its administrator, a unique seria number, and various flags and timeouts. The most important record type is the A (Address) record. It holds a 32-bit IPv4 address of an interfac for some host. The corresponding AAAA, or ''quad A,'' record holds a 128-bit IPv6 address. Ever Internet host must have at least one IP address so that other machines can communicate with it. Som hosts have two or more network interfaces, in which case they will have two or more type A or AAA resource records. Consequently, DNS can return multiple addresses for a single name.
Q10(a)	**Discuss about static and dynamic web pages.** **static web page:-** The basis of the Web is transferring Web pages from server to client. In the simplest form, Web page are static. That is, they are just files sitting on some server that present themselves in the same way eac time they are fetched and viewed. Just because they are static does not mean that the pages are inert a the browser, however. A page containing a video can be a static Web page. As mentioned earlier, th lingua franca of the Web, in which most pages are written, is HTML. The home pages of teachers ar

usually static HTML pages.
The home pages of companies are usually dynamic pages put together by a Web design company. In thi
section, we will take a brief look at static HTML pages as a foundation for later material. Reader
already familiar with HTML can skip ahead to the next section, where we describe dynamic content an
Web services.

dynamic web page:-

The static page model we have used so far treats pages as multimedia documents that are convenientl
linked together. It was a fitting model in the early days of the Web, as vast amounts of information wer
put online. Nowadays, much of the excitement around the Web is using it for applications and service
Examples include buying products on e-commerce sites, searching library catalogs, exploring map
reading and sending email, and collaborating on documents. These new uses are like tradition
application software (e.g., mail readers and word processors). The twist is that these applications ru
inside the browser, with user data stored on servers in Internet data centers. They use Web protocols t
access information via the Internet, and the browser to display a user interface. The advantage of thi
approach is that users do not need to install separate application programs, and user data can be accesse
from different computers and backed up by the service operator. It is proving so successful that it i
rivaling traditional application software. Of course, the fact that these applications are offered for free b
large providers helps. This model is the prevalent form of cloud computing, in which computing move
off individual desktop computers and into shared clusters of servers in the Internet.
There is more to dynamic content, however. The page that is returned may itself contain programs tha
run in the browser. In our map example, the program would let the user find routes and explore nearb
areas at different levels of detail. It would update the page, zooming in or out as directed by the use
(step 4). To handle some interactions, the program may need more data from the server. In this case, th
program will send a request to the server (step 5) that will retrieve more information from the databas
(step 6) and return a response (step 7). The program will then continue updating the page (step 4). Th
requests and responses happen in the background; the user may not even be aware of them because th
page URL and title typically do not change. By including client-side programs, the page can present
more responsive interface than with server-side programs alone.

Q10(b)

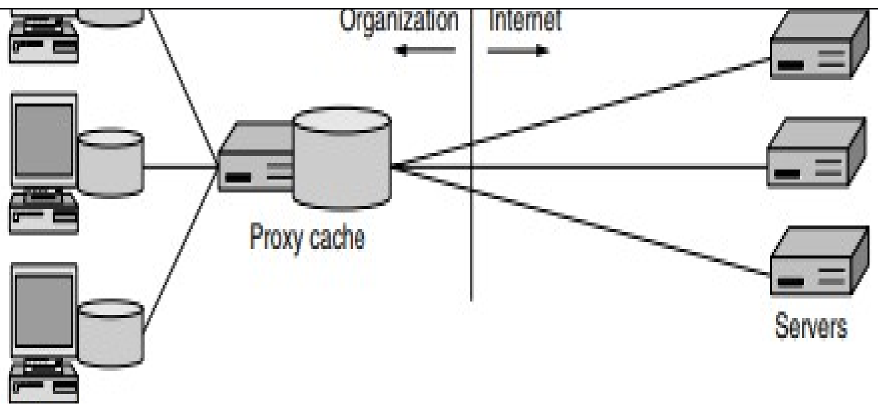
Explain in detail about server farm and web proxies.

Server Farms

It can only serve so many Web requests before the load is too great. The solution in this case is to us
more than one computer to make a Web server.
The difficulty with this seemingly simple model is that the set of computers that make up the server far
must look like a single logical Web site to clients. If they do not, we have just set up different Web site
that run in parallel. There are several possible solutions to make the set of servers appear to be one We
site. All of the solutions assume that any of the servers can handle a request from any client. To do thi
each server must have a copy of the Web site. The servers are shown as connected to a common back
end database by a dashed line for this purpose.

Web Proxies

Web requests and responses are sent using HTTP. A Web proxy is used to share a cache among users.
proxy is an agent that acts on behalf of someone else, such as the user. There are many kinds of proxie
For instance, an ARP proxy replies to ARP requests on behalf of a user who is elsewhere (and cann
reply for himself). A Web proxy fetches Web requests on behalf of its users. It normally provide
caching of the Web responses, and since it is shared across users it has a substantially larger cache than
browser.
When a proxy is used, the typical setup is for an organization to operate one Web proxy for all of i
users. The organization might be a company or an ISP. Both stand to benefit by speeding up We
requests for its users and reducing its bandwidth needs. While flat pricing, independent of usage, i
common for end users, most companies and ISPs are charged according to the bandwidth that they use.
This setup is shown in Fig. To use the proxy, each browser is configured to make page requests to th
proxy instead of to the page's real server. If the proxy has the page, it returns the page immediately.
not, it fetches the page from the server, adds it to the cache for future use, and returns it to the client th
requested it.

	 As well as sending Web requests to the proxy instead of the real server, clients perform their ow caching using its browser cache. The proxy is only consulted after the browser has tried to satisfy th request from its own cache. That is, the proxy provides a second level of caching.