As Per New VTU Syllabus w.e.f 2015-16
Choice Based Credit System(CBCS)

SUNSTAR

SUNSTAR EXAM SCANNER

COMPUTER NETWORKS

(V SEM.B.E. CSE / ISE)

SYLLABUS

Computer Networks			
[AS PER CHOICE BASED CREDIT SYSTEM (CBCS) SCHEME) (EFFECTIVE FROM THE ACADEMIC YEAR 2016 - 2017)			
Subject Code	15CS52	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

MODULE 1

Application Layer: Principles of Network Applications: Network Application Architectures, Processes Communicating, Transport Services Available to Applications, Transport Services Provided by the Internet, Application-Layer Protocols. The Web and HTTP: Overview of HTTP, Non-persistent and Persistent Connections, HTTP Message Format, User-Server Interaction: Cookies, Web Caching, The Conditional GET, File Transfer: FTP Commands & Replies, Electronic Mail in the Internet: SMTP, Comparison with HTTP, Mail Message Format, Mail Access Protocols, DNS; The Internet's Directory Service: Services Provided by DNS, Overview of How DNS Works, DNS Records and Messages, Peer-to-Peer Applications: P2P File Distribution, Distributed Hash Tables, Socket Programming: creating Network Applications: Socket Programming with UDP, Socket Programming with TCP. **10 Hours**

MODULE 2

Transport Layer : Introduction and Transport-Layer Services: Relationship Between Transport and Network Layers, Overview of the Transport Layer in the Internet, Multiplexing and Demultiplexing: Connectionless Transport: UDP,UDP Segment Structure, UDP Checksum, Principles of Reliable Data Transfer: Building a Reliable Data Transfer Protocol, Pipelined Reliable Data Transfer Protocols, Go-Back-N, Selective repeat, Connection-Oriented Transport TCP: The TCP Connection, TCP Segment Structure, Round-Trip Time Estimation and Timeout, Reliable Data Transfer, Flow Control, TCP Connection Management, Principles of Congestion Control: The Causes and the Costs of Congestion, Approaches to Congestion Control, Network-assisted congestion-control example, ATM ABR Congestion control, TCP Congestion Control: Fairness. **10 Hours**

MODULE 3

The Network layer: What's Inside a Router?: Input Processing, Switching, Output Processing, Where Does Queuing Occur? Routing control plane, IPv6,A Brief foray into IP Security, Routing Algorithms: The Link-State (LS) Routing Algorithm, The Distance-Vector (DV) Routing Algorithm, Hierarchical Routing, Routing in the Internet, Intra-AS Routing in the Internet: RIP, Intra-AS Routing in the Internet: OSPF, Inter/AS Routing: BGP, Broadcast Routing Algorithms and Multicast. **10 Hours**

MODULE 4

Wireless and Mobile Networks: Cellular Internet Access: An Overview of Cellular Network Architecture, 3G Cellular Data Networks: Extending the Internet to Cellular subscribers, On to 4G:LTE,Mobility management: Principles, Addressing, Routing to a mobile node, Mobile IP, Managing mobility in cellular Networks, Routing calls to a Mobile user, Handoffs in GSM, Wireless and Mobility: Impact on Higher-layer protocols. **10 Hours**

MODULE 5

Multimedia Networking: Properties of video, properties of Audio, Types of multimedia Network Applications, Streaming stored video: UDP Streaming, HTTP Streaming, Adaptive streaming and DASH, content distribution Networks, case studies: Netflix, You Tube and Kankan.

Network Support for Multimedia: Dimensioning Best-Effort Networks, Providing Multiple Classes of Service, Diffserv, Per-Connection Quality-of- Service (QoS) Guarantees: Resource Reservation and Call Admission **10 Hours**

Fifth Semester B.E. Degree Examination

CBCS - Model Question Paper - 1

COMPUTER NETWORKS

Time: 3 hrs. Max. Marks: 80

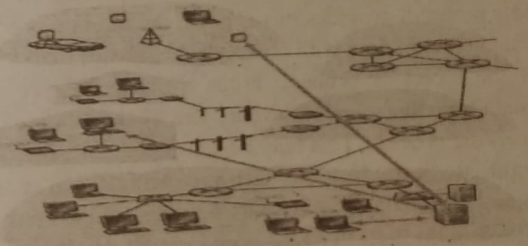
Note : ***Answer any FIVE full questions, selecting ONE full question from each module.***

MODULE - 1

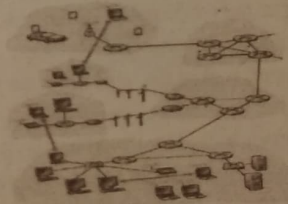
1. a. Explain architectural paradigms of network applications? (06 Marks)

Ans. The two predominant architectural paradigms used in modern network applications are: **the client-server architecture** or **the peer-to-peer (P2P) architecture**

In a client-server architecture, there is an always-on host, called the server, which services requests from many other hosts, called clients. Example is the Web application for which an always-on Web server services requests from browsers running on client hosts. When a Web server receives a request for an object from a client host, it responds by sending the requested object to the client host. In client-server architecture, clients do not directly communicate with each other; for example, in the Web application, two browsers do not directly communicate. Another characteristic of the client-server architecture is that the server has a fixed, well-known address, called an IP address. Because the server has a fixed, well-known address, and because the server is always on, a client can always contact the server by sending a packet to the server's IP address. The client-server architecture include the Web, FTP, Telnet, and e-mail. The client-server architecture is shown in Figure (1).



(1) Client - server Architecture



(2) Peer - to - peer Architecture

In a P2P architecture, there is minimal reliance on dedicated servers in data centers. The application exploits direct communication between pairs of intermittently connected hosts, called peers. The peers are not owned by the service provider, but are instead desktops and laptops controlled by users, with most of the peers residing in homes, universities, and offices. The peers communicate without passing through a dedicated server, the architecture is called peer-to-peer. Today's most popular and traffic-intensive applications are based on P2P architectures. These applications include file sharing (e.g., BitTorrent), peer-assisted download acceleration (e.g., Xunlei), Internet Telephony (e.g., Skype), and IPTV (e.g., Kankan and PPstream). The P2P architecture is illustrated in Figure (2). P2P architectures are also cost effective, they normally don't require significant server infrastructure and server bandwidth .

b. What are the services that a transport layer protocol can offer to applications invoking it? **(10 Marks)**

Ans. We can broadly classify the services along four dimensions: reliable data transfer, throughput, timing, and security.

(1) Reliable Data Transfer- Packets can get lost within a computer network. For many applications such as electronic mail, file transfer, remote host access, Web document transfers, and financial applications data loss can have devastating consequences. Thus, to support these applications, it has to be guaranteed that the data sent by one end of the application is delivered correctly and completely to the other end of the application. If a protocol provides such a guaranteed data delivery service, it is said to provide reliable data transfer. transport-layer protocol can potentially provide an application process-to-process reliable data transfer. When a transport protocol provides this service, the sending process can just pass its data into the socket and know with complete confidence that the data will arrive without errors at the receiving process. When a transport-layer protocol doesn't provide reliable data transfer, some of the data sent by the sending process may never arrive at the receiving process. This may be acceptable for loss-tolerant applications, most notably multimedia applications such as conversational audio/video that can tolerate some amount of data loss. In these multimedia applications, lost data might result in a small glitch in the audio/video—not a crucial impairment.

(2) Throughput- throughput, in the context of a communication session between two processes along a network path, is the rate at which the sending process can deliver bits to the receiving process. Because other sessions will be sharing the bandwidth along the network path, and because these other sessions will be coming and going, the available throughput can fluctuate with time. These observations lead to another natural service that a transport-layer protocol could provide, namely, guaranteed available throughput at some specified rate. With such a service, the application could request a guaranteed throughput of r bits/sec, and the transport protocol would then ensure that the available throughput is always at least r bits/sec. Such a guaranteed throughput service would appeal to many applications.

If the transport protocol cannot provide this throughput, the application would need to encode at a lower rate or may have to give up, since receiving, say, half of the needed throughput is of little or no use to this Internet telephony application. Applications that have throughput requirements are said to be bandwidth-sensitive applications. Many multimedia applications are bandwidth sensitive, some multimedia applications may use adaptive coding techniques to encode digitized voice or video at a rate that matches the currently available throughput. While bandwidth-sensitive applications have specific throughput requirements, elastic applications can make use of as much, or as little, throughput as happens to be available. Electronic mail, file transfer, and Web transfers are all elastic applications.

(3) Timing- A transport-layer protocol can also provide timing guarantees. timing guarantees can come in many shapes and forms. An example guarantee might be that every bit that the sender pumps into the socket arrives at the receiver's socket no more than 100 msec later. Such a service would be appealing to interactive real-time applications, such as Internet telephony, virtual environments, teleconferencing, and multiplayer games, all of which require tight timing constraints on data delivery in order to be effective. Long delays in Internet telephony, for example, tend to result in unnatural pauses in the conversation; in a multiplayer game or virtual interactive environment, a long delay between taking an action and seeing the response from the environment makes the application feel less realistic. For non-real-time applications, lower delay is always preferable to higher delay, but no tight constraint is placed on the end-to-end delays.

(4) Security -transport protocol can provide an application with one or more security services. For example, in the sending host, a transport protocol can encrypt all data transmitted by the sending process, and in the receiving host, the transport-layer protocol can decrypt the data before delivering the data to the receiving process. Such a service would provide confidentiality between the two processes, even if the data is somehow observed between sending and receiving processes. A transport protocol can also provide other security services in addition to confidentiality, including data integrity and end-point authentication

OR

2. a. Compare SMTP with Http? **(06 Marks)**

Ans. Both protocols are used to transfer files from one host to another: HTTP transfers files from a Web server to a Web client; SMTP transfers files from one mail server to another mail server. When transferring the files, both persistent HTTP and SMTP use persistent connections. Thus, the two protocols have common characteristics. However, there are important differences.

First, HTTP is mainly a pull protocol—someone loads information on a Web server and users use HTTP to pull the information from the server at their convenience. In particular, the TCP connection is initiated by the machine that wants to receive the file. On the other hand, SMTP is primarily a push protocol—the sending mail server pushes the file to the receiving mail server. In particular, the TCP connection

is initiated by the machine that wants to send the file.
Second difference, which we alluded to earlier, is that SMTP requires each message, including the body of each message, to be in 7-bit ASCII format. If the message contains characters that are not 7-bit ASCII or contains binary data, then the message has to be encoded into 7-bit ASCII. HTTP data does not impose this restriction.
Third important difference concerns how a document consisting of text and images is handled.HTTP encapsulates each object in its own HTTP response message. Internet mail places all of the message's objects into one message.

b. What is DNS? What are the 3 classes of DNS Server **(06 Marks)**

Ans. Domain name system (DNS): The DNS is defined as a distributed database implemented in a hierarchy of DNS servers, and an application-layer protocol that allows hosts to query the distributed database.
Three classes of DNS servers:
1) Root DNS servers- In the Internet there are 13 root DNS servers, most of which are located in North America. An October 2006 map of the root DNS servers is shown in Figure (1); a list of the current root DNS servers is available via Although we have referred to each of the 13 root DNS servers as if it were a single server, each "server" is actually a network of replicated servers, for both security and reliability purposes. All together, there are 247 root servers as of fall 2011.
2) Top-level domain (TLD) servers: These servers are responsible for top-level domains such as com, org, net, edu, and gov, and all of the country top-level domains such as uk, fr, ca, and jp. The company Verisign Global Registry Services maintains the TLD servers for the com top-level domain, and the company Educause maintains the TLD servers for the edu top level domain.
3) Authoritative DNS servers: Every organization with publicly accessible hosts on the Internet must provide publicly accessible DNS records that map the names of those hosts to IP addresses. An organization's authoritative DNS server houses these DNS records. An organization can choose to implement its own authoritative DNS server to hold these records; alternatively, the organization can pay to have these records stored in an authoritative DNS server of some service provider. Most universities and large companies implement and maintain their own primary and secondary authoritative DNS server.

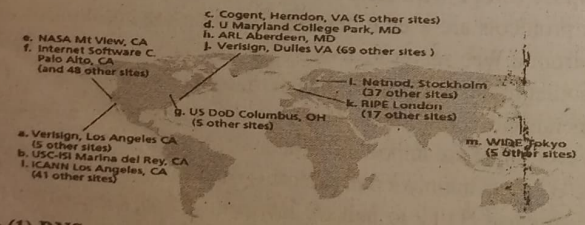


Fig (1) DNS root servers in 2012(name, organization, location)

c. What is Conditional GET? Explain? **(04 Marks)**

Ans. Caching can reduce user-perceived response times, it introduces a new problem—the copy of an object residing in the cache may be stale. The object housed in the Web server may have been modified since the copy was cached at the client. HTTP has a mechanism that allows a cache to verify that its objects are up to date. This mechanism is called the conditional GET.
An HTTP request message is a so-called conditional GET message if (1) the request message uses the GET method and (2) the request message includes an If-Modified-Since: header line.

MODULE-2

3. a. Explain Multiplexing and Demultiplexing of Transport layer? **(06 Marks)**

Transport-layer multiplexing and demultiplexing, extending the host-to-host delivery service provided by the network layer to a process-to-process delivery service for applications running on the hosts. At the destination host, the transport layer receives segments from the network layer just below. The transport layer has the responsibility of delivering the data in these segments to the appropriate application process running in the host. Example: Suppose you are sitting in front of your computer, and you are downloading Web pages while running one FTP session and two Telnet sessions. You therefore have four network application processes running—two Telnet processes, one FTP process, and one HTTP process. When the transport layer in your computer receives data from the network layer below, it needs to direct the received data to one of these four processes process can have one or more sockets, doors through which data passes from the network to the process and through which data passes from the process to the network. As shown in Figure (1), the transport layer in the receiving host does not actually deliver data directly to a process, instead to an intermediary socket. Because at any given time there can be more than one socket in the receiving host, each socket has a unique identifier.
The format of the identifier depends on whether the socket is a UDP or a TCP socket, consider how a receiving host directs an incoming transport-layer segment to the appropriate socket. Each transport-layer segment has a set of fields in the segment for this purpose. At the receiving end, the transport layer examines these fields to identify the receiving socket and then directs the segment to that socket. This job of delivering the data in a transport-layer segment to the correct socket is called demultiplexing.
The job of gathering data chunks at the source host from different sockets, encapsulating each data chunk with header information to create segments, and passing the segments to the network layer is called multiplexing.
The transport layer in the middle host in Figure (1) must demultiplex segments arriving from the network layer below to either process P1 or P2 above; this is done by directing the arriving segment's data to the corresponding process's socket. The transport layer in the middle host must also gather outgoing data from these sockets, form transport-layer segments, and pass these segments down to the network layer.

We have introduced multiplexing and demultiplexing in the context of the Internet transport protocols, it's important to realize that they are concerns whenever a single protocol at one layer is used by multiple protocols at the next higher layer.

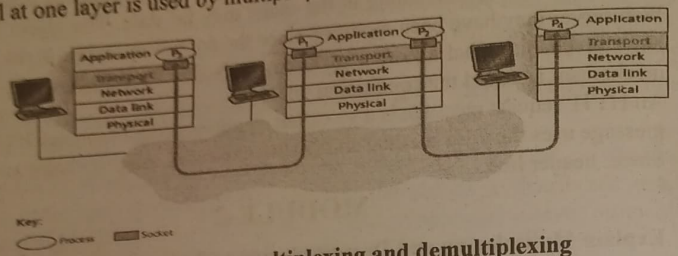


Fig (1) Transport-layer multiplexing and demultiplexing

To illustrate the demultiplexing job, recall the household analogy where Each of the kids is identified by his or her name. When Bill receives a batch of mail from the mail carrier, he performs a demultiplexing operation by observing to whom the letters are addressed and then hand delivering the mail to his brothers and sisters. Ann performs a multiplexing operation when she collects letters from her brothers and sisters and gives the collected mail to the mail person. transport-layer multiplexing requires (1) that sockets have unique identifiers, and (2) that each segment have special fields that indicate the socket to which the segment is to be delivered. These special fields, illustrated in Figure (2), are the source port number field and the destination port number field. Each port number is a 16-bit number, ranging from 0 to 65535. The port numbers ranging from 0 to 1023 are called well-known port numbers and are restricted, which means that they are reserved for use by well-known application protocols such as HTTP and FTP.

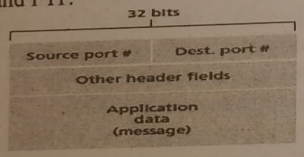


Fig (2) Source and destination port-number fields in a transport - layer

When we develop a new application , we must assign the application a port number. It should now be clear how the transport layer could implement the demultiplexing service: Each socket in the host could be assigned a port number, and when a segment arrives at the host, the transport layer examines the destination port number in the segment and directs the segment to the corresponding socket. The segment's data then passes through the socket into the attached process.

b. Explain UDP Segment Structure? (06 Marks)

Ans. The UDP segment structure, shown in Figure (1), is defined in RFC 768. The application data occupies the data field of the UDP segment. For example, for DNS, the data field contains either a query message or a response message. For a streaming audio application, audio samples fill the data field. The UDP header has only four

fields, each consisting of two bytes. The port numbers allow the destination host to pass the application data to the correct process running on the destination end system. The length field specifies the number of bytes in the UDP segment An explicit length value is needed since the size of the data field may differ from one UDP segment to the next. The checksum is used by the receiving host to check whether errors have been introduced into the segment. In truth, the checksum is also calculated over a few of the fields in the IP header in addition to the UDP segment. The length field specifies the length of the UDP segment, including the header, in bytes.

The UDP checksum provides for error detection. The checksum is used to determine whether bits within the UDP segment have been altered as it moved from source to destination.UDP at the sender side performs the 1s complement of the sum of all the 16-bit words in the segment, with any overflow encountered during the sum being wrapped around. This result is put in the checksum field of the UDP segment. Here we give a simple example of the checksum calculation. You can find details about efficient implementation of the calculation in RFC 1071 and performance over real data in. As an example, suppose that we have the following three 16-bit words:

0110011001100000
0101010101010101
1000111100001100

The sum of first two of these 16-bit words is

0110011001100000
0101010101010101
1011101110110101

Adding the third word to the above sum gives

1011101110110101
1000111100001100
0100101011000010

This last addition had overflow, which was wrapped around. The 1s complement is obtained by converting all the 0s to 1s and converting all the 1s to 0s. Thus the 1s complement of the sum 0100101011000010 is 1011010100111101, which becomes the checksum. At the receiver, all four 16-bit words are added, including the checksum. If no errors are introduced into the packet, then clearly the sum at the receiver will be 1111111111111111. If one of the bits is a 0, then we know that errors have been introduced into the packet.

UDP provides a checksum in the first place, as many link layer protocols also provide error checking. The reason is that there is no guarantee that all the links between source and destination provide error checking; that is, one of the links may use a link-layer protocol that does not provide error checking. Even if segments are correctly transferred across a link, it's possible that bit errors could be introduced when a segment is stored in a router's memory. Given that neither link-by-link reliability nor in-memory error detection is guaranteed, UDP must provide error detection at the transport layer, on an end-end basis, if the end-end data transfer service is to provide error detection. This is an example of the celebrated end-end principle in

system design which states that since certain functionality must be implemented on an end-end basis: "functions placed at the lower levels may be redundant or of little value when compared to the cost of providing them at the higher level." Because IP is supposed to run over just about any layer-2 protocol, it is useful for the transport layer to provide error checking as a safety measure.UDP provides error checking, it does not do anything to recover from an error. Some implementations of UDP simply discard the damaged segment; others pass the damaged segment to the application with a warning. TCP is also more complex than UDP

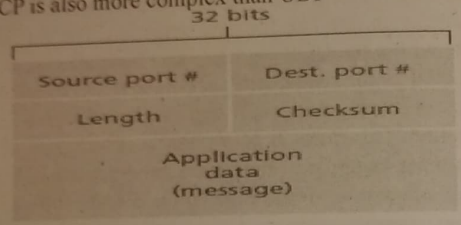


Figure (1) : UDP segment structure

c. Define ARQ Protocol? What are the three additional protocol capabilities required in ARQ protocol to handle the presence of bit errors? (04 Marks)

Ans. Message-dictation protocol uses both positive acknowledgments ("OK") and negative acknowledgments ("Please repeat that."). These control messages allow the receiver to let the sender know what has been received correctly, and what has been received in error and thus requires repeating. In a computer network setting, reliable data transfer protocols based on such retransmission are known as ARQ protocols. Three additional protocol capabilities are required in ARQ protocols to handle the presence of bit errors:

- **Error detection:** a mechanism is needed to allow the receiver to detect when bit errors have occurred. UDP uses the Internet checksum field for exactly this purpose. These techniques allow the receiver to detect and possibly correct packet bit errors.we need only know that these techniques require that extra bits be sent from the sender to the receiver; these bits will be gathered into the packet checksum field of the rdt2.0 data packet.
- **Receiver feedback:** the sender and receiver are typically executing on different end systems, possibly separated by thousands of miles, the only way for the sender to learn of the receiver's view of the world is for the receiver to provide explicit feedback to the sender. The positive (ACK) and negative (NAK) acknowledgment replies in the message-dictation scenario are examples of such feedback. Our rdt2.0 protocol will similarly send ACK and NAK packets back from the receiver to the sender. In principle, these packets need only be one bit long; for example, a 0 value could indicate a NAK and a value of 1 could indicate an ACK.
- **Retransmission:** A packet that is received in error at the receiver will be retransmitted by the sender.

OR

4. a. Explain AIMD form of Congestion Control? (06 Marks)

Ans. TCP's congestion control consists of linear (additive) increase in cwnd of 1 MSS per RTT and then a halving (multiplicative decrease) of cwnd on a triple duplicate-ACK event. TCP congestion control is often referred to as an additive-increase, multiplicative decrease (AIMD) form of congestion control. AIMD congestion control gives rise to the "saw tooth" behavior shown in Figure (1), it illustrates our earlier intuition of TCP "probing" for bandwidth TCP linearly increases its congestion window size until a triple duplicate-ACK event occurs. It then decreases its congestion window size by a factor of two but then again begins increasing it linearly, probing to see if there is additional available bandwidth.

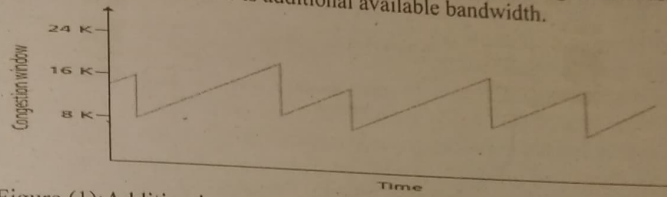


Figure (1):Additive-increase, multiplicative-decrease congestion control

TCP implementations use the Reno algorithm. Many variations of the Reno algorithm have been proposed. The TCP Vegas algorithm attempts to avoid congestion while maintaining good throughput.

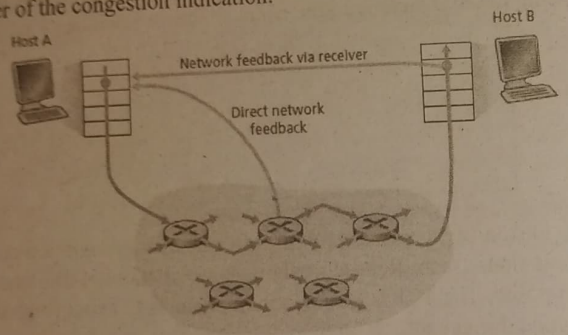
(1) The basic idea of Vegas is to detect congestion in the routers between source and destination before packet loss occurs

(2) lower the rate linearly when this imminent packet loss is detected. Imminent packet loss is predicted by observing the RTT. The longer the RTT of the packets, the greater the congestion in the routers. Linux supports a number of congestion-control algorithms and allows a system administrator to configure which version of TCP will be used. The default version of TCP in Linux version 2.6.18 was set to CUBIC, a version of TCP developed for high-bandwidth applications. TCP's AIMD algorithm was developed based on a tremendous amount of engineering insight and experimentation with congestion control in. operational networks.TCP's congestion-control algorithm serves as a distributed asynchronous-optimization algorithm that results in several important aspects of user and network performance being simultaneously optimized

b. Explain Network-Assisted Congestion Control in detail ? (04 Marks)

Ans. With network-assisted congestion control, network-layer components provide explicit feedback to the sender regarding the congestion state in the network. This feedback may be as simple as a single bit indicating congestion at a link. This approach was taken in the early IBM SNA and DEC DECnet architectures, was recently proposed for TCP/IP networks and is used in ATM available bit-rate (ABR) congestion control. Sophisticated network feedback is also possible. For example,

one form of ATM ABR congestion control that allows a router to inform the sender explicitly of the transmission rate it can support on an outgoing link. The XCP protocol provides router-computed feedback to each source, carried in the packet header, regarding how that source should increase or decrease its transmission rate. For network-assisted congestion control, congestion information is typically fed back from the network to the sender in one of two ways, as shown in Figure (2). Direct feedback may be sent from a network router to the sender. This form of notification typically takes the form of a choke packet. The second form of notification occurs when a router marks/updates a field in a packet flowing from sender to receiver to indicate congestion. Upon receipt of a marked packet, the receiver then notifies the sender of the congestion indication.



Congestion-control algorithm in ATM ABR—a protocol that takes a network-assisted approach toward congestion control. Our goal here is not to describe aspects of the ATM architecture in, but rather to illustrate a protocol that takes a markedly different approach toward congestion control from that of the Internet's TCP protocol. We only present below those few aspects of the ATM architecture that are needed to understand ABR congestion control. Fundamentally ATM takes a virtual-circuit (VC) oriented approach toward packet switching. Each switch on the source-to-destination path will maintain state about the source to destination VC. This per-VC state allows a switch to track the behavior of individual senders and to take source-specific congestion-control actions This per-VC state at network switches makes ATM ideally suited to perform network-assisted congestion control. ABR has been designed as an elastic data transfer service in a manner reminiscent of TCP. When the network is underloaded, ABR service should be able to take advantage of the spare available bandwidth; when the network is congested, ABR service should throttle its transmission rate to some predetermined minimum transmission rate.

c. Explain Sequence and Acknowledgement number in Telnet? (06 Marks)

Ans. Telnet, defined in RFC 854, is a popular application-layer protocol used for remote login. It runs over TCP and is designed to work between any pair of hosts. The bulk data transfer applications.Telnet is an interactive application. Telnet example TCP sequence and acknowledgment numbers. Many users now prefer to use the SSH

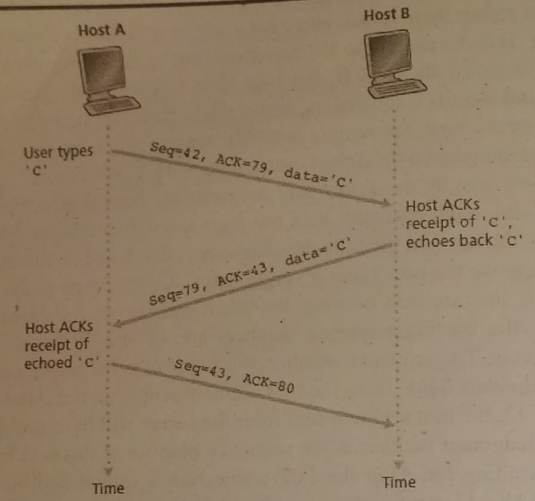
protocol rather than Telnet, since data sent in a Telnet connection is not encrypted, making Telnet vulnerable to eavesdropping attacks .Suppose Host A initiates a Telnet session with Host B. Because Host A initiates the session, it is labeled the client, and Host B is labeled the server. Each character typed by the user will be sent to the remote host; the remote host will send back a copy of each character, which will be displayed on the Telnet user's screen. This "echo back" is used to ensure that characters seen by the Telnet user have already been received and processed at the remote site. Each character thus traverses the network twice between the time the user hits the key and the time the character is displayed on the user's monitor.

Now suppose the user types a single letter, 'C,' and then grabs a coffee. The TCP segments that are sent between the client and server. As shown in Figure (1), we suppose the starting sequence numbers are 42 and 79 for the client and server, respectively. The sequence number of a segment is the sequence number of the first byte in the data field. Thus, the first segment sent from the client will have sequence number 42; the first segment sent from the server will have sequence number 79.the acknowledgment number is the sequence number of the next byte of data that the host is waiting for. After the TCP connection is established but before any data is sent, the client is waiting for byte 79 and the server is waiting for byte 42. As shown in Figure (1), three segments are sent. The first segment is sent from the client to the server, containing the 1-byte ASCII representation of the letter 'C' in its data field. This first segment also has 42 in its sequence number field. Also, because the client has not yet received any data from the server, this first segment will have 79 in its acknowledgment number field.

The second segment is sent from the server to the client. It serves a dual purpose. First it provides an acknowledgment of the data the server has received. By putting 43 in the acknowledgment field, the server is telling the client that it has successfully received everything up through byte 42 and is now waiting for bytes 43 onward. The second purpose of this segment is to echo back the letter 'C.' Thus, the second segment has the ASCII representation of 'C' in its data field. This second segment has the sequence number 79, the initial sequence number of the server-to client data flow of this TCP connection, as this is the very first byte of data that the server is sending.

Note that the acknowledgment for client-to server data is carried in a segment carrying server-to-client data; this acknowledgment is said to be piggybacked on the server-to-client data segment.

The third segment is sent from the client to the server. Its sole purpose is to acknowledge the data it has received from the server. This segment has an empty data field. The segment has 80 in the acknowledgment number field because the client has received the stream of bytes up through byte sequence number 79 and it is now waiting for bytes 80 onward. You might think it odd that this segment also has a sequence number since the segment contains no data. But because TCP has a sequence number field, the segment needs to have some sequence number.



MODULE - 3

5. a. Write a note on ICMP? (6 Marks)

Ans. ICMP, specified in [RFC 792], is used by hosts and routers to communicate network-layer information to each other use of ICMP is for error reporting. For example, when running a Telnet, FTP, or HTTP session, you may have encountered an error message such as "Destination network unreachable." This message had its origins in ICMP. At some point, an IP router was unable to find a path to the host specified in your Telnet, FTP, or HTTP application. That router created and sent a type-3 ICMP message to your host indicating the error. ICMP is often considered part of IP but architecturally it lies just above IP, as ICMP messages are carried inside IP datagrams. ICMP messages are carried as IP payload, just as TCP or UDP segments are carried as IP payload. Similarly, when a host receives an IP datagram with ICMP specified as the upper-layer protocol, it demultiplexes the datagram's contents to ICMP, just as it would demultiplex a datagram's content to TCP or UDP.

ICMP messages have a type and a code field, and contain the header and the first 8 bytes of the IP datagram that caused the ICMP message to be generated in the first place. Selected ICMP message types are shown in Figure(1). Note that ICMP messages are used not only for signaling error conditions. The well-known ping program sends an ICMP type 8 code 0 message to the specified host. The destination host, seeing the echo request, sends back a type 0 code 0 ICMP echo reply. TCP/IP implementations support the ping server directly in the operating system; that is, the server is not a process. Provides the source code for the ping client program. Note that the client program needs to be able to instruct the operating system to generate an ICMP message of type 8 code 0. Another interesting ICMP message is the source quench message. This message is seldom used in practice. Its original purpose was

to perform congestion control to allow a congested router to send an ICMP source quench message to a host to force that host to reduce its transmission rate. TCP has its own congestion-control mechanism that operates at the transport layer, without the use of network-layer feedback such as the ICMP source quench message. Trace route program, which allows us to trace a route from a host to any other host in the world. Interestingly, Trace route is implemented with ICMP messages. To determine the names and addresses of the routers between source and destination, Trace route in the source sends a series of ordinary IP datagrams to the destination. Each of these datagrams carries a UDP segment with an unlikely UDP port number.

The first of these datagrams has a TTL of 1, the second of 2, the third of 3, and so on. The source also starts timers for each of the datagrams. When the nth datagram arrives at the nth router, the nth router observes that the TTL of the datagram has just expired. According to the rules of the IP protocol, the router discards the datagram and sends an ICMP warning message to the source (type 11 code 0). This warning message includes the name of the router and its IP address. When this ICMP message arrives back at the source, the source obtains the round-trip time from the timer and the name and IP address of the nth router from the ICMP message.

How does a Traceroute source know when to stop sending UDP segments? Source increments the TTL field for each datagram it sends. Thus, one of the datagrams will eventually make it all the way to the destination host. Because this datagram contains a UDP segment with an unlikely port number, the destination host sends a port unreachable ICMP message (type 3 code 3) back to the source. When the source host receives this particular ICMP message, it knows it does not need to send additional probe packets.the source host learns the number and the identities of routers that lie between it and the destination host and the round-trip time between the two hosts. the Traceroute client program must be able to instruct the operating system to generate UDP datagrams with specific TTL values and must also be able to be notified by its operating system when ICMP messages arrive. Now that you understand how Traceroute works, you may want to go back and play with it some more

ICMP Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

(10 Marks)

b. Explain Dijkstra's algorithm ?

Ans. The link-state routing algorithm is known as Dijkstra's algorithm. Dijkstra's algorithm computes the least-cost path from one node to all other nodes in the network. Dijkstra's algorithm is iterative and has the property that after the kth iteration of the algorithm, the least-cost paths are known to k destination nodes, and among the least-cost paths to all destination nodes, these k paths will have the k smallest costs.

Let us define the following notation:

- D(v): cost of the least-cost path from the source node to destination v as of this iteration of the algorithm.
- p(v): previous node (neighbor of v) along the current least-cost path from the source to v.
- N_ : subset of nodes; v is in N_ if the least-cost path from the source to v is definitively known.

The global routing algorithm consists of an initialization step followed by a loop. The number of times the loop is executed is equal to the number of nodes in the network. Upon termination, the algorithm will have calculated the shortest paths from the source node u to every other node in the network.

Link-State (LS) Algorithm for Source Node u

```
1 Initialization:
2 N' = {u}
3 for all nodes v
4 if v is a neighbor of u
5 then D(v) = c(u,v)
6 else D(v) = ∞
7
8 Loop
9 find w not in N' such that D(w) is a minimum
10 add w to N'
11 update D(v) for each neighbor v of w and not in N':
12 D(v) = min( D(v), D(w) + c(w,v) )
13 /* new cost to v is either old cost to v or known
14 least path cost to w plus cost from w to v */
15 until N'= N
```

As an example, consider the network in Figure (1) and compute the least-cost paths from u to all possible destinations. A tabular summary of the algorithm's computation is shown in Table (1), where each line in the table gives the values of the algorithm's variables at the end of the iteration.

Let's consider the few first steps in detail.

- In the initialization step, the currently known least-cost paths from u to its directly attached neighbors, v, x, and w, are initialized to 2, 1, and 5, respectively. Note in particular that the cost to w is set to 5 (even though we will soon see that a lesser-cost path does indeed exist) since this is the cost of the direct (one hop) link

from u to w. The costs to y and z are set to infinity because they are not directly connected to u.

- In the first iteration, we look among those nodes not yet added to the set N_ and find that node with the least cost as of the end of the previous iteration. That node is x, with a cost of 1, and thus x is added to the set N_. Line 12 of the LS algorithm is then performed to update D(v) for all nodes v, yielding the results shown in the second line (Step 1) in Table (1). The cost of the path to v is unchanged. The cost of the path to w (which was 5 at the end of the initialization) through node x is found to have a cost of 4. Hence this lower-cost path is selected and w's predecessor along the shortest path from u is set to x. Similarly, the cost to y (through x) is computed to be 2, and the table is updated accordingly.
- In the second iteration, nodes v and y are found to have the least-cost paths (2), and we break the tie arbitrarily and add y to the set N_ so that N_ now contains u, x, and y. The cost to the remaining nodes not yet in N_, that is, nodes v, w, and z, are updated via line 12 of the LS algorithm, yielding the results shown in the third row in the Table (1).
- And so on. . . .

When the LS algorithm terminates, we have, for each node, its predecessor along the least cost path from the source node. For each predecessor, we also have its predecessor, and so in this manner we can construct the entire path from the source to all destinations. The forwarding table in a node, say node u, can then be constructed from this information by storing, for each destination, the next-hop node on the least-cost path from u to the destination. Figure (1) shows the resulting least-cost paths and forwarding table in u for the network in Figure (1).

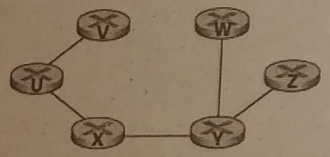
step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

What is the computational complexity of this algorithm?

Given n nodes (not counting the source), how much computation must be done in the worst case to find the least-cost paths from the source to all destinations? In the first iteration, we need to search through all n nodes to determine the node, w, not in N_ that has the minimum cost. In the second iteration, we need to check n – 1 nodes to determine the minimum cost; in the third iteration n – 2 nodes, and so on. Overall, the total number of nodes we need to search through over all the iterations is n(n + 1)/2, and thus we say that the preceding implementation of the LS algorithm has worst-case complexity of order n squared: O(n2).

A simple network topology where link costs are equal to the load carried on the link, for example, reflecting the delay that would be experienced. In this example, link costs are not symmetric; that is, c(u,v) equals c(v,u) only if the load carried on both

directions on the link (u,v) is the same. In this example, node z originates a unit of traffic destined for w, node x also originates a unit of traffic destined for w, and node y injects an amount of traffic equal to e, also destined for w. The initial routing is shown with the link costs corresponding to the amount of traffic carried. When the LS algorithm is next run, node y determines that the clockwise path to w has a cost of 1, while the counter clockwise path to w (which it had been using) has a cost of 1 + e. least-cost path to w is now clockwise. Similarly, x determines that its new least-cost path to w is also clockwise, resulting in costs. When the LS algorithm is run next, nodes x, y, and z all detect a zero-cost path to w in the counterclockwise direction, and all route their traffic to the counterclockwise routes. The next time the LS algorithm is run, x, y, and z all then route their traffic to the clockwise routes. What can be done to prevent such oscillations (which can occur in any algorithm, not just an LS algorithm, that uses a congestion or delay-based link metric)? One solution would be to mandate that link costs not depend on the amount of traffic carried an unacceptable solution since one goal of routing is to avoid highly congested links. Another solution is to ensure that not all routers run the LS algorithm at the same time. This seems a more reasonable solution, since if routers ran the LS algorithm with the same periodicity, the execution instance of the algorithm would not be the same at each node. Researchers have found that routers in the Internet can self synchronize among themselves [Floyd Synchronization 1994]. They initially execute the algorithm with the same period but at different instants of time, the algorithm execution instance can eventually become, and remain, synchronized at the routers. One way to avoid such self synchronization is for each router to randomize the time it sends out a link advertisement. the distance-vector routing algorithm is used today.



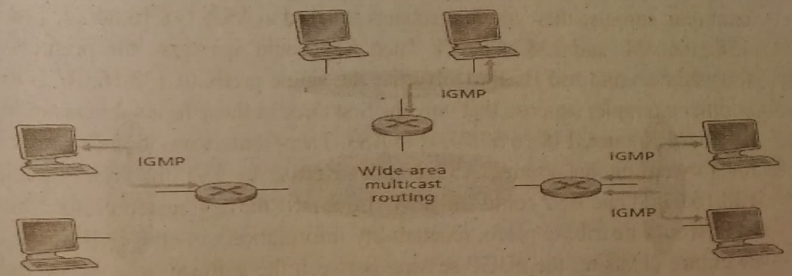
Destination	Link
v	(u, v)
w	(u, x)
x	(u, x)
y	(u, x)
z	(u, x)

OR

6. a. Write a short note on IGMP? (06 Marks)

Ans. The IGMP protocol version 3 operates between a host and its directly attached router as shown in Figure (1). Figure shows three first-hop multicast routers, each connected to its attached hosts via one outgoing local interface. This local interface is attached to a LAN in this example, and while each LAN has multiple attached hosts, at most a few of these hosts will typically belong to a given multicast group at any given time. IGMP provides the means for a host to inform its attached router that an application running on the host wants to join a specific multicast group. Given that the scope of IGMP interaction is limited to a host and its attached router, another protocol is

clearly required to coordinate the multicast routers throughout the Internet, so that multicast datagrams are routed to their final destinations. This latter functionality is accomplished by network-layer multicast routing algorithms, such as those we will consider shortly. Network-layer multicast in the Internet thus consists of two complementary components: IGMP and multicast routing protocols. IGMP has only three message types. Like ICMP, IGMP messages are carried (encapsulated) within an IP datagram, with an IP protocol number of 2. The membership_ query message is sent by a router to all hosts on an attached interface to determine the set of all multicast groups that have been joined by the hosts on that interface. Hosts respond to a membership_ query message with an IGMP membership_report message. membership_report messages can also be generated by a host when an application first joins a multicast group without waiting for a membership_query message from the router. The final type of IGMP message is the leave_group message.this message is optional. But if it is optional, how does a router detect when a host leaves the multicast group? The router infers that a host is no longer in the multicast group if it no longer responds to a membership_query message with the given group address. This is an example of what is sometimes called soft state in an Internet protocol. In a soft state protocol, the state is removed via a timeout event The term soft state was coined by Clark, who described the notion of periodic state refresh messages being sent by an end system, and suggested that with such refresh messages, state could be lost in a crash and then automatically restored by subsequent refresh messages all transparently to the end system and without invoking any explicit crash-recovery procedures: ". . . the state information would not be critical in maintaining the desired type of service associated with the flow. Type of service would be enforced by the end points, which would periodically send messages to ensure that the proper type of service was being associated with the flow. The state information associated with the flow could be lost in a crash without permanent disruption of the service features being used. This concept "soft state," and it may very well permit us to achieve our primary goals of survivability and flexibility. . ." It has been argued that soft-state protocols result in simpler control than hard state protocols, which not only require state to be explicitly added and removed, but also require mechanisms to recover from the situation where the entity responsible for removing state has terminated prematurely or failed.



b. Explain Border Gateway Protocol(BGP)? **(06 Marks)**

Ans. The Border Gateway Protocol version 4, specified in RFC 4271 is the de facto standard inter-AS routing protocol in today's Internet. It is commonly referred to as BGP4 or simply as BGP. As an inter-AS routing protocol BGP provides each AS a means to

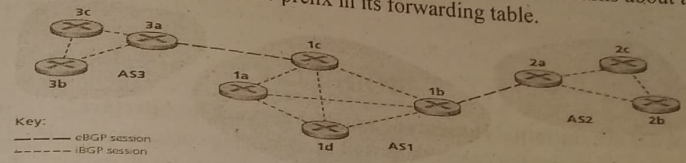
1. Obtain subnet reachability information from neighboring ASs.
2. Propagate the reachability information to all routers internal to the AS.
3. Determine "good" routes to subnets based on the reachability information and on AS policy.

BGP allows each subnet to advertise its existence to the rest of the Internet. A subnet screams "I exist and I am here," and BGP makes sure that all the ASs in the Internet know about the subnet and how to get there. If it weren't for BGP, each subnet would be isolated alone and unknown by the rest of the Internet.

BGP is an absolutely critical protocol for the Internet in essence, it is the protocol that glues the whole thing together we need to acquire at least a rudimentary understanding of how it works. In BGP, pairs of routers exchange routing information over semipermanent TCP connections using port 179. The semi-permanent TCP connections for the network are shown in Figure. There is typically one such BGP TCP connection for each link that directly connects two routers in two different ASs; thus, in Figure, there is a TCP connection between gateway routers 3a and 1c and another TCP connection between gateway routers 1b and 2a. There are also semipermanent BGP TCP connections between routers within an AS. Figure displays a common configuration of one TCP connection for each pair of routers internal to an AS, creating a mesh of TCP connections within each AS. For each TCP connection, the two routers at the end of the connection are called BGP peers, and the TCP connection along with all the BGP messages sent over the connection is called a BGP session. BGP session that spans two Ass is called an external BGP (eBGP) session, and a BGP session between routers in the same AS is called an internal BGP (iBGP) session.

In Figure, the eBGP sessions are shown with the long dashes; the iBGP sessions are shown with the short dashes.BGP session lines in Figure 4.40 do not always correspond to the physical links. BGP allows each AS to learn which destinations are reachable via its neighboring ASs. In BGP, destinations are not hosts but instead are CIDRized prefixes, with each prefix representing a subnet or a collection of subnets.for example, suppose there are four subnets attached to AS2: 138.16.64/24, 138.16.65/24, 138.16.66/24, and 138.16.67/24. Then AS2 could aggregate the prefixes for these four subnets and use BGP to advertise the single prefix to 138.16.64/22 to AS1. As another example, suppose that only the first three of those four subnets are in AS2 and the fourth subnet, 138.16.67/24, is in AS3. Then, routers use longest-prefix matching for forwarding datagrams, AS3 could advertise to AS1 the more specific prefix 138.16.67/24 and AS2 could still advertise to AS1 the aggregated prefix 138.16.64/22. BGP would distribute prefix reachability information over the BGP sessions shown in Figure (1).using the eBGP session between the gateway routers 3a and 1c, AS3 sends AS1 the list of prefixes that are reachable from AS3; and AS1 sends AS3 the list of prefixes that are reachable from AS1. Similarly, AS1 and AS2 exchange prefix

reachability information through their gateway routers 1b and 2a., when a gateway router receives eBGP learned prefixes, the gateway router uses its iBGP sessions to distribute the prefixes to the other routers in the AS. Thus, all the routers in AS1 learn about AS3 prefixes, including the gateway router 1b. The gateway router 1b can therefore re-advertise AS3's prefixes to AS2. When a router learns about a new prefix, it creates an entry for the prefix in its forwarding table.

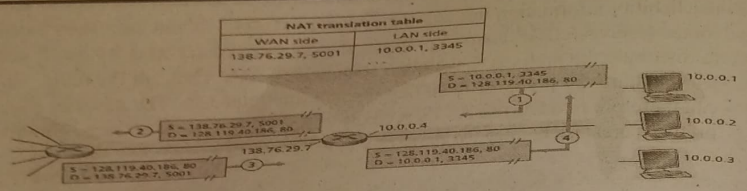


c. Write a note on Network Address Translation? **(04 Marks)**

Ans. To manage IP addresses, larger block of addresses would have to be allocated. There is a simpler approach to address allocation: network address translation (NAT) Figure (1) shows the operation of a NAT-enabled router. The NAT-enabled router, residing in the home, has an interface that is part of the home network on the right of Figure (1). Addressing within the home network is same as above all four interfaces in the home network have the same subnet address of 10.0.0/24. The address space 10.0.0.0/8 is one of three portions of the IP address space that is reserved in [RFC 1918] for a private network or a realm with private addresses, such as the home network in Figure (1) realm with private addresses refers to a network whose addresses only have meaning to devices within that network.

To see the importance, consider the fact that there are hundreds of thousands of home networks, many using the same address space, 10.0.0.0/24. Devices within a given home network can send packets to each other using 10.0.0.0/24 addressing. However, packets forwarded beyond the home network into the larger global Internet clearly cannot use these addresses because there are hundreds of thousands of networks using this block of addresses. That is, the 10.0.0.0/24 addresses can only have meaning within the given home network. But if private addresses only have meaning within a given network, how is addressing handled when packets are sent to or received from the global Internet, where addresses are necessarily unique? The answer lies in understanding NAT.

The NAT-enabled router does not look like a router to the outside world. The NAT router behaves to the outside world as a single device with a single IP address. In Figure (1), all traffic leaving the home router for the larger Internet has a source IP address of 138.76.29.7, and all traffic entering the home router must have a destination address of 138.76.29.7. In essence, the NAT-enabled router is hiding the details of the home network from the outside world. If all datagrams arriving at the NAT router from the WAN have the same destination IP address, then how does the router know the internal host to which it should forward a given datagram? The trick is to use a NAT translation table at the NAT router, and to include port numbers as well as IP addresses in the table entries.



MODULE-4

7. a. What are the initial elements of a mobile network architecture? (6 Marks)

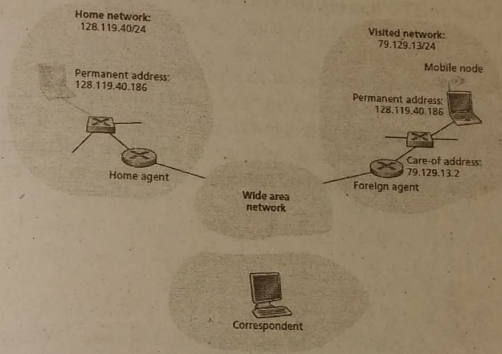
Figure illustrates the concepts, as well as addressing concepts considered below. In Figure (1), note that agents are shown as being collocated with routers), alternatively they could be executing on other hosts or servers in the network. User mobility to be transparent to network applications, it is desirable for a mobile node to keep its address as it moves from one network to another. When a mobile node is resident in a foreign network, all traffic addressed to the node's permanent address now needs to be routed to the foreign network.

One option is for the foreign network to advertise to all other networks that the mobile node is resident in its network. This could be via the usual exchange of intradomain and interdomain routing information and would require few changes to the existing routing infrastructure. The foreign network could simply advertise to its neighbors that it has a highly specific route to the mobile node's permanent address. These neighbors would then propagate this routing information throughout the network as part of the normal procedure of updating routing information and forwarding tables. When the mobile node leaves one foreign network and joins another, the new foreign network would advertise a new, highly specific route to the mobile node, and the old foreign network would withdraw its routing

This solves two problems at once, and it does so without making significant changes to the network-layer infrastructure. Other networks know the location of the mobile node, and it is easy to route datagrams to the mobile node, since the forwarding tables will direct datagrams to the foreign network. A significant drawback, however, is that of scalability. If mobility management were to be the responsibility of network routers, the routers would have to maintain forwarding table entries for potentially millions of mobile nodes, and update these entries as nodes move. An alternative approach is to push mobility functionality from the network core to the network edge a recurring theme in our study of Internet architecture. A natural way to do this is via the mobile node's home network. In much the same way that parents of the mobile twenty something track their child's location, the home agent in the mobile node's home network can track the foreign network in which the mobile node resides. A protocol information regarding the mobile node. Between the mobile node and the home agent will certainly be needed to update the mobile node's location.

Consider the foreign agent in more detail. The conceptually simplest approach, shown in Figure (1), is to locate foreign agents at the edge routers in the foreign network.

One role of the foreign agent is to create a so-called care-of address (COA) for the mobile node, with the network portion of the COA matching that of the foreign network. There are thus two addresses associated with a mobile node, its permanent address and its COA, sometimes known as a foreign address. In the example in Figure (1), the permanent address of the mobile node is 128.119.40.186. When visiting network 79.129.13/24, the mobile node has a COA of 79.129.13.2. A second role of the foreign agent is to inform the home agent that the mobile node is resident in its network and has the given COA. The COA will be used to "reroute" datagrams to the mobile node via its foreign agent. We have separated the functionality of the mobile node and the foreign agent, it is worth noting that the mobile node can also assume the responsibilities of the foreign agent. For example, the mobile node could obtain a COA in the foreign network.



b. How mobility is managed in cellular networks? (06 Marks)

Ans. GSM adopts an indirect routing approach first routing the correspondent's call to the mobile user's home network and from there to the visited network. In GSM terminology, the mobile users's home network is referred to as the mobile user's home public land mobile network.Since the PLMN acronym is a bit of a mouthful, and mindful of our quest to avoid an alphabet soup of acronyms, we'll refer to the GSM home PLMN simply as the home network. The home network is the cellular provider with which the mobile user has a subscription. The visited PLMN, which we'll refer to simply as the visited network, is the network in which the mobile user is currently residing. As in the case of mobile IP, the responsibilities of the home and visited networks are quite different.

- The home network maintains a database known as the home location register (HLR), which contains the permanent cell phone number and subscriber profile information for each of its subscribers. The HLR also contains information about the current locations of these subscribers. That is, if a mobile user is currently roaming in another provider's cellular network, the HLR contains enough information to obtain an address in the visited network to which a call to the

mobile user should be routed. A special switch in the home network, known as the Gateway Mobile services Switching Center (GMSC) is contacted by a correspondent when a call is placed to a mobile user.

- The visited network maintains a database known as the visitor location register (VLR). The VLR contains an entry for each mobile user that is currently in the portion of the network served by the VLR. VLR entries thus come and go as mobile users enter and leave the network. A VLR is usually co-located with the mobile switching center (MSC) that coordinates the setup of a call to and from the visited network. provider's cellular network will serve as a home network for its subscribers and as a visited network for mobile users whose subscription is with a different cellular provider.

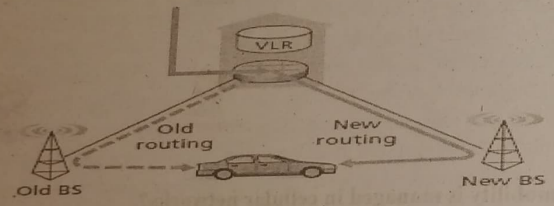
c. Define Handoff? What are the reasons for handoff to occur ? **(04 Marks)**

Ans. A handoff occurs when a mobile station changes its association from one base station to another during a call.

There may be several reasons for handoff to occur, including

(1) The signal between the current base station and the mobile may have deteriorated to such an extent that the call is in danger of being dropped, and

(2) A cell may have become overloaded, handling a large number of calls. This congestion may be alleviated by handing off mobiles to less congested nearby cells.



OR

8. a. What are the key fields in the agent advertisement message? **(10 Marks)**

Ans. The more important fields in the extension are the following:

- Home agent bit (H) - Indicates that the agent is a home agent for the network in which it resides.
- Foreign agent bit (F) - Indicates that the agent is a foreign agent for the network in which it resides.
- Registration required bit (R) - Indicates that a mobile user in this network must register with a foreign agent. In particular, a mobile user cannot obtain a careof address in the foreign network (for example, using DHCP) and assume thefunctionality of the foreign agent for itself, without registering with the foreign agent.
- M, G encapsulation bits - Indicate whether a form of encapsulation other than IPin- IP encapsulation will be used.
- Care-of address (COA) fields - A list of one or more care-of addresses provided by the foreign agent. In our example below, the COA will be associated with the

foreign agent, who will receive datagrams sent to the COA and then forward them to the appropriate mobile node. The mobile user will select one of these addresses as its COA when registering with its home agent.

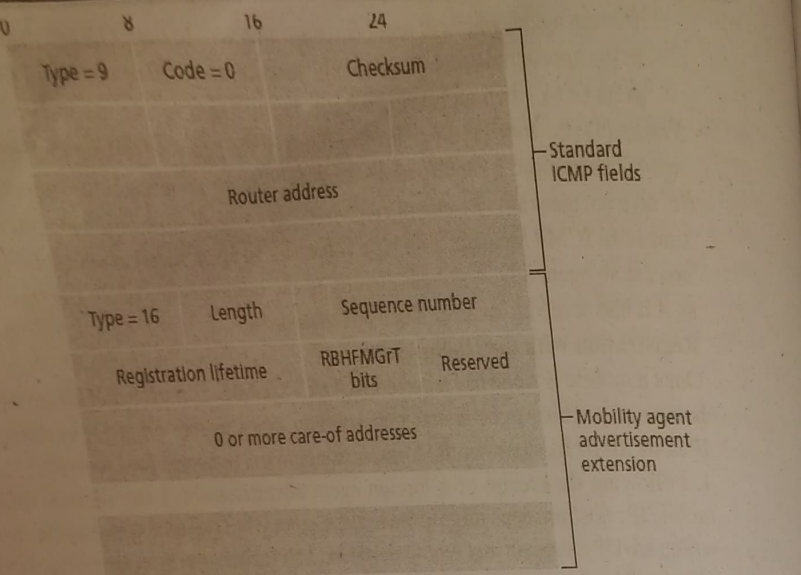
Figure (1) illustrates some of the key fields in the agent advertisement message. With agent solicitation, a mobile node wanting to learn about agents without waiting to receive an agent advertisement can broadcast an agent solicitation message, which is simply an ICMP message with type value 10. An agent receiving the solicitation will unicast an agent advertisement directly to the mobile node, which can then proceed as if it had received an unsolicited advertisement.

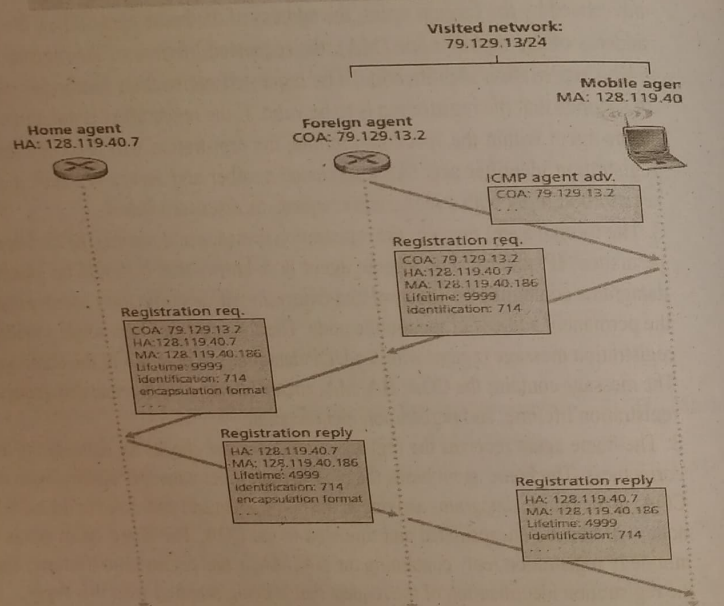
Registration with the Home Agent

Once a mobile IP node has received a COA, that address must be registered with the home agent. This can be done either via the foreign agent or directly by the mobile IP node itself. We consider the former case below. Four steps are involved.

1. Following the receipt of a foreign agent advertisement, a mobile node sends a mobile IP registration message to the foreign agent. The registration message is carried within a UDP datagram and sent to port 434. The registration message carries a COA advertised by the foreign agent, the address of the home agent (HA), the permanent address of the mobile node (MA), the requested lifetime of the registration, and a 64-bit registration identification. The requested registration lifetime is the number of seconds that the registration is to be valid. If the registration is not renewed at the home agent within the specified lifetime, the registration will become invalid. The registration identifier acts like a sequence number and serves to match a received registration reply with a registration request, as discussed below.
2. The foreign agent receives the registration message and records the mobile node's permanent IP address. The foreign agent now knows that it should be looking for datagrams containing an encapsulated datagram whose destination address matches the permanent address of the mobile node. The foreign agent then sends a mobile IP registration message (again, within a UDP datagram) to port 434 of the home agent. The message contains the COA, HA, MA, encapsulation format requested, requested registration lifetime, and registration identification.
3. The home agent receives the registration request and checks for authenticity and correctness. The home agent binds the mobile node's permanent IP address with the COA; in the future, datagrams arriving at the home agent and addressed to the mobile node will now be encapsulated and tunneled to the COA. The home agent sends a mobile IP registration reply containing the HA, MA, actual registration lifetime, and the registration identification of the request that is being satisfied with this reply.
4. The foreign agent receives the registration reply and then forwards it to the mobile node.

At this point, registration is complete, and the mobile node can receive datagrams sent to its permanent address. Figure (2) illustrates these steps. Note that the home agent specifies a lifetime that is smaller than the lifetime requested by the mobile node. A foreign agent need not explicitly deregister a COA when a mobile node leaves its network. This will occur automatically, when the mobile node moves to a new network and registers a new coa





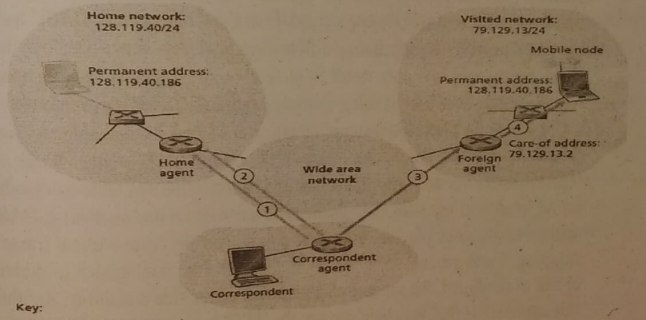
b. Explain Direct routing to a mobile user? **(06 Marks)**

Ans. Direct routing overcomes the inefficiency of triangle routing, but does so at the cost of additional complexity. In the direct routing approach, a correspondent agent in the correspondent's network first learns the COA of the mobile node. This can be done by having the correspondent agent query the home agent, assuming that the mobile node

has an up-to-date value for its COA registered with its home agent. It is also possible for the correspondent itself to perform the function of the correspondent agent, just as a mobile node could perform the function of the foreign agent. This is shown as steps 1 and 2 in Figure (1). The correspondent agent then tunnels datagrams directly to the mobile node's COA, in a manner analogous to the tunnelling performed by the home agent, steps 3 and 4 in Figure (1).

While direct routing overcomes the triangle routing problem, it introduces two important additional challenges:

- A mobile-user location protocol is needed for the correspondent agent to query the home agent to obtain the mobile node's COA.
- When the mobile node moves from one foreign network to another, how will data now be forwarded to the new foreign network? In the case of indirect routing, this problem was easily solved by updating the COA maintained by the home agent. However, with direct routing, the home agent is queried for the COA by the correspondent agent only once, at the beginning of the session. Thus, updating the COA at the home agent, while necessary, will not be enough to solve the problem of routing data to the mobile node's new foreign network. One solution would be to create a new protocol to notify the correspondent of the changing COA. An alternate solution, and one that we'll see adopted in practice in GSM networks, works as follows. Suppose data is currently being forwarded to the mobile node in the foreign network where the mobile node was located when the session first started. We'll identify the foreign agent in that foreign network where the mobile node was first found as the anchor foreign agent. When the mobile node moves to a new foreign network, the mobile node registers with the new foreign agent, and the new foreign agent provides the anchor foreign agent with the mobile node's new COA. When the anchor foreign agent receives an encapsulated datagram for a departed mobile node, it can then re-encapsulate the datagram and forward it to the mobile node using the new COA. If the mobile node later moves yet again to a new foreign network, the foreign agent in that new visited network would then contact the anchor foreign agent in order to set up forwarding to this new foreign network.



MODULE – 5

9. a. Write a note on DASH? **(06 Marks)**

Ans. All clients receive the same encoding of the video, despite the large variations in the amount of bandwidth available to a client, both across different clients and also over time for the same client. This has led to the development of a new type of HTTP-based streaming, often referred to as Dynamic Adaptive Streaming over HTTP (DASH). In DASH, the video is encoded into several different versions, with each version having a different bit rate and, correspondingly, a different quality level. The client dynamically requests chunks of video segments of a few seconds in length from the different versions. When the amount of available bandwidth is high, the client naturally selects chunks from a high-rate version; and when the available bandwidth is low, it naturally selects from a low-rate version. The client selects different chunks one at a time with HTTP GET request messages.

On one hand, DASH allows clients with different Internet access rates to stream in video at different encoding rates. Clients with low-speed 3G connections can receive a low bit-rate version, and clients with fiber connections can receive a high-quality version. On the other hand, DASH allows a client to adapt to the available bandwidth if the end-to-end bandwidth changes during the session. This feature is particularly important for mobile users, who typically see their bandwidth availability fluctuate as they move with respect to the base stations. Comcast, for example, has deployed an adaptive streaming system in which each video source file is encoded into 8 to 10 different MPEG-4 formats, allowing the highest quality video format to be streamed to the client, with adaptation being performed in response to changing network and device conditions.

With DASH, each video version is stored in the HTTP server, each with a different URL. The HTTP server also has a manifest file, which provides a URL for each version along with its bit rate. The client first requests the manifest file and learns about the various versions. The client then selects one chunk at a time by specifying a URL and a byte range in an HTTP GET request message for each chunk. While downloading chunks, the client also measures the received bandwidth and runs a rate determination algorithm to select the chunk to request next. Naturally, if the client has a lot of video buffered and if the measured receive bandwidth is high, it will choose a chunk from a high-rate version. And naturally if the client has little video buffered and the measured received bandwidth is low, it will choose a chunk from a low-rate version. DASH therefore allows the client to freely switch among different quality levels. Since a sudden drop in bit rate by changing versions may result in noticeable visual quality degradation, the bit-rate reduction may be achieved using multiple intermediate versions to smoothly transition to a rate where the client's consumption rate drops below its available receive bandwidth. When the network conditions improve, the client can then later choose chunks from higher bit-rate versions. By dynamically monitoring the available bandwidth and client buffer level, and adjusting the transmission rate with version switching, DASH can often achieve continuous playout at the best possible quality level without frame freezing or skipping. Furthermore, since the client maintains the intelligence to determine which chunk to send next, the scheme also improves server-side scalability. Another benefit of this approach is that the client can use the HTTP byte-range request to precisely control the amount of prefetched video that it buffers locally. Each audio version has its own

quality level and bit rate and has its own URL. The client dynamically selects both video and audio chunks, and locally synchronizes audio and video playout.

b. Define Expedited forwarding PHB and Assured forwarding PHB? (04 Marks)

Ans. The **expedited forwarding PHB** specifies that the departure rate of a class of traffic from a router must equal or exceed a configured rate.

The **assured forwarding PHB** divides traffic into four classes, where each AF class is guaranteed to be provided with some minimum amount of bandwidth and buffering.

c. Explain any 2 Scheduling Mechanism?

Ans. **First-In-First-Out (FIFO)** **(06 Marks)**

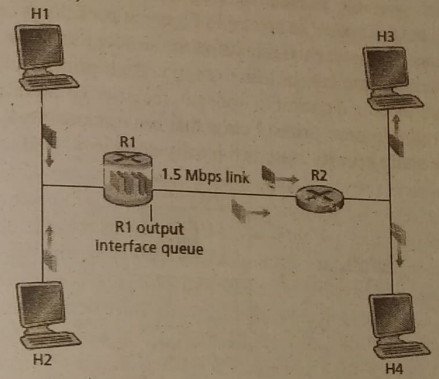
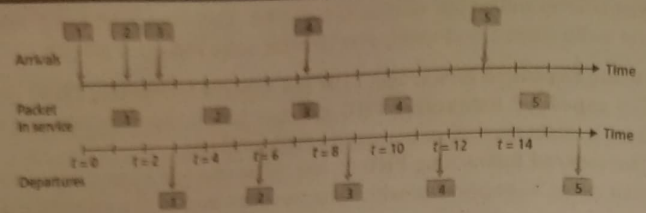


Figure shows the queuing model abstractions for the FIFO link-scheduling discipline. Packets arriving at the link output queue wait for transmission if the link is currently busy transmitting another packet. If there is not sufficient buffering space to hold the arriving packet, the queue's packet-discarding policy then determines whether the packet will be dropped (lost) or whether other packets will be removed from the queue to make space for the arriving packet. we will ignore packet discard. When a packet is completely transmitted over the outgoing link (that is, receives service) it is removed from the queue. The FIFO (also known as first-come-first-served, or FCFS) scheduling discipline selects packets for link transmission in the same order in which they arrived at the output link queue. FIFO queuing from bus stops or other service centers, where arriving customers join the back of the single waiting line, remain in order, and are then served when they reach the front of the line.

Figure shows the FIFO queue in operation. Packet arrivals are indicated by numbered arrows above the upper timeline, with the number indicating the order in which the packet arrived. Individual packet departures are shown below the lower timeline. The time that a packet spends in service (being transmitted) is indicated by the shaded rectangle between the two timelines. Because of the FIFO discipline, packets leave in the same order in which they arrived. Note that after the departure of packet 4, the link remains idle until the arrival of packet 5.



Priority Queuing

Under priority queuing, packets arriving at the output link are classified into priority classes at the output queue, as shown in Figure. a packet's priority class may depend on an explicit marking that it carries in its packet header its source or destination IP address, its destination port number, or other criteria. Each priority class typically has its own queue. When choosing a packet to transmit, the priority queuing discipline will transmit a packet from the highest priority class that has a nonempty queue. The choice among packets in the same priority class is typically done in a FIFO manner.

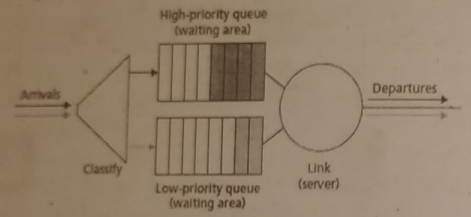
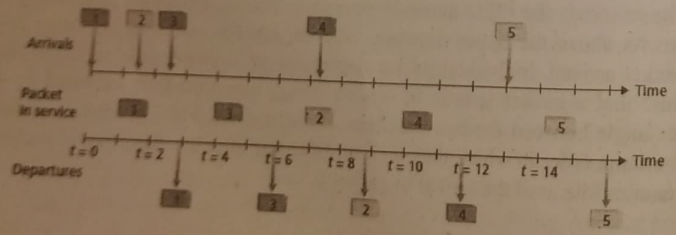


Figure illustrates the operation of a priority queue with two priority classes. Packets 1, 3, and 4 belong to the high-priority class, and packets 2 and 5 belong to the low-priority class. Packet 1 arrives and, finding the link idle, begins transmission. During the transmission of packet 1, packets 2 and 3 arrive and are queued in the low- and high-priority queues, respectively. After the transmission of packet 1, packet 3 (a high-priority packet) is selected for transmission over packet 2 (which, even though it arrived earlier, is a low-priority packet). At the end of the transmission of packet 3, packet 2 then begins transmission. Packet 4 (a high-priority packet) arrives during the transmission of packet 2 (a low-priority packet). Under a nonpreemptive priority queuing discipline, the transmission of a packet is not interrupted once it has begun. In this case, packet 4 queues for transmission and begins being transmitted after the transmission of packet 2 is completed.



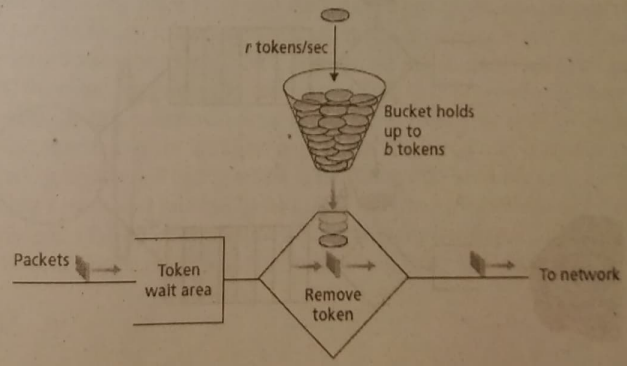
OR

10. a. Explain Policing Mechanism? **(10 Marks)**

Ans. The rate at which a class or flow is allowed to inject packets into the network, is an important QoS mechanism. The three important policing criteria, each differing from the other according to the time scale over which the packet flow is policed:

- **Average rate.** The network may wish to limit the long-term average rate at which a flow's packets can be sent into the network. A issue here is the interval of time over which the average rate will be policed. A flow whose average rate is limited to 100 packets per second is more constrained than a source that is limited to 6,000 packets per minute, even though both have the same average rate over a long enough interval of time. For example, the latter constraint would allow a flow to send 1,000 packets in a given second-long interval of time, while the former constraint would disallow this sending behavior.
- **Peak rate.** While the average-rate constraint limits the amount of traffic that can be sent into the network over a relatively long period of time, a peak-rate constraint limits the maximum number of packets that can be sent over a shorter period of time. Using our example above, the network may police a flow at an average rate of 6,000 packets per minute, while limiting the flow's peak rate to 1,500 packets per second.
- **Burst size.** The network may also wish to limit the maximum number of packets (the "burst" of packets) that can be sent into the network over an extremely short interval of time. In the limit, as the interval length approaches zero, the burst size limits the number of packets that can be instantaneously sent into the network. Even though it is physically impossible to instantaneously send multiple packets into the network, the abstraction of a maximum burst size is a useful one.

The leaky bucket mechanism is an abstraction that can be used to characterize these policing limits. In Figure 1, a leaky bucket consists of a bucket that can hold up to b tokens. Tokens are added to this bucket as follows. New tokens, which may potentially be added to the bucket, are always being generated at a rate of r tokens per second. (We assume here for simplicity that the unit of time is a second.) If the bucket is filled with less than b tokens when a token is generated, the newly generated token is added to the bucket; otherwise the newly generated token is ignored, and the token bucket remains full with b tokens. Let us now consider how the leaky bucket can be used to police a packet flow.

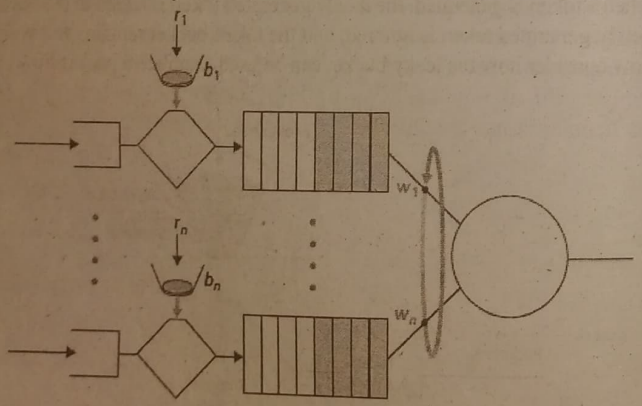


Suppose that before a packet is transmitted into the network, it must first remove a token from the token bucket. If the token bucket is empty, the packet must wait for a token. Let us now consider how this behavior polices a traffic flow. Because there can be at most b tokens in the bucket, the maximum burst size for a leaky-bucket policed flow is b packets. Furthermore, because the token generation rate is r, the maximum number of packets that can enter the network of any interval of time of length t is rt + b. Thus, the token-generation rate, r, serves to limit the long-term average rate at which packets can enter the network. It is also possible to use leaky buckets (specifically, two leaky buckets in series) to police a flow's peak rate in addition to the longterm average rate.

Leaky Bucket + Weighted Fair Queuing = Provable Maximum Delay in a Queue

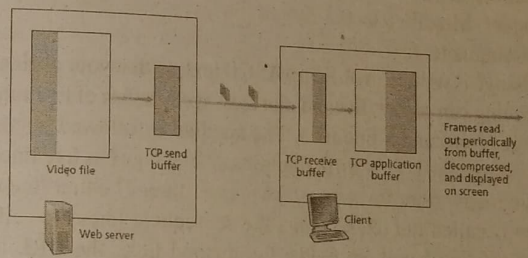
Consider a router's output link that multiplexes n flows, each policed by a leaky bucket with parameters bi and ri, i = 1, . . . , n, using WFQ scheduling. We use the term flow here loosely to refer to the set of packets that are not distinguished from each other by the scheduler. In practice, a flow might be comprised of traffic from a single end to- end connection or a collection of many such connections, Figure. WFQ that each flow, i, is guaranteed to receive a share of the link bandwidth equal to at least R · wi/(Σwj), where R is the transmission rate of the link in packets/sec. What then is the maximum delay that a packet will experience while waiting for service in the WFQ (that is, after passing through the leaky bucket)? Let us focus on flow 1. Suppose that flow 1's token bucket is initially full. A burst of b1 packets then arrives to the leaky bucket policer for flow 1. These packets remove all of the tokens (without wait) from the leaky bucket and then join the WFQ waiting area for flow 1. Since these b1 packets are served at a rate of at least R · wi /(Σwj) packet/sec, the last of these packets will then have a maximum delay, dmax, until its transmission is completed, where The rationale behind this formula is that if there are b1 packets in the queue and packets are being serviced (removed) from the queue at a rate of at least R · w1/ (Σwj) packets per second, then the amount of time until the last bit of the last packet is transmitted cannot be more than b1/(R · w1/(Σwj)).

$$d_{max} = \frac{b_1}{R.w_1 / \sum w_j}$$



b. Explain interaction between client and server for HTTP Streaming ? (06 Marks)

Ans. Figure (1) illustrates the interaction between client and server for HTTP streaming. At the server side, the portion of the video file in white has already been sent into the server's socket, while the darkened portion is what remains to be sent. After "passing through the socket door," the bytes are placed in the TCP send buffer before being transmitted into the Internet. Because the TCP send buffer is shown to be full, the server is momentarily prevented from sending more bytes from the video file into the socket. On the client side, the client application (media player) reads bytes from the TCP receive buffer (through its client socket) and places the bytes into the client application buffer. At the same time, the client application periodically grabs video frames from the client application buffer, decompresses the frames, and displays them on the user's screen. Note that if the client application buffer is larger than the video file, then the whole process of moving bytes from the server's storage to the client's application buffer is equivalent to an ordinary file download over HTTP—the client simply pulls the video off the server as fast as TCP will allow!



Consider now what happens when the user pauses the video during the streaming process. During the pause period, bits are not removed from the client application buffer, even though bits continue to enter the buffer from the server. If the client application buffer is finite, it may eventually become full, which will cause "back pressure" all the way back to the server. Specifically, once the client application buffer becomes full, bytes can no longer be removed from the client TCP receive buffer, so it too becomes full. Once the client receive TCP buffer becomes full, bytes can no longer be removed from the client TCP send buffer, so it also becomes full. Once the TCP send buffer becomes full, the server cannot send any more bytes into the socket. Thus, if the user pauses the video, the server may be forced to stop transmitting, in which case the server will be blocked until the user resumes the video.In fact, even during regular playback (that is, without pausing), if the client application buffer becomes full, back pressure will cause the TCP buffers to become full, which will force the server to reduce its rate. To determine the resulting rate, note that when the client application removes f bits, it creates room for f bits in the client application buffer, which in turn allows the server to send for additional bits. Thus, the server send rate can be no higher than the video consumption rate at the client. Therefore, a full client application buffer indirectly imposes a limit on the rate that video can be sent from server to client when streaming over HTTP.

Fifth Semester B.E. Degree Examination

CBCS - Model Question Paper - 2

COMPUTER NETWORKS

Time: 3 hrs. Max. Marks: 80

Note : *Answer any FIVE full questions, selecting ONE full question from each module.*

MODULE - 1

1. a. Explain the general format of an Http request message? (10 Marks)

Ans. HTTP Request Message - a typical HTTP request message:

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
```

The message is written in ordinary ASCII text, so that your ordinary computer-literate human being can read it. Second, the message consists of five lines, each followed by a carriage return and a line feed. The last line is followed by an additional carriage return and line feed. This particular request message has five lines, a request message can have many more lines or as few as one line. The first line of an HTTP request message is called the request line; the subsequent lines are called the header lines. The request line has three fields: the method field, the URL field, and the HTTP version field. The method field can take on several different values, including GET, POST, HEAD, PUT, and DELETE. The great majority of HTTP request messages use the GET method. The GET method is used when the browser requests an object, with the requested object identified in the URL field.

In this example, the browser is requesting the object /somedir/page.html. The version is self-explanatory; in this example, the browser implements version HTTP/1.1. The header lines in the example. The header line Host: www.someschool.edu specifies the host on which the object resides. There is already a TCP connection in place to the host. The information provided by the host header line is required by Web proxy caches. By including the Connection: close header line, the browser is telling the server that it doesn't want to bother with persistent connections; it wants the server to close the connection after sending the requested object. The User-agent: header line specifies the user agent, that is, the browser type that is making the request to the server. Here the user agent is Mozilla/5.0, a Firefox browser. This header line is useful because the server can actually send different versions of the same object to different types of user agents.Finally, the Acceptlanguage: header indicates that the user prefers to receive a French version of the object, if such an object exists on the server; otherwise, the server should send its default version. The Accept-language: header is just one of many content negotiation headers available in HTTP.

The general format of a request message, as shown in Figure (1). After the header lines there is an "entity body." The entity body is empty with the GET method, but is used with the POST method. An HTTP client often uses the POST method when the user fills out a form—for example, when a user provides search words to a search engine. With a POST message, the user is still requesting a Web page from the server, but the specific contents of the Web page depend on what the user entered into the form fields. If the value of the method field is POST, then the entity body contains what the user entered into the form fields.HTML forms often use the GET method and include the inputted data in the requested URL. For example, if a form uses the GET method, has two fields, and the inputs to the two fields are monkeys and bananas, then the URL will have the structure www.somesite.com/animalsearch?monkeys&bananas. In day-today Web surfing, you have probably noticed extended URLs of this sort. The HEAD method is similar to the GET method. When a server receives a request with the HEAD method, it responds with an HTTP message but it leaves out the requested object. Application developers often use the HEAD method for debugging. The PUT method is often used in conjunction with Web publishing tools. It allows a user to upload an object to a specific path on a specific Web server. The PUT method is also used by applications that need to upload objects to Web servers. The DELETE method allows a user, or an application, to delete an object on a Web server.

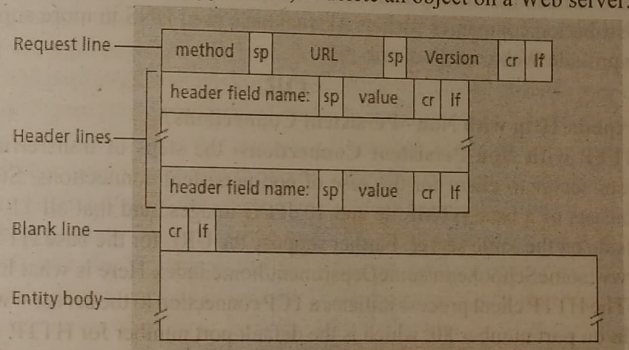


Figure (1) - General format of an HTTP request message.

b. What are theDNS Services used in translating host names to IP addresses? (06 Marks)

Ans. DNS provides a few other important services in addition to translating hostnames to IP addresses:

(1) Host aliasing - A host with a complicated hostname can have one or more alias names. For example, a hostname such as relay1.west-coast.enterprise. com could have, say, two aliases such as enterprise.com and www.enterprise.com. In this case, the hostname relay1.westcoast. enterprise.com is said to be a canonical hostname. Alias hostnames, when present, are typically more mnemonic than canonical hostnamesDNS can be invoked by an application to obtain the canonical hostname for a supplied alias hostname as well as the IP address of the host.

(2) Mail server aliasing - it is highly desirable that e-mail addresses be mnemonic. For example, if Bob has an account with Hotmail, Bob's e-mail address might be as simple as bob@hotmail.com.the hostname of the Hotmail mail server is more complicated and much less mnemonic than simply hotmail.com. DNS can be invoked by a mail application to obtain the canonical hostname for a supplied alias hostname as well as the IP address of the host. In fact, the MX record permits a company's mail server and Web server to have identical hostnames; for example, a company's Web server and mail server can both be called enterprise.com.

(3) Load distribution- DNS is also used to perform load distribution among replicated servers, such as replicated Web servers. Busy sites, such as cnn.com, are replicated over multiple servers, with each server running on a different end system and each having a different IP address. For replicated Web servers, a set of IP addresses is thus associated with one canonical hostname. The DNS database contains this set of IP addresses. When clients make a DNS query for a name mapped to a set of addresses, the server responds with the entire set of IP addresses, but rotates the ordering of the addresses within each reply. Because a client typically sends its HTTP request message to the IP address that is listed first in the set, DNS rotation distributes the traffic among the replicated servers. DNS rotation is also used for e mail so that multiple mail servers can have the same alias name. Also, content distribution companies such as Akamai have used DNS in more sophisticated ways to provide Web content distribution

OR

2. a. Explain Http with Non –Persistent Connections? **(10 Marks)**

Ans. HTTP with Non-Persistent Connections- the steps of transferring a Web page from server to client for the case of non-persistent connections. Suppose the page consists of a base HTMLfile and 10 JPEG images, and that all 11 of these objects reside on the same server. Further suppose the URL for the base HTML file is http://www.someSchool.edu/someDepartment/home.index Here is what happens:

1.The HTTP client process initiates a TCP connection to the server www.someSchool.edu on port number 80, which is the default port number for HTTP. Associated with the TCP connection, there will be a socket at the client and a socket at the server.

2.The HTTP client sends an HTTP request message to the server via its socket. The request message includes the path name /someDepartment/home.index.

3.The HTTP server process receives the request message via its socket, retrieves the object /someDepartment/home.index from its storage (RAM or disk), encapsulates the object in an HTTP response message, and sends the response message to the client via its socket.

4.The HTTP server process tells TCP to close the TCP connection.

5.The HTTP client receives the response message. The TCP connection terminates. The message indicates that the encapsulated object is an HTML file. The client extracts the file from the response message, examines the HTML file, and finds references to the 10 JPEG objects.

6.The first four steps are then repeated for each of the referenced JPEG objects.

As the browser receives the Web page, it displays the page to the user. Two different browsers may interpret a Web page in somewhat different ways. HTTP has nothing to do with how a Web page is interpreted by a client. The HTTP specifications ([RFC 1945] and [RFC 2616]) define only the communication protocol between the client HTTP program and the server HTTP program. The steps above illustrate the use of non-persistent connections, where each TCP connection is closed after the server sends the object the connection does not persist for other objects. Note that each TCP connection transports exactly one request message and one response message. Thus, in this example, when a user requests the Web page, 11 TCP connections are generated. In the steps described above, we were intentionally vague about whether the client obtains the 10 JPEGs over 10 serial TCP connections, or whether some of the JPEGs are obtained over parallel TCP connections. Users can configure modern browsers to control the degree of parallelism. In their default modes, most browsers open 5 to 10 parallel TCP connections, and each of these connections handles one request-response transaction. If the user prefers, the maximum number of parallel connections can be set to one, in which case the 10 connections are established serially. The use of parallel connections shortens the response time. Calculation to estimate the amount of time that elapses from when a client requests the base HTML file until the entire file is received by the client. We define the round-trip time (RTT), which is the time it takes for a small packet to travel from client to server and then back to the client. The RTT includes packet-propagation delays, packet queuing delays in intermediate routers and switches, and packet-processing delays. Consider what happens when a user clicks on a hyperlink. Figure (1) this causes the browser to initiate a TCP connection between the browser and the Web server; this involves a "three-way handshake" the client sends a small TCP segment to the server, the server acknowledges and responds with a small TCP segment, and, finally, the client acknowledges back to the server. The first two parts of the three-way handshake take one RTT. After completing the first two parts of the handshake, the client sends the HTTP request message combined with the third part of the three-way handshake into the TCP connection. Once the request message arrives at the server, the server sends the HTML file into the TCP connection. This HTTP request/response eats up another RTT. Thus, roughly, the total response time is two RTTs plus the transmission time at the server of the HTML file.

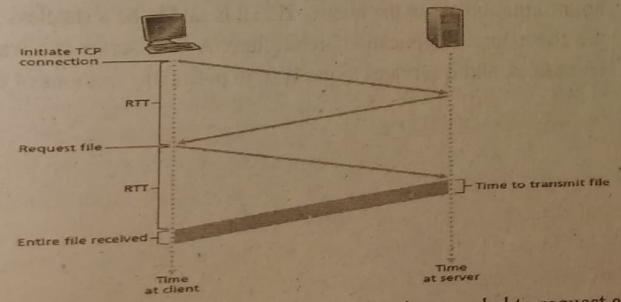
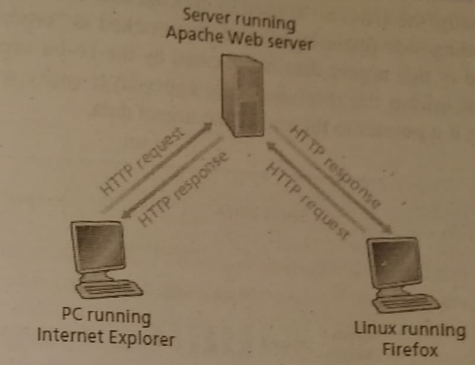


Figure (1) Back-of-the-envelope calculation for the time needed to request and receive an HTML file

b. Write Briefly on Http? **(06 Marks)**

Ans. The Hypertext Transfer Protocol (HTTP), the Web's application-layer protocol, is at the heart of the Web. HTTP is implemented in two programs: a client program and a server program. The client program and server program, executing on different end systems, talk to each other by exchanging HTTP messages.

HTTP defines the structure of these messages and how the client and server exchange the messages. HTTP defines how Web clients request Web pages from Web servers and how servers transfer Web pages to clients. The interaction between client and server in general idea is illustrated in Figure (1). When a user requests a Web page, the browser sends HTTP request messages for the objects in the page to the server. The server receives the requests and responds with HTTP response messages that contain the objects. HTTP uses TCP as its underlying transport protocol. The HTTP client first initiates a TCP connection with the server. Once the connection is established, the browser and the server processes access TCP through their socket interfaces. On the client side the socket interface is the door between the client process and the TCP connection; on the server side it is the door between the server process and the TCP connection. The client sends HTTP request messages into its socket interface and receives HTTP response messages from its socket interface. Similarly, the HTTP server receives request messages from its socket interface and sends response messages into its socket interface. Once the client sends a message into its socket interface, the message is out of the client's hands and is "in the hands" of TCP. TCP provides a reliable data transfer service to HTTP. This implies that each HTTP request message sent by a client process eventually arrives intact at the server; each HTTP response message sent by the server process eventually arrives intact at the client. The great advantages of a layered architecture HTTP need not worry about lost data or the details of how TCP recovers from loss or reordering of data within the network. That is the job of TCP and the protocols in the lower layers of the protocol stack. It is important to note that the server sends requested files to clients without storing any state information about the client. If a particular client asks for the same object twice in a period of a few seconds, the server does not respond by saying that it just served the object to the client; instead, the server resends the object, as it has completely forgotten what it did earlier. Because an HTTP server maintains no information about the clients, HTTP is said to be a stateless protocol. Web uses the client-server application architecture. A Web server is always on, with a fixed IP address, and it services requests from potentially millions of different browsers.



MODULE-2

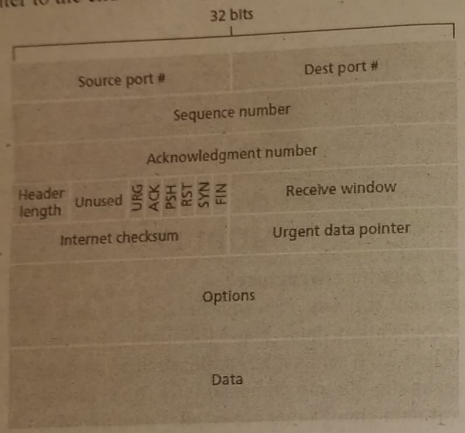
3. a. Explain TCP Segment Structure? **(10 Marks)**

Ans. The TCP segment consists of header fields and a data field. The data field contains a chunk of application data. The MSS limits the maximum size of a segment's data field. When TCP sends a large file, such as an image as part of a Web page, it typically breaks the file into chunks of size MSS. Interactive applications, however, often transmit data chunks that are smaller than the MSS; for example, with remote login applications like Telnet, the data field in the TCP segment is often only one byte. Because the TCP header is typically 20 bytes segments sent by Telnet may be only 21 bytes in length. Figure (1) shows the structure of the TCP segment. with UDP, the header includes source and destination port numbers, which are used for multiplexing/demultiplexing data from/to upper-layer applications. with UDP, the header includes a checksum field.

A TCP segment header also contains the following fields:

- The 32-bit sequence number field and the 32-bit acknowledgment number field are used by the TCP sender and receiver in implementing a reliable data transfer service.
- The 16-bit receive window field is used for flow control. it is used to indicate the number of bytes that a receiver is willing to accept.
- The 4-bit header length field specifies the length of the TCP header in 32-bit words. The TCP header can be of variable length due to the TCP options field.
- The optional and variable-length options field is used when a sender and receiver negotiate the maximum segment size (MSS) or as a window scaling factor for use in high-speed networks. A time-stamping option is also defined.
- The flag field contains 6 bits. The ACK bit is used to indicate that the value carried in the acknowledgment field is valid; that is, the segment contains an acknowledgment for a segment that has been successfully received. The RST, SYN, and FIN bits are used for connection setup and teardown. Setting the PSH bit indicates that the receiver should pass the data to the upper layer immediately.

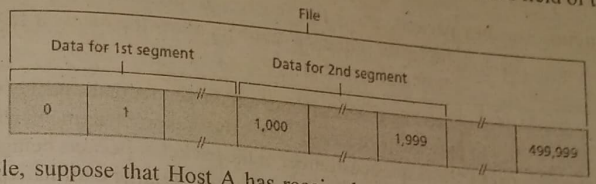
Finally, the URG bit is used to indicate that there is data in this segment that the sending-side upper-layer entity has marked as "urgent." The location of the last byte of this urgent data is indicated by the 16-bit urgent data pointer field. TCP must inform the receiving- side upper-layer entity when urgent data exists and pass it a pointer to the end of the urgent data.



Sequence Numbers and Acknowledgment Numbers -important fields in the TCP segment header are the sequence number field and the acknowledgment number field. These fields are a critical part of TCP's reliable data transfer service TCP views data as an unstructured, but ordered, stream of bytes. TCP's use of sequence numbers reflects this view in that sequence numbers are over the stream of transmitted bytes and not over the series of transmitted segments. The sequence number for a segment is therefore the byte-stream number of the first byte in the segment. Example. Suppose that a process in Host A wants to send a stream of data to a process in Host B over a TCP connection. The TCP in Host A will implicitly number each byte in the data stream. Suppose that the data stream consists of a file consisting of 500,000 bytes, that the MSS is 1,000 bytes, and that the first byte of the data stream is numbered 0. As shown in Figure (2), TCP constructs 500 segments out of the data stream.

The first segment gets assigned sequence number 0, the second segment gets assigned sequence number 1,000, and the third segment gets assigned sequence number 2,000. Each sequence number is inserted in the sequence number field in the header of the appropriate TCP segment. Consider acknowledgment numbers. These are a little trickier than sequence numbers. TCP is full-duplex, so that Host A may be receiving data from Host B while it sends data to Host B. Each of the segments that arrive from Host B has a sequence number for the data flowing from B to A. The acknowledgment number that Host A puts in its segment is the sequence number of the next byte Host A is expecting from Host B. Suppose that Host A has received all bytes numbered 0 through 535 from B and suppose that it is about to send a segment

to Host B. Host A is waiting for byte 536 and all the subsequent bytes in Host B's data stream. So Host A puts 536 in the acknowledgment number field of the segment it sends to B.



example, suppose that Host A has received one segment from Host B containing bytes 0 through 535 and another segment containing bytes 900 through 1,000. For some reason Host A has not yet received bytes 536 through 899. In this example, Host A is still waiting for byte 536 in order to re-create B's data stream. Thus, A's next segment to B will contain 536 in the acknowledgment number field. Because TCP only acknowledges bytes up to the first missing byte in the stream, TCP is said to provide cumulative acknowledgments. Host A received the third segment (bytes 900 through 1,000) before receiving the second segment (bytes 536 through 899). Thus, the third segment arrived out of order. There are basically two choices: either (1) the receiver immediately discards out-of-order segments (2) the receiver keeps the out-of-order bytes and waits for the missing bytes to fill in the gaps. In Figure (2), we assumed that the initial sequence number was zero. both sides of a TCP connection randomly choose an initial sequence number. This is done to minimize the possibility that a segment that is still present in the network from an earlier, already-terminated connection between two hosts is mistaken for a valid segment in a later connection between these same two hosts.

b. Give the summary of reliable data transfer mechanisms and their use?

(06 Marks)

Ans.

Mechanism	Use, Comments
Checksum	Used Id detect bit errors in a transmitted packet.
Timer	Used Id timeout/retransmit a packet, possibly because the poctel (or its Mi was lost with in the channel. Because timeouts can occur when o packet is delayed but not lost (premature timeout), or when a packet hos teen received by the receiver but the receiver-to-sender ACK bcs been lost, duplicate copies of n packet my be received by a receiver.

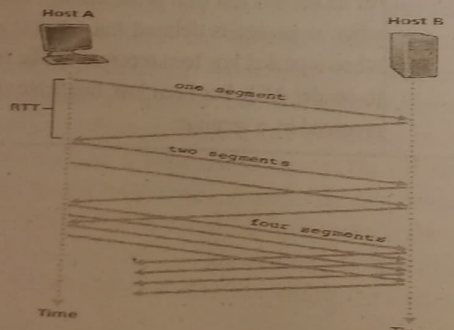
Sequence number	Used for sequential numbering of packets of data flowing from sender to receiver, Gaps in me sequence numbers of received packets allow lite receiver to detect a lost pocket. Packets with duplicate sequence numbers allow the receiver Id detect duplicate copies of a packet
Acknowledgment	Used by the receiver to tell the sender that a packet or set of packets has been received correctly, Acknowledgment will typically can the sequence number of the packet or packets being acknowledged. Acknowledgment may be individual or cumulative, depending cm the protocol.
Negative acknowledgment	Used by the receiver to tell the sender that a packet has not been received correctly. Negative acknowledgments will typically carry the sequence number of the packet that was not received correctly.
Window, pipelining	The sender may be restricted to sending only packets with sequence numbers That fall within a given range. By allowing multiple packets to be transmitted but not yet acknowledged, sender utilization can be increased over of stup-and-wait mode of operation. Well see shortly that the window size may be set on the basis of the received ability to receive and buffer messages, or the level of congestion in the network, or both.

OR

4. a. Explain TCP Congestion –Control algorithm **(10 Marks)**

Ans. **TCP congestion-control algorithm** which was first described in [Jacobson 1988] and is standardized in [RFC 5681]. The algorithm has three major components: (1) slow start, (2) congestion avoidance, and (3) fast recovery. Slow start and congestion avoidance are mandatory components of TCP, differing in how they increase the size of cwnd in response to received ACKs. shortly that slow start increases the size of cwnd more rapidly than congestion avoidance. Fast recovery is recommended, but not required, for TCP senders.

Slow Start



When a TCP connection begins, the value of cwnd is typically initialized to a small value of 1 MSS [RFC 3390], resulting in an initial sending rate of roughly MSS/RTT. For example, if MSS = 500 bytes and RTT = 200 msec, the resulting initial sending rate is only about 20 kbps. Since the available bandwidth to the TCP sender may be much larger than MSS/RTT, the TCP sender would like to find the amount of available bandwidth quickly. Thus, in the slow-start state, the value of cwnd begins at 1 MSS and increases by 1 MSS every time a transmitted segment is first acknowledged. In the example of Figure, TCP sends the first segment into the network and waits for an acknowledgment. When this acknowledgment arrives, the TCP sender increases the congestion window by one MSS and sends out two maximum-sized segments. These segments are then acknowledged, with the sender increasing the congestion window by 1 MSS for each of the acknowledged segments, giving a congestion window of 4 MSS, and so on. This process results in a doubling of the sending rate every RTT. Thus, the TCP send rate starts slow but grows exponentially during the slow start phase..

First, if there is a loss event (i.e., congestion) indicated by a timeout, the TCP sender sets the value of cwnd to 1 and begins the slow start process anew. It also sets the value of a second state variable, ssthresh to cwnd/2 half of the value of the congestion window value when congestion was detected.

second way in which slow start may end is directly tied to the value of ssthresh. Since ssthresh is half the value of cwnd when congestion was last detected, it might be a bit reckless to keep doubling cwnd when it reaches or surpasses the value of ssthresh. Thus, when the value of cwnd equals ssthresh, slow start ends and TCP transitions into congestion avoidance mode. TCP increases cwnd more cautiously when in congestion-avoidance mode. The final way in which slow start can end is if three duplicate ACKs are detected, in which case TCP performs a fast retransmit and enters the fast recovery state.. TCP's behavior in slow start is summarized in the FSM description of TCP congestion control in Figure (2). The slow-start algorithm traces it roots to [Jacobson 1988]; an approach similar to slow start was also proposed independently.

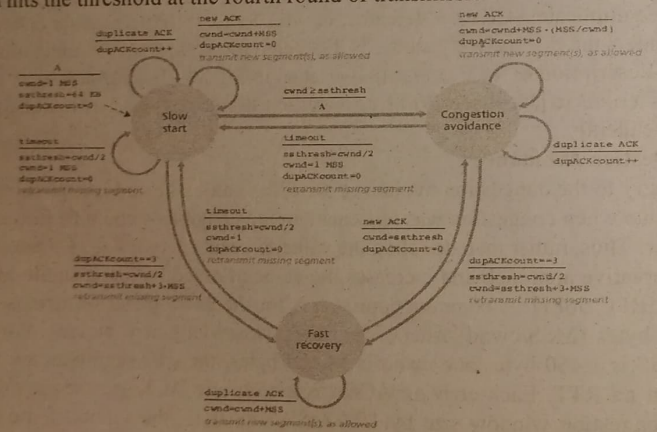
Congestion Avoidance

On entry to the congestion-avoidance state, the value of cwnd is approximately half its value when congestion was last encountered congestion could be just around the corner! Thus, rather than doubling the value of cwnd every RTT, TCP adopts a more conservative approach and increases the value of cwnd by just a single MSS every RTT [RFC 5681]. A common approach is for the TCP sender to increase cwnd by MSS bytes (MSS/cwnd) whenever a new acknowledgment arrives. For example, if MSS is 1,460 bytes and cwnd is 14,600 bytes, then 10 segments are being sent within an RTT. Each arriving ACK (assuming one ACK per segment) increases the congestion window size by 1/10 MSS, and thus, the value of the congestion window will have increased by one MSS after ACKs when all 10 segments have been received. But when should congestion avoidance's linear increase end? TCP's congestion-avoidance algorithm behaves the same when a timeout occurs. As in the

case of slow start: The value of cwnd is set to 1 MSS, and the value of ssthresh is updated to half the value of cwnd when the loss event occurred. That a loss event also can be triggered by a triple duplicate ACK event. In this case, the network is continuing to deliver segments from sender to receiver So TCP's behavior to this type of loss event should be less drastic than with a timeout-indicated loss: TCP halves the value of cwnd and records the value of ssthresh to be half the value of cwnd when the triple duplicate ACKs were received. The fast-recovery state is then entered.

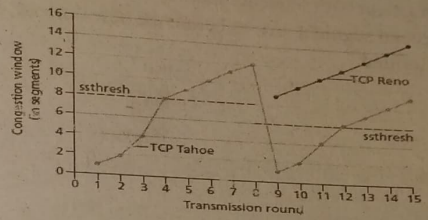
Fast Recovery

In fast recovery, the value of cwnd is increased by 1 MSS for every duplicate ACK received for the missing segment that caused TCP to enter the fast-recovery state. Eventually, when an ACK arrives for the missing segment, TCP enters the congestion-avoidance state after deflating cwnd. If a timeout event occurs, fast recovery transitions to the slow-start state after performing the same actions as in slow start and congestion avoidance: The value of cwnd is set to 1 MSS, and the value of ssthresh is set to half the value of cwnd when the loss event occurred. Fast recovery is a recommended, but not required, component of TCP [RFC 5681]. It is interesting that an early version of TCP, known as TCP Tahoe, unconditionally cut its congestion window to 1 MSS and entered the slow-start phase after either a timeout-indicated or triple-duplicate-ACK-indicated loss event. The newer version of TCP, TCP Reno, incorporated fast recovery. Figure (3) illustrates the evolution of TCP's congestion window for both Reno and Tahoe. In this figure, the threshold is initially equal to 8 MSS. For the first eight transmission rounds, Tahoe and Reno take identical actions. The congestion window climbs exponentially fast during slow start and hits the threshold at the fourth round of transmission.



The congestion window then climbs linearly until a triple duplicate- ACK event occurs, just after transmission round 8. Note that the congestion window is 12 • MSS when this loss event occurs. The value of ssthresh is then set to 0.5 • cwnd = 6

• MSS. Under TCP Reno, the congestion window is set to cwnd = 6 • MSS and then grows linearly. Under TCP Tahoe, the congestion window is set to 1 MSS and grows exponentially until it reaches the value of ssthresh, at which point it grows linearly. Figure (2) presents the complete FSM description of TCP's congestion control algorithms—slow start, congestion avoidance, and fast recovery. The figure also indicates where transmission of new segments or retransmitted segments can occur. Although it is important to distinguish between TCP error control/retransmission and TCP congestion control, it's also important to appreciate how these two aspects of TCP are inextricably linked.



b. Explain few Scenarios of a reliable data transfer? (6 Marks)

Ans. Few Scenarios of a reliable data transfer are:

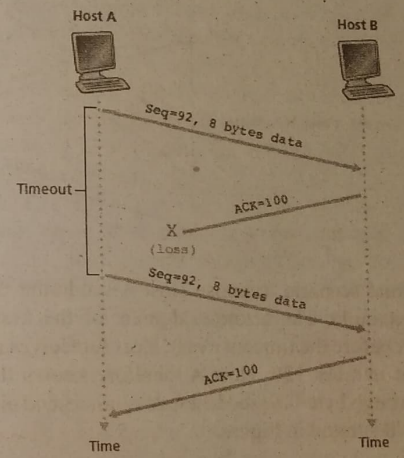
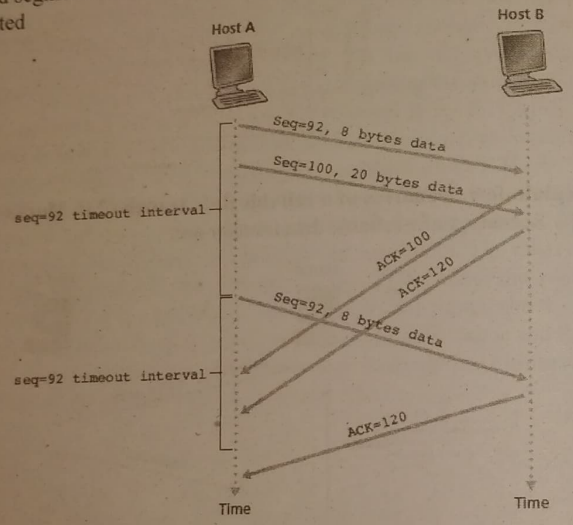
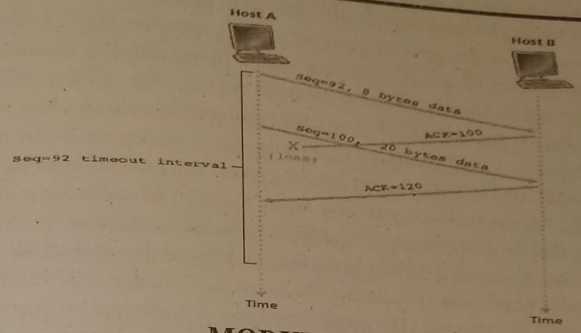


Figure depicts the first scenario, in which Host A sends one segment to Host B. Suppose that this segment has sequence number 92 and contains 8 bytes of data. After sending this segment, Host A waits for a segment from B with acknowledgment number 100. Although the segment from A is received at B, the acknowledgment from B to A gets lost. In this case, the timeout event occurs, and Host A retransmits the same segment. Of course, when Host B receives the retransmission, it observes from the sequence number that the segment contains data that has already been

received. Thus, TCP in Host B will discard the bytes in the retransmitted segment.
In a second scenario, shown in Figure , Host A sends two segments back to back. The first segment has sequence number 92 and 8 bytes of data, and the second segment has sequence number 100 and 20 bytes of data. Suppose that both segments arrive intact at B, and B sends two separate acknowledgments for each of these segments. The first of these acknowledgments has acknowledgment number 100; the second has acknowledgment number 120. Suppose now that neither of the acknowledgments arrives at Host A before the timeout. When the timeout event occurs, Host A resends the first segment with sequence number 92 and restarts the timer. As long as the ACK for the second segment arrives before the new timeout, the second segment will not be retransmitted



In a third and final scenario, suppose Host A sends the two segments, exactly as in the second example. The acknowledgment of the first segment is lost in the network, but just before the timeout event, Host A receives an acknowledgment with acknowledgment number 120. Host A therefore knows that Host B has received everything up through byte 119; so Host A does not resend either of the two segments. This scenario is illustrated in Figure.



MODULE - 3

5. a. Explain datagram format of IPV4? **(10 Marks)**

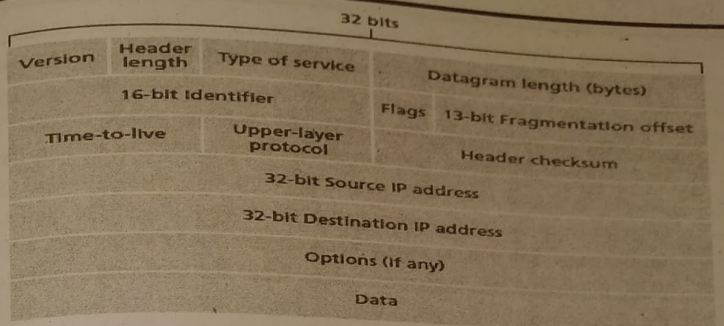
Ans. The IPv4 datagram format is shown in Figure. The key fields in the IPv4 datagram are the following:

- **Version number**: These 4 bits specify the IP protocol version of the datagram. By looking at the version number, the router can determine how to interpret the remainder of the IP datagram. Different versions of IP use different datagram formats. The datagram format for the current version of IP, IPv4, is shown in Figure. The datagram format for the new version of IP (IPv6) is discussed at the end of this section.
- **Header length**: Because an IPv4 datagram can contain a variable number of options these 4 bits are needed to determine where in the IP datagram the data actually begins. Most IP datagrams do not contain options, so the typical IP datagram has a 20-byte header.
- **Type of service**: The type of service (TOS) bits were included in the IPv4 header to allow different types of IP datagrams (for example, datagrams particularly requiring low delay, high throughput, or reliability) to be distinguished from each other. For example, it might be useful to distinguish real-time datagrams (such as those used by an IP telephony application) from non-real-time traffic (for example, FTP). The specific level of service to be provided is a policy issue determined by the router's administrator.
- **Datagram length**: This is the total length of the IP datagram measured in bytes. Since this field is 16 bits long, the theoretical maximum size of the IP datagram is 65,535 bytes. However, datagrams are rarely larger than 1,500 bytes.
- **Identifier, flags, fragmentation offset:** These three fields have to do with so-called IP fragmentation,. Interestingly, the new version of IP, IPv6, does not allow for fragmentation at routers.
- **Time-to-live**: The time-to-live (TTL) field is included to ensure that datagrams do not circulate forever in the network. This field is decremented by one each time the datagram is processed by a router. If the TTL field reaches 0, the datagram must be dropped.

- **Protocol:** This field is used only when an IP datagram reaches its final destination. The value of this field indicates the specific transport-layer protocol to which the data portion of this IP datagram should be passed. For example, a value of 6 indicates that the data portion is passed to TCP, while a value of 17 indicates that the data is passed to UDP. For a list of all possible values.Note that the protocol number in the IP datagram has a role that is analogous to the role of the port number field in the transport layer segment. The protocol number is the glue that binds the network and transport layers together, whereas the port number is the glue that binds the transport and application layers together. The link-layer frame also has a special field that binds the link layer to the network layer.
- **Header checksum**: The header checksum aids a router in detecting bit errors in a received IP datagram. The header checksum is computed by treating each 2 bytes in the header as a number and summing these numbers using 1s complement arithmetic. The 1s complement of this sum, known as the Internet checksum, is stored in the checksum field. A router computes the header checksum for each received IP datagram and detects an error condition if the checksum carried in the datagram header does not equal the computed checksum. Routers typically discard datagrams for which an error has been detected. Note that the checksum must be recomputed and stored again at each router, as the TTL field, and possibly the options field as well, may change.fast algorithms for computing the Internet checksum is [RFC 1071].First, note that only the IP header is check summed at the IP layer, while the TCP/UDP checksum is computed over the entire TCP/UDP segment.

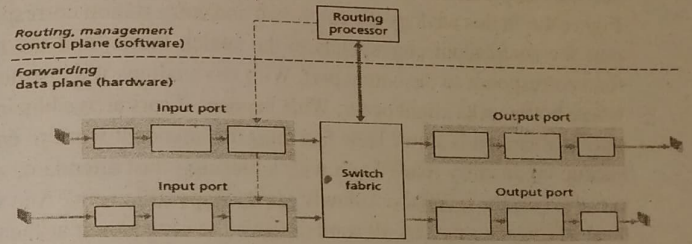
Second, TCP/UDP and IP do not necessarily both have to belong to the same protocol stack. TCP can, in principle, run over a different protocol (for example, ATM) and IP can carry data that will not be passed to TCP/UDP.

- **Source and destination IP addresses:** When a source creates a datagram, it inserts its IP address into the source IP address field and inserts the address of the ultimate destination into the destination IP address field.
- **Options**: The options fields allow an IP header to be extended. Header options were meant to be used rarely hence the decision to save overhead by not including the information in options fields in every datagram header. Since datagram headers can be of variable length, one cannot determine a priori where the data field will start. Also, since some datagrams may require options processing and others may not, the amount of time needed to process an IP datagram at a router can vary greatly. These considerations become particularly important for IP processing in high-performance routers and hosts.
- **Data (payload):** the data field of the IP datagram contains the transport-layer segment to be delivered to the destination. However, the data field can carry other types of data, such as ICMP messages Note that an IP datagram has a total of 20 bytes of header if the datagram carries a TCP segment, then each datagram carries a total of 40 bytes of header along with the application-layer message.



b. Explain generic router architecture with its components? (06 Marks)

Ans.



A high-level view of a generic router architecture is shown in Figure (1). Four router components can be identified:

- **Input ports--**An input port performs several key functions. It performs the physical layer function of terminating an incoming physical link at a router; this is shown in the leftmost box of the input port and the rightmost box of the output port in Figure (1). An input port also performs link-layer functions needed to interoperate with the link layer at the other side of the incoming link; this is represented by the middle boxes in the input and output ports. Perhaps most crucially, the lookup function is also performed at the input port; this will occur in the rightmost box of the input port. It is here that the forwarding table is consulted to determine the router output port to which an arriving packet will be forwarded via the switching fabric. Control packets are forwarded from an input port to the routing processor. Note that the term port here referring to the physical input and output router interfaces is distinctly different from the software ports associated with network applications and sockets .
- **Switching fabric-** The switching fabric connects the router's input ports to its output ports. This switching fabric is completely contained within the router a network inside of a network router!
- **Output ports--** An output port stores packets received from the switching fabric and transmits these packets on the outgoing link by performing the necessary link-layer and physical-layer functions.

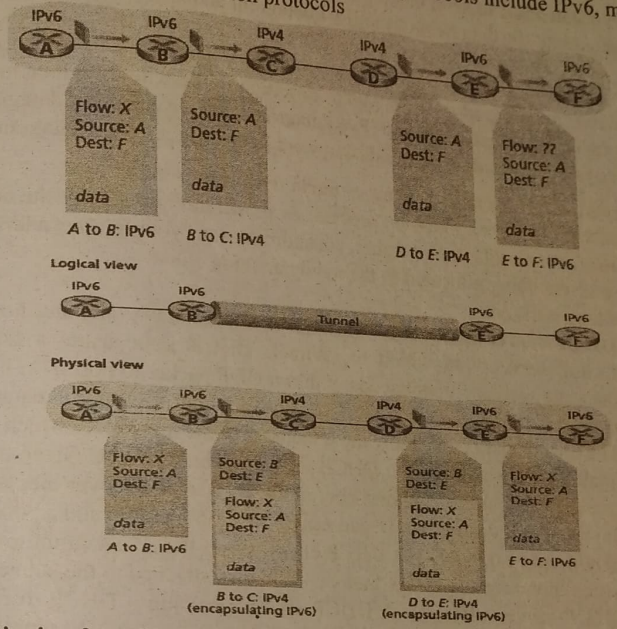
- **Routing processor--** The routing processor executes the routing protocols maintains routing tables and attached link state information, and computes the forwarding table for the router. It also performs the network management functions .A router's input ports, output ports, and switching fabric together implement the forwarding function and are almost always implemented in hardware, as shown in Figure (1). These forwarding functions are sometimes collectively referred to as the router forwarding plane .Forwarding plane hardware can be implemented either using a router vendor's own hardware designs, or constructed using purchased merchant-silicon chips While the forwarding plane operates at the nanosecond time scale, a router's control functions executing the routing protocols, responding to attached links that go up or down, and performing management functions router control plane functions are usually implemented in software and execute on the routing processor the principal router components in Figure 4.6 in this analogy the entry road and entry station correspond to the input port the roundabout corresponds to the switch fabric; and the roundabout exit road corresponds to the output port. With this analogy, it's instructive to consider where bottlenecks might occur. What happens if cars arrive blazingly fast but the station attendant is slow? How fast must the attendant work to ensure there's no backup on an entry road? Even with a blazingly fast attendant, what happens if cars traverse the roundabout slowly can backups still occur? And what happens if most of the entering cars all want to leave the roundabout at the same exit ramp can backups occur at the exit ramp or elsewhere? How should the roundabout operate if we want to assign priorities to different cars, or block certain cars from entering the roundabout in the first place? These are all analogous to critical questions faced by router and switch designers.

OR

6. a. What is Tunneling? Explain? What are the needs to changes from IPV4 to IPV6 (10Marks)

Ans. An alternative to the dual-stack approach, also discussed in RFC 4213, is known as **tunneling**. Tunneling can solve the problem noted above, allowing, for example, E to receive the IPv6 datagram originated by A. The basic idea behind tunneling is the following. Suppose two IPv6 nodes want to interoperate using IPv6 datagrams but are connected to each other by intervening IPv4 routers. We refer to the intervening set of IPv4 routers between two IPv6 routers as a tunnel, as illustrated in Figure (1). With tunneling, the IPv6 node on the sending side of the tunnel takes the entire IPv6 datagram and puts it in the data field of an IPv4 datagram. This IPv4 datagram is then addressed to the IPv6 node on the receiving side of the tunnel and sent to the first node in the tunnel. The intervening IPv4 routers in the tunnel route this IPv4 datagram among themselves, just as they would any other datagram, blissfully unaware that the IPv4 datagram itself contains a complete IPv6 datagram. The IPv6 node on the receiving side of the tunnel eventually receives the IPv4 datagram determines that the IPv4 datagram contains an IPv6 datagram, extracts the IPv6

datagram, and then routes the IPv6 datagram exactly as it would if it had received the IPv6 datagram from a directly connected IPv6 neighbor. IPv6 is enormously difficult to change network-layer protocols. These protocols include IPv6, multicast protocols and resource reservation protocols



Transitioning from IPv4 to IPv6

A flag day involving hundreds of millions of machines and millions of network administrators and users is even more unthinkable today. RFC 4213 describes two approaches for gradually integrating IPv6 hosts and routers into an IPv4 world . The way to introduce IPv6-capable nodes is a dual-stack approach, where IPv6 nodes also have a complete IPv4 implementation. Such a node, referred to as an IPv6/IPv4 node in RFC 4213, has the ability to send and receive both IPv4 and IPv6 datagrams. When interoperating with an IPv4 node, an IPv6/IPv4 node can use IPv4 datagrams; when interoperating with an IPv6 node, it can speak IPv6. IPv6/IPv4 nodes must have both IPv6 and IPv4 addresses. They must be able to determine whether another node is IPv6-capable or IPv4-only. This problem can be solved using the DNS which can return an IPv6 address if the node name being resolved is IPv6-capable, or otherwise return an IPv4 address. Of course, if the node issuing the DNS request is only IPv4-capable, the DNS returns only an IPv4 address. In the dual-stack approach, if either the sender or the receiver is only IPv4- capable, an IPv4 datagram must be used. As a result, it is possible that two IPv6- capable nodes can end up, in essence, sending IPv4 datagrams to each other. This is illustrated in Figure. Suppose Node A is IPv6

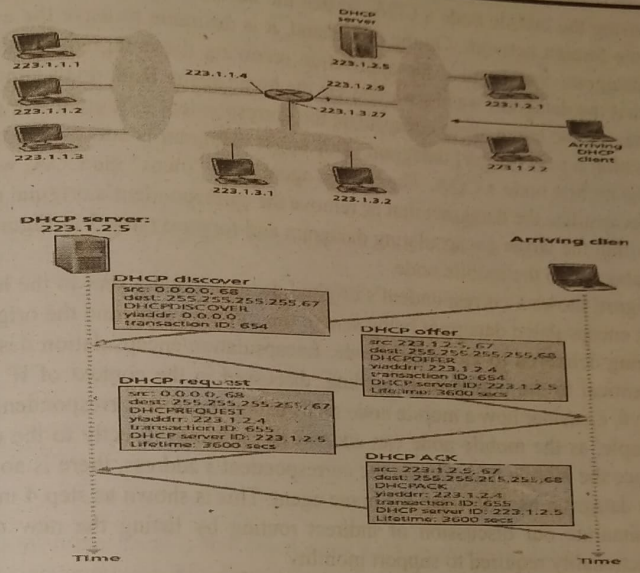
capable and wants to send an IP datagram to Node F, which is also IPv6-capable. Nodes A and B can exchange an IPv6 datagram. However, Node B must create an IPv4 datagram to send to C. Certainly, the data field of the IPv6 datagram can be copied into the data field of the IPv4 datagram and appropriate address mapping can be done. However, in performing the conversion from IPv6 to IPv4, there will be IPv6-specific fields in the IPv6 datagram (for example, the flow identifier field) that have no counterpart in IPv4. The information in these fields will be lost. Thus, even though E and F can exchange IPv6 datagrams, the arriving IPv4 datagrams at E from D do not contain all of the fields that were in the original IPv6 datagram sent from A.

b. Explain DHCP client-Server scenario with interaction? (06 Marks)

Ans. DHCP protocol is a four-step process, as shown in Figure(2) for the network setting shown in Figure(1). In this figure, yia address(as in "your Internet address") indicates the address being allocated to the newly arriving client.

The four steps are:

- **DHCP server discovery**---The first task of a newly arriving host is to find a DHCP server with which to interact. This is done using a DHCP discover message, which a client sends within a UDP packet to port 67. The UDP packet is encapsulated in an IP datagram. But to whom should this datagram be sent? The host doesn't even know the IP address of the network to which it is attaching, much less the address of a DHCP server for this network. Given this, the DHCP client creates an IP datagram containing its DHCP discover message along with the broadcast destination IP address of 255.255.255.255 and a "this host" source IP address of 0.0.0.0. The DHCP client passes the IP datagram to the link layer, which then broadcasts this frame to all nodes attached to the subnet.
- **DHCP server offer(s)**---A DHCP server receiving a DHCP discover message responds to the client with a DHCP offer message that is broadcast to all nodes on the subnet, again using the IP broadcast address of 255.255.255.255. Since several DHCP servers can be present on the subnet, the client may find itself in the enviable position of being able to choose from among several offers. Each server offer message contains the transaction ID of the received discover message, the proposed IP address for the client, the network mask, and an IP address lease time—the amount of time for which the IP address will be valid. It is common for the server to set the lease time to several hours or days.
- **DHCP request**--- The newly arriving client will choose from among one or more server offers and respond to its selected offer with a DHCP request message, echoing back the configuration parameters.
- **DHCP ACK**---The server responds to the DHCP request message with a DHCP ACK message, confirming the requested parameters.



Once the client receives the DHCP ACK, the interaction is complete and the client can use the DHCP-allocated IP address for the lease duration. Since a client may want to use its address beyond the lease's expiration, DHCP also provides a mechanism that allows a client to renew its lease on an IP address. The value of DHCP's plug-and-play capability is clear, considering the fact that the alternative is to manually configure a host's IP address.

MODULE-4

7. a. Explain Indirect Routing to a mobile node? (10 Marks)

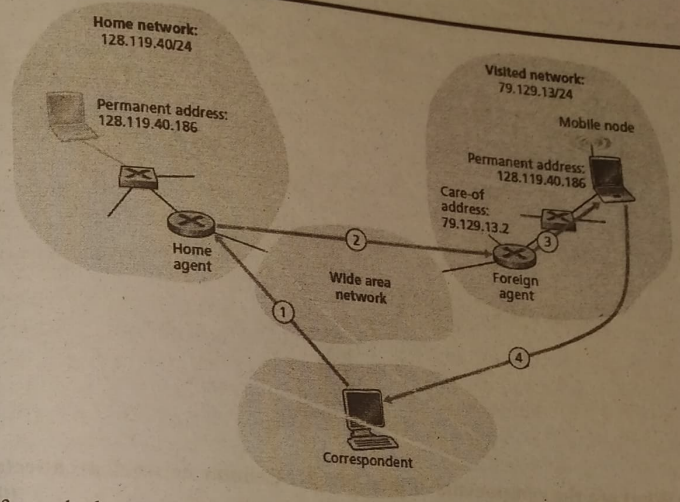
Ans. In the indirect routing approach, the correspondent simply addresses the datagram to the mobile node's permanent address and sends the datagram into the network, blissfully unaware of whether the mobile node is resident in its home network or is visiting a foreign network; mobility is thus completely transparent to the correspondent. Such datagrams are first routed, as usual, to the mobile node's home network. This is illustrated in step 1 in Figure (1). Our attention towards the home agent. In addition to being responsible for interacting with a foreign agent to track the mobile node's COA, the home agent has another very important function. Its second job is to be on the lookout for arriving datagrams addressed to nodes whose home network is that of the home agent but that are currently resident in a foreign network. The home agent intercepts these datagrams and then forwards them to a mobile node in a two-step process. The datagram is first forwarded to the foreign agent, using the mobile node's COA (step 2 in Figure), and then forwarded from the foreign agent to the mobile node (step 3 in Figure). It is instructive to consider this rerouting in more detail. The home agent will need to address the datagram

using the mobile node's COA, so that the network layer will route the datagram to the foreign network. On the other hand, it is desirable to leave the correspondent's datagram intact, since the application receiving the datagram should be unaware that the datagram was forwarded via the home agent. Both goals can be satisfied by having the home agent encapsulate the correspondent's original complete datagram within a new (larger) datagram. This larger datagram is addressed and delivered to the mobile node's COA. The foreign agent, who "owns" the COA, will receive and decapsulate the datagram that is, remove the correspondent's original datagram from within the larger encapsulating datagram and forward (step 3 in Figure) the original datagram to the mobile node.

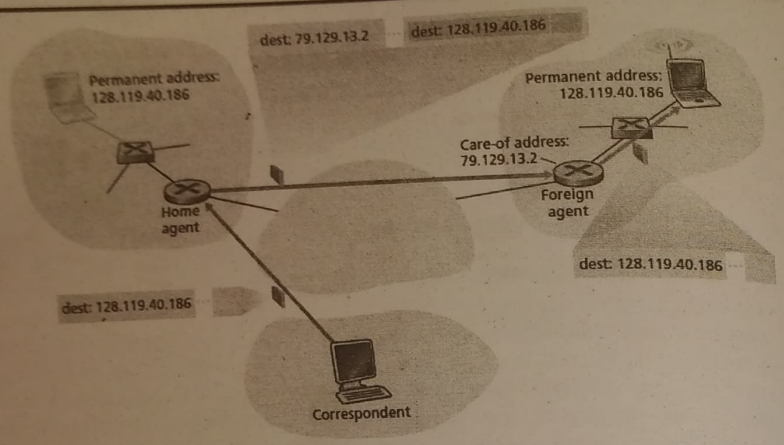
Figure shows a correspondent's original datagram being sent to the home network, an encapsulated datagram being sent to the foreign agent, and the original datagram being delivered to the mobile node. Encapsulation/encapsulation described here is identical to the notion of tunnelling, discussed in the context of IP multicast and IPv6. Consider how a mobile node sends datagrams to a correspondent. This is quite simple, as the mobile node can address its datagram directly to the correspondent Since the mobile node knows the correspondent's address, there is no need to route the datagram back through the home agent. This is shown as step 4 in Figure. Let's summarize our discussion of indirect routing by listing the new network layer functionality required to support mobility

- A mobile-node–to–foreign-agent protocol. The mobile node will register with the foreign agent when attaching to the foreign network. Similarly, a mobile node will deregister with the foreign agent when it leaves the foreign network.
- A foreign-agent–to–home-agent registration protocol. The foreign agent will register the mobile node's COA with the home agent. A foreign agent need not explicitly deregister a COA when a mobile node leaves its network, because the subsequent registration of a new COA, when the mobile node moves to a new network, will take care of this.
- A home-agent datagram encapsulation protocol. Encapsulation and forwarding of the correspondent's original datagram within a datagram addressed to the COA.
- A foreign-agent decapsulation protocol. Extraction of the correspondent's original datagram from the encapsulating datagram, and the forwarding of the original datagram to the mobile node.

The previous discussion provides all the pieces foreign agents, the home agent, and indirect forwarding needed for a mobile node to maintain an ongoing connection while moving among networks. As an example of how these pieces fit together, assume the mobile node is attached to foreign network A, has registered a COA in network A with its home agent, and is receiving datagrams that are being indirectly routed through its home agent. The mobile node now moves to foreign network B and registers with the foreign agent in network B, which informs the home agent of the mobile node's new COA. From this point on, the home agent will reroute datagrams to foreign network B. As far as a correspondent is concerned, mobility is transparent—datagrams are routed via the same home agent both before and after the move.

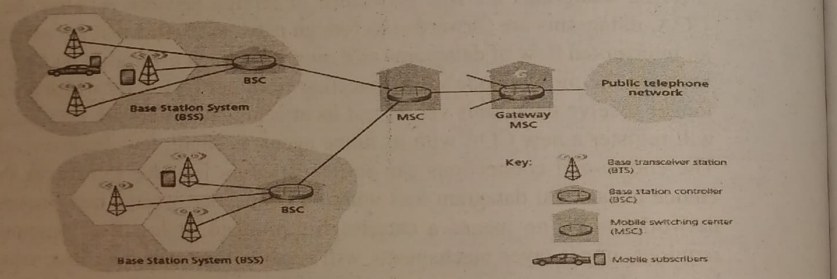


As far as the home agent is concerned, there is no disruption in the flow of datagrams arriving datagrams are first forwarded to foreign network A; after the change in COA, datagrams are forwarded to foreign network B. But will the mobile node see an interrupted flow of datagrams as it moves between networks? As long as the time between the mobile node's disconnection from network A (at which point it can no longer receive datagrams via A) and its attachment to network B (at which point it will register a new COA with its home agent) is small, few datagrams will be lost. The end-to-end connections can suffer datagram loss due to network congestion. Hence occasional datagram loss within a connection when a node moves between networks is by no means a catastrophic problem. If loss-free communication is required, upper-layer mechanisms will recover from datagram loss, whether such loss results from network congestion or from user mobility. An indirect routing approach is used in the mobile IP standard.



b. What are the components of the GSM 2G Cellular network architecture? **(06 Marks)**

Ans.



Cellulr Network Architecture, 2G: Voice Connections to the Telephone Network

The term cellular refers to the fact that the region covered by a cellular network is partitioned into a number of geographic coverage areas, known as cells, shown as hexagons on the left side of Figure (1). GSM has its own particular nomenclature. Each cell contains a base transceiver station (BTS) that transmits signals to and receives signals from the mobile stations in its cell. The coverage area of a cell depends on many factors, including the transmitting power of the BTS, the transmitting power of the user devices, obstructing buildings in the cell, and the height of base station antennas. Figure (1) shows each cell containing one base transceiver station residing in the middle of the cell, many systems today place the BTS at corners where three cells intersect, so that a single BTS with directional antennas can service three cells. The GSM standard for 2G cellular systems uses combined FDM/TDM (radio) for the air interface. Pure FDM, the channel is partitioned into a number of frequency bands with each band devoted to a call. Pure TDM, time is partitioned into frames

with each frame further partitioned into slots and each call being assigned the use of a particular slot in the revolving frame. In combined FDM/TDM systems, the channel is partitioned into a number of frequency sub-bands; within each sub-band, time is partitioned into frames and slots. Thus, for a combined FDM/TDM system, if the channel is partitioned into F sub-bands and time is partitioned into T slots, then the channel will be able to support F.T simultaneous calls. we saw in that cable access networks also use a combined FDM/TDM approach. GSM systems consist of 200-kHz frequency bands with each band supporting eight TDM calls. GSM encodes speech at 13 kbps and 12.2 kbps.

A GSM network's base station controller (BSC) will typically service several tens of base transceiver stations. The role of the BSC is to allocate BTS radio channels to mobile subscribers, perform paging (finding the cell in which a mobile user is resident), and perform handoff of mobile users. The base station controller and its controlled base transceiver stations collectively constitute a GSM base station system (BSS). The mobile switching center (MSC) plays the central role in user authorization and accounting (e.g., determining whether a mobile device is allowed to connect to the cellular network), call establishment and teardown, and handoff. A single MSC will typically contain up to five BSCs, resulting in approximately 200K subscribers per MSC. A cellular provider's network will have a number of MSCs, with special MSCs known as gateway MSCs connecting the provider's cellular network to the larger public telephone network.

OR

8. a. What are the 3 approaches of TCP over wireless links? **(06 Marks)**

Ans. Three broad classes of approaches are possible for dealing with this problem:

- **Local recovery**: Local recovery protocols recover from bit errors when and where (e.g., at the wireless link) they occur, e.g., the 802.11 ARQ protocol we studied
- **TCP sender awareness of wireless links**. In the local recovery approaches, the TCP sender is blissfully unaware that its segments are traversing a wireless link. An alternative approach is for the TCP sender and receiver to be aware of the existence of a wireless link, to distinguish between congestive losses occurring in the wired network and corruption/loss occurring at the wireless link, and to invoke congestion control only in response to congestive wired-network losses. [Balakrishnan 1997] investigates various types of TCP, assuming that end systems can make this distinction. [Liu 2003] investigates techniques for distinguishing between losses on the wired and wireless segments of an end-to-end path.
- **Split-connection approaches** - In a split-connection approach [Bakre 1995], the end-to end connection between the mobile user and the other end point is broken into two transport layer connections: one from the mobile host to the wireless access point, and one from the wireless access point to the other communication end point (which we'll assume here is a wired host). The end-to-end connection is thus formed by the concatenation of a wireless part and a wired part. The transport layer over the wireless segment can be a standard TCP connection [Bakre 1995], or a specially tailored error recovery protocol on top of UDP. investigates the

use of a transport-layer selective repeat protocol over the wireless connection. Measurements reported in [Wei 2006] indicate that split TCP connections are widely used in cellular data networks, and that significant improvements can indeed be made through the use of split TCP connections.

b. Write a note on UMTS (Universal Mobile Telecommunications service)? (04 Marks)

Ans. One of themost popular 3 G technology. The 3G radio access network is the wireless first-hop network that we see as a 3G user. The Radio Network Controller (RNC) typically controls several cell base transceiver stations similar to the base stations that we encountered in 2G systems. Each cell's wireless link operates between the mobile nodes and a base transceiver station, just as in 2G networks. The RNC connects to both the circuit-switched cellular voice network via an MSC, and to the packet-switched Internet via an SGSN. Thus, while 3G cellular voice and cellular data services use different core networks, they share a common first/last-hop radio access network. A significant change in 3G UMTS over 2G networks is that rather than using GSM's FDMA/TDMA scheme, UMTS uses a CDMA technique known as Direct Sequence Wideband CDMA (DS-WCDMA) [Dahlman 1998] within TDMA slots.

c. Define a) Home Network b) Home Agent c)Foreign Network?(06 Marks)

Ans. **Home Network -**In a network setting, the permanent home of a mobile node (such as a laptop or smartphone) is known as the home network

b) Home Agent - the entity within the home network that performs the mobility management functions on behalf of the mobile node is known as the home agent.

c) Foreign Network - The network in which the mobile node is currently residing is known as the foreign (or visited) network,

MODULE – 5

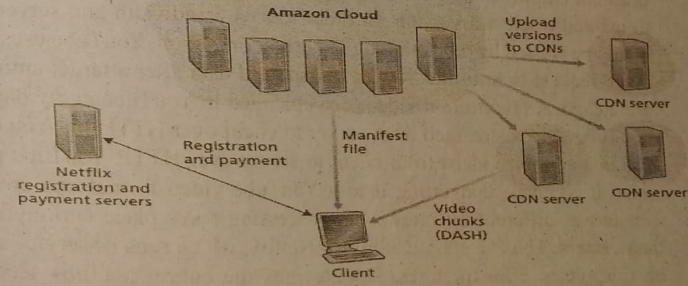
9. a. Explain Netflix and YouTube? (10 Marks)

Ans. Generating almost 30 percent of the downstream U.S. Internet traffic in 2011, Netflix has become the leading service provider for online movies and TV shows in the United States [Sandvine 2011]. In order to rapidly deploy its large-scale service, Netflix has made extensive use of third-party cloud services and CDNs. Indeed, Netflix is an interesting example of a company deploying a large-scale online service by renting servers, bandwidth, storage, and database services from third parties while using hardly any infrastructure of its own. The following discussion is adapted from a very readable measurement study of the Netflix architecture [Adhikari 2012]. Netflix employs many of the techniques covered which includes video distribution using a CDN and adaptive streaming over HTTP. Figure (2) shows the basic architecture of the Netflix video streaming platform. It has four major components: the registration and payment servers, the Amazon cloud, multiple CDN providers, and clients. In its own hardware infrastructure, Netflix maintains registration and payment servers, which handle registration of newaccounts and capture credit-card payment information. Except for these basic functions, Netflix runs its online service by employing machines (or virtual machines) in the Amazon cloud. Some of the

functions taking place in the Amazon cloud include:

- Content ingestion. Before Netflix can distribute a movie to its customers, it must first ingest and process the movie. Netflix receives studio master versions of movies and uploads them to hosts in the Amazon cloud.
- Content processing. The machines in the Amazon cloud create many different formats for each movie, suitable for a diverse array of client video players running on desktop computers, smartphones, and game consoles connected to televisions.
- A different version is created for each of these formats and at multiple bit rates, allowing for adaptive streaming over HTTP using DASH.
- Uploading versions to the CDNs. Once all of the versions of a movie have been created, the hosts in the Amazon cloud upload the versions to the CDNs.

To deliver the movies to its customers on demand, Netflix makes extensive use of CDN technology. In fact, as of this writing in 2012, Netflix employs not one but three third-party CDN companies at the same time Akamai, Limelight, and Level-3. Having described the components of the Netflix architecture, the interaction between the client and the various servers that are involved in movie delivery. The Web pages for browsing the Netflix video library are served from servers in the Amazon cloud. When the user selects a movie to "Play Now," the user's client obtains a manifest file, also from servers in the Amazon cloud. The manifest file includes a variety of information, including a ranked list of CDNs and the URLs for the different versions of the movie, which are used for DASH playback. The ranking of the CDNs is determined by Netflix, and may change from one streaming session to the next. Typically the client will select the CDN that is ranked highest in the manifest file. After the client selects a CDN, the CDN leverages DNS to redirect the client to a specific CDN server.



The client and that CDN server then interact using DASH. the client uses the byte-range header in HTTP GET request messages, to request chunks from the different versions of the movie. Netflix uses chunks that are approximately four-seconds long [Adhikari 2012]. While the chunks are being downloaded, the client measures the received throughput and runs a rate-determination algorithm to determine the quality of the next chunk to request. Netflix embodies many of the key principles discussed earlier in this section, including adaptive streaming and CDN distribution. Netflix also nicely illustrates how a major Internet service, generating almost 30 percent of

Internet traffic, can run almost entirely on a third-party cloud and third-party CDN infrastructures, using very little infrastructure of its own!

YouTube

With approximately half a billion videos in its library and half a billion video views per day [Ding 2011], YouTube is indisputably the world's largest video-sharing site. YouTube began its service in April 2005 and was acquired by Google in November 2006. Although the Google/YouTube design and protocols are proprietary, through several independent measurement efforts we can gain a basic understanding about how YouTube operates [Zink 2009; Torres 2011; Adhikari 2011a]. As with Netflix, YouTube makes extensive use of CDN technology to distribute its videos [Torres 2011]. Unlike Netflix, however, Google does not employ third-party CDNs but instead uses its own private CDN to distribute YouTube videos. Google has installed server clusters in many hundreds of different locations. From a subset of about 50 of these locations, Google distributes YouTube videos [Adhikari 2011a]. Google uses DNS to redirect a customer request to a specific cluster, as described in Section 7.2.4. Most of the time, Google's cluster selection strategy directs the client to the cluster for which the RTT between client and cluster is the lowest; however, in order to balance the load across clusters, sometimes the client is directed (via DNS) to a more distant cluster [Torres 2011]. if a cluster does not have the requested video, instead of fetching it from somewhere else and relaying it to the client, the cluster may return an HTTP redirect message, thereby redirecting the client to another cluster [Torres 2011]. YouTube employs HTTP streaming,

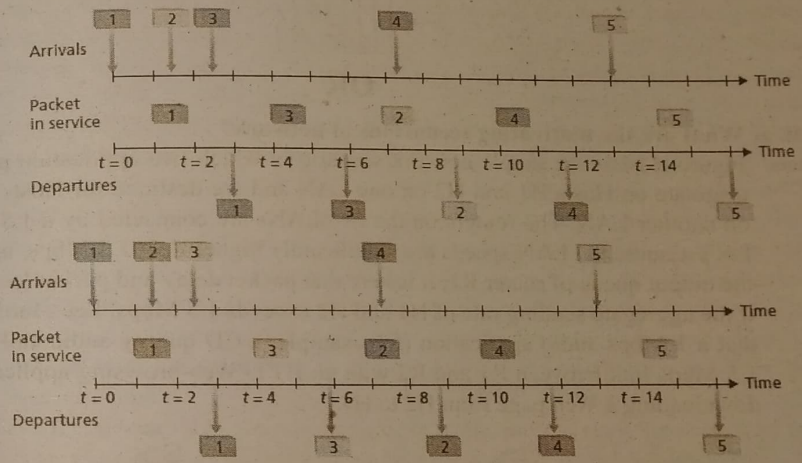
YouTube often makes a small number of different versions available for a video, each with a different bit rate and corresponding quality level. As of 2011, YouTube does not employ adaptive streaming (such as DASH), but instead requires the user to manually select a version. In order to save bandwidth and server resources that would be wasted by repositioning or early termination, YouTube uses the HTTP byte range request to limit the flow of transmitted data after a target amount of video is prefetched. A few million videos are uploaded to YouTube every day. Not only are YouTube videos streamed from server to client over HTTP, but YouTube uploaders also upload their videos from client to server over HTTP. YouTube processes each video it receives, converting it to a YouTube video format and creating multiple versions at different bit rates. This processing takes place entirely within Google data centers. Thus, in stark contrast to Netflix, which runs its service almost entirely on third-party infrastructures, Google runs the entire YouTube service within its own vast infrastructure of data centers, private CDN, and private global network interconnecting its data centers and CDN clusters.

b. Explain Round robin and Weighted fair queueing (WFQ)? (06 Marks)

Ans. Under the round robin queuing discipline, packets are sorted into classes as with priority queuing. However, rather than there being a strict priority of service among classes, a round robin scheduler alternates service among the classes. In the simplest form of round robin scheduling, a class 1 packet is transmitted, followed by a class

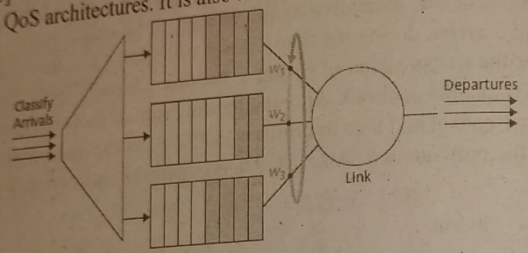
2 packet, followed by a class 1 packet, followed by a class 2 packet, and so on. A so-called work-conserving queuing discipline will never allow the link to remain idle whenever there are packets (of any class) queued for transmission. A work-conserving round robin discipline that looks for a packet of a given class but finds none will immediately check the next class in the round robin sequence.

Figure (2) illustrates the operation of a two-class round robin queue. In this example, packets 1, 2, and 4 belong to class 1, and packets 3 and 5 belong to the second class. Packet 1 begins transmission immediately upon arrival at the output queue. Packets 2 and 3 arrive during the transmission of packet 1 and thus queue for transmission. After the transmission of packet 1, the link scheduler looks for a class 2 packet and thus transmits packet 3. After the transmission of packet 3, the scheduler looks for a class 1 packet and thus transmits packet 2. After the transmission of packet 2, packet 4 is the only queued packet; it is thus transmitted immediately after packet 2.



A generalized abstraction of round robin queuing that has found considerable use in QoS architectures is the so-called weighted fair queuing (WFQ) discipline [Demers 1990; Parekh 1993]. WFQ is illustrated in Figure . Arriving packets are classified and queued in the appropriate per-class waiting area. As in round robin scheduling, a WFQ scheduler will serve classes in a circular manner first serving class 1, then serving class 2, then serving class 3, and then (assuming there are three classes) repeating the service pattern. WFQ is also a work conserving queuing discipline and thus will immediately move on to the next class in the service sequence when it finds an empty class queue. WFQ differs from round robin in that each class may receive a differential amount of service in any interval of time. Specifically, each class, i, is assigned a weight, wi. Under WFQ, during any interval of time during which there are class I packets to send, class i will then be guaranteed to receive a fraction of service equal to wi/(Σwj), where the sum in the denominator is taken over all classes that also have packets queued for transmission. In the worst case, even if all classes

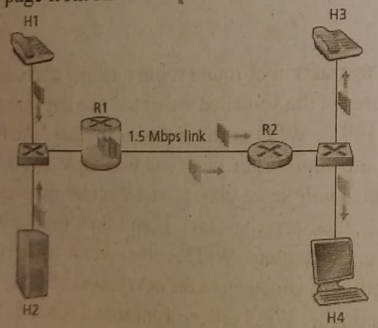
have queued packets, class i will still be guaranteed to receive a fraction wi /(Σwj) of the bandwidth. Thus, for a link with transmission rate R, class i will always achieve a throughput of at least R · wi /(Σwj). Our description of WFQ has been an idealized one, as we have not considered the fact that packets are discrete units of data and a packet's transmission will not be interrupted to begin transmission of another packet; [Demers 1990] and [Parekh 1993] discuss this packetization issue. WFQ plays a central role in QoS architectures. It is also available in today's router products.



OR

10. a. What are the motivating scenarious of network? **(10 Marks)**

Ans. Figure (1)shows a simple network scenario in which two application packet flows originate on Hosts H1 and H2 on one LAN and are destined for Hosts H3 and H4 on another LAN. The routers on the two LANs are connected by a 1.5 Mbps link. Let's assume the LAN speeds are significantly higher than 1.5 Mbps, and focus on the output queue of router R1; it is here that packet delay and packet loss will occur if the aggregate sending rate of H1 and H2 exceeds 1.5 Mbps. Let's further suppose that a 1 Mbps audio application (for example, a CD quality audio call) shares the 1.5 Mbps link between R1 and R2 with an HTTPWeb-browsing application that is downloading a Web page from H2 to H4.



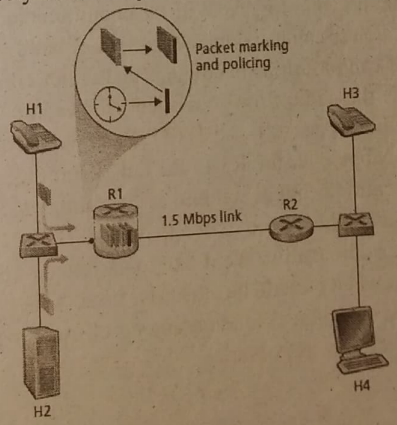
In the best-effort Internet, the audio and HTTP packets are mixed in the output queue at R1 and (typically) transmitted in a first-in-first-out (FIFO) order. In this scenario, a burst of packets from the Web server could potentially fill up the queue, causing IP audio packets to be excessively delayed or lost due to buffer overflow at R1. How should we solve this potential problem? Given that the HTTP Web-browsing application does not have time constraints, our intuition might be to give strict priority to audio packets

at R1. Under a strict priority scheduling discipline, an audio packet in the R1 output buffer would always be transmitted before any HTTP packet in the R1 output buffer. The link from R1 to R2 would look like a dedicated link of 1.5 Mbps to the audio traffic, with HTTP traffic using the R1-to-R2 link only when no audio traffic is queued. In order for R1 to distinguish between the audio and HTTP packets in its queue, each packet must be marked as belonging to one of these two classes of traffic. This was the original goal of the type-of-service (ToS) field in IPv4. It is our first insight into mechanisms needed to provide multiple classes of traffic.

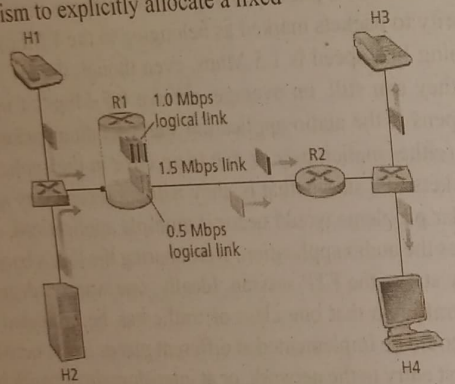
Insight 1: Packet marking allows a router to distinguish among packets belonging to different classes of traffic.

Note that although our example considers a competing multimedia and elastic flow, the same insight applies to the case that platinum, gold, and silver classes of service are implemented a packet-marking mechanism is still needed to indicate that class of service to which a packet belongs. Now suppose that the router is configured to give priority to packets marked as belonging to the 1 Mbps audio application. Since the outgoing link speed is 1.5 Mbps, even though the HTTP packets receive lower priority, they can still, on average, receive 0.5 Mbps of transmission service. But what happens if the audio application starts sending packets at a rate of 1.5 Mbps or higher (either maliciously or due to an error in the application)? In this case, the HTTP packets will starve, that is, they will not receive any service on the R1-to-R2 link. Similar problems would occur if multiple applications, all with the same class of service as the audio application, were sharing the link's bandwidth; they too could collectively starve the FTP session. Ideally, one wants a degree of isolation among classes of traffic so that one class of traffic can be protected from the other. This protection could be implemented at different places in the network at each and every router, at first entry to the network, or at inter-domain network boundaries. This then is our second insight:

Insight 2: It is desirable to provide a degree of traffic isolation among classes so that one class is not adversely affected by another class of traffic that misbehaves



We'll examine several specific mechanisms for providing such isolation among traffic classes. We note here that two broad approaches can be taken. First, it is possible to perform traffic policing, as shown in Figure. If a traffic class or flow must meet certain criteria (for example, that the audio flow not exceed a peak rate of 1 Mbps), then a policing mechanism can be put into place to ensure that these criteria are indeed observed. If the policed application misbehaves, the policing mechanism will take some action (for example, drop or delay packets that are in violation of the criteria) so that the traffic actually entering the network conforms to the criteria. The leaky bucket mechanism that we'll examine shortly is perhaps the most widely used policing mechanism. In Figure, the packet classification and marking mechanism (Insight 1) and the policing mechanism (Insight 2) are both implemented together at the network's edge, either in the end system or at an edge router. A complementary approach for providing isolation among traffic classes is for the link-level packet-scheduling mechanism to explicitly allocate a fixed



amount of link bandwidth to each class. For example, the audio class could be allocated 1 Mbps at R1, and the HTTP class could be allocated 0.5 Mbps. In this case, the audio and HTTP flows see a logical link with capacity 1.0 and 0.5 Mbps, respectively, as shown in Figure. With strict enforcement of the link level allocation of bandwidth, a class can use only the amount of bandwidth that has been allocated; in particular, it cannot utilize bandwidth that is not currently being used by others. For example, if the audio flow goes silent (for example, if the speaker pauses and generates no audio packets), the HTTP flow would still not be able to transmit more than 0.5 Mbps over the R1-to-R2 link, even though the audio flow's 1 Mbps bandwidth allocation is not being used at that moment. Since bandwidth is a "use-it-or-lose-it" resource, there is no reason to prevent HTTP traffic from using bandwidth not used by the audio traffic. We'd like to use bandwidth as efficiently as possible, never wasting it when it could be otherwise used. This gives rise to our third insight: Insight 3: While providing isolation among classes or flows, it is desirable to use resources as efficiently as possible.

b. Explain Streaming stored audio/video? **(06 Marks)**

Ans. We focus here on streaming stored video, which typically combines video and audio components. Streaming stored audio (such as streaming music) is very similar to streaming stored video, although the bit rates are typically much lower. In this class of applications, the underlying medium is prerecorded video, such as a movie, a television show, a prerecorded sporting event, or a prerecorded usergenerated video (such as those commonly seen on YouTube). These prerecorded videos are placed on servers, and users send requests to the servers to view the videos on demand. Many Internet companies today provide streaming video, including YouTube (Google), Netflix, and Hulu. By some estimates, streaming stored video makes up over 50 percent of the downstream traffic in the Internet access networks today. Streaming stored video has three key distinguishing features.

- Streaming - In a streaming stored video application, the client typically begins video playout within a few seconds after it begins receiving the video from the server. This means that the client will be playing out from one location in the video while at the same time receiving later parts of the video from the server. This technique, known as streaming, avoids having to download the entire video file before playout begins.
- Interactivity - the media is prerecorded, the user may pause, reposition forward, reposition backward, fast-forward, and so on through the video content. The time from when the user makes such a request until the action manifests itself at the client should be less than a few seconds for acceptable responsiveness.
- Continuous playout - Once playout of the video begins, it should proceed according to the original timing of the recording. Therefore, data must be received from the server in time for its playout at the client; otherwise, users experience video frame freezing or frame skipping By far, the most important performance measure for streaming video is average throughput. In order to provide continuous playout, the network must provide an average throughput to the streaming application that is at least as large the bit rate of the video itself. by using buffering and prefetching, it is possible to provide continuous playout even when the throughput fluctuates, as long as the average throughput (averaged over 5–10 seconds) remains above the video rate [Wang 2008]. For many streaming video applications, prerecorded video is stored on, and streamed from, a CDN rather than from a single data center. There are also many P2P video streaming applications for which the video is stored on users' hosts (peers),with different chunks of video arriving from different peers that may spread around the globe.

Fifth Semester B.E. Degree Examination

CBCS - Model Question Paper - 3

COMPUTER NETWORKS

Time: 3 hrs. Max. Marks: 80

Note : *Answer any FIVE full questions, selecting ONE full question from each module.*

MODULE - 1

1. a. What is a Socket? Explain socket communication between two processes that communicate over the internet? **(06 Marks)**

Ans. A process sends messages into, and receives messages from, the network through a software interface called **a socket**. Consider an analogy to help us understand processes and sockets. A process is analogous to a house and its socket is analogous to its door. When a process wants to send a message to another process on another host, it shoves the message out its door (socket). This sending process assumes that there is a transportation infrastructure on the other side of its door that will transport the message to the door of the destination process. Once the message arrives at the destination host, the message passes through the receiving process's door (socket), and the receiving process then acts on the message

Figure (1) illustrates socket communication between two processes that communicate over the Internet. As shown in this figure, a socket is the interface between the application layer and the transport layer within a host. It is also referred to as the Application Programming Interface (API) between the application and the network, since the socket is the programming interface with which network applications are built. The application developer has control of everything on the application-layer side of the socket but has little control of the transport layer side of the socket. The only control that the application developer has on the transport layer side is (1) the choice of transport protocol and (2) perhaps the ability to fix a few transport-layer parameters such as maximum buffer and maximum segment sizes Once the application developer chooses a transport protocol the application is built using the transport layer services provided by that protocol.

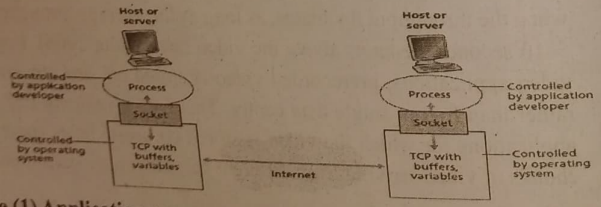


Figure (1) Application processes, sockets, and underlying transport protocol

b. Explain Transport services provided by the internet? **(06 Marks)**

Ans. Transport services provided by the Internet are The Internet makes two transport protocols available to applications, UDP and TCP. When you create a new network application for the Internet, one of the first decisions you have to make is whether to use UDP or TCP. Each of these protocols offers a different set of services to the invoking applications. Figure (2) shows the service requirements for some selected applications.

Application	Data Loss	Throughput	Time-Sensitive
File transfer/download	No loss	Elastic	No
E-mail	No loss	Elastic	No
Web documents	No loss	Elastic (few kbps)	No
Internet telephony/ Video conferencing	Loss-tolerant	Audio: few kbps–1Mbps Video: 10 kbps–5 Mbps	Yes: 100s of msec
Streaming stored audio/video	Loss-tolerant	Same as above	Yes: few seconds
Interactive games	Loss-tolerant	Few kbps–10 kbps	Yes: 100s of msec
Instant messaging	No loss	Elastic	Yes and no

TCP Services

The TCP service model includes a connection-oriented service and a reliable data transfer service. When an application invokes TCP as its transport protocol, the application receives both of these services from TCP.

- **Connection-oriented service**--TCP has the client and server exchange transport layer control information with each other before the application-level messages begin to flow. This so-called handshaking procedure alerts the client and server, allowing them to prepare for an onslaught of packets. After the handshaking phase, a TCP connection is said to exist between the sockets of the two processes. The connection is a full-duplex connection in that the two processes can send messages to each other over the connection at the same time. When the application finishes sending messages, it must tear down the connection
- **Reliable data transfer service**-- The communicating processes can rely on TCP to deliver all data sent without error and in the proper order. When one side of the application passes a stream of bytes into a socket, it can count on TCP to deliver the same stream of bytes to the receiving socket, with no missing or duplicate bytes .TCP also includes a congestion-control mechanism, a service for the general welfare of the Internet rather than for the direct benefit of the communicating processes. The TCP congestion-control mechanism throttles a sending process when the network is congested between sender and receiver

UDP Services

UDP is a no-frills, lightweight transport protocol, providing minimal services. UDP is connectionless, so there is no handshaking before the two processes start to communicate. UDP provides an unreliable data transfer service—that is, when a process sends a message into a UDP socket, UDP provides no guarantee that the message will ever reach the receiving process. Messages that do arrive at the receiving process may arrive out of order. UDP does not include a congestion-control mechanism, so the sending side of UDP can pump data into the layer below at any rate it pleases.

c. Define Application –layer protocol? **(04 Marks)**

Ans. An application-layer protocol defines how an application's processes, running on different end systems, pass messages to each other. an application-layer protocol defines:

- The types of messages exchanged, for example, request messages and response messages
- The syntax of the various message types, such as the fields in the message and how the fields are delineated
- The semantics of the fields, that is, the meaning of the information in the fields
- Rules for determining when and how a process sends messages and responds to messages

OR

2. a. Write short notes on POP3 and IMAP? (10 Marks)

Ans. POP3

POP3 is an extremely simple mail access protocol. It is defined in [RFC 1939], which is short and quite readable. The protocol is so simple, its functionality is rather limited. POP3 begins when the user agent opens a TCP connection to the mail server on port 110. With the TCP connection established, POP3 progresses through three phases: authorization, transaction, and update.

(1) Authorization, the user agent sends a username and a password to authenticate the user. (2) transaction, the user agent retrieves messages; also during this phase, the user agent can mark messages for deletion, remove deletion marks, and obtain mail statistics. (3) occurs after the client has issued the quit command, ending the POP3 session; at this time, the mail server deletes the messages that were marked for deletion.

In a POP3 transaction, the user agent issues commands, and the server responds to each command with a reply. There are two possible responses: +OK (sometimes followed by server-to-client data), used by the server to indicate that the previous command was fine; and -ERR, used by the server to indicate that something was wrong with the previous command. The authorization phase has two principal commands: user <username> and pass <password>. To illustrate these two commands, we suggest that you Telnet directly into a POP3 server, using port 110, and issue these commands. Suppose that mail Server is the name of your mail server. You will see something like: telnet mailServer 110 +OK POP3 server ready user bob +OK pass hungry +OK user successfully logged on If you misspell a command, the POP3 server will reply with an -ERR message. The transaction phase, A user agent using POP3 can often be configured to "download and delete" or to "download and keep." The sequence of commands issued by a POP3 user agent depends on which of these two modes the user agent is operating in. In the download-and-delete mode, the user agent will issue the list, retr, and dele commands. As an example, suppose the user has two messages in his or her mailbox. In the dialogue below, C: is the user agent and S: is the mail server. The transaction will look something like:

C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: (blah blah ...
S: .....
S: .....blah)

S: .
C: dele 1
C: retr 2
S: (blah blah ...
S: .....
S: .....blah)
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off

The user agent first asks the mail server to list the size of each of the stored messages. The user agent then retrieves and deletes each message from the server. Note that after the authorization phase, the user agent employed only four commands: list, retr, dele, and quit. The syntax for these commands is defined in RFC 1939. After processing the quit command, the POP3 server enters the update phase and removes messages 1 and 2 from the mailbox. A problem with this download-and-delete mode is that the recipient, Bob, may be nomadic and may want to access his mail messages from multiple machines, for example, his office PC, his home PC, and his portable computer. The download and-delete mode partitions Bob's mail messages over these three machines; in particular, if Bob first reads a message on his office PC, he will not be able to reread the message from his portable at home later in the evening. In the download-and keep mode, the user agent leaves the messages on the mail server after downloading them. In this case, Bob can reread messages from different machines; he can access a message from work and access it again later in the week from home. During a POP3 session between a user agent and the mail server, the POP3 server maintains some state information; in particular, it keeps track of which user messages have been marked deleted. However, the POP3 server does not carry state information across POP3 sessions. This lack of state information across sessions greatly simplifies the implementation of a POP3 server.

IMAP

With POP3 access, once Bob has downloaded his messages to the local machine, he can create mail folders and move the downloaded messages into the folders. Bob can then delete messages, move messages across folders, and search for messages. But this paradigm namely, folders and messages in the local machine—poses a problem for the nomadic user, who would prefer to maintain a folder hierarchy on a remote server that can be accessed from any computer. This is not possible with POP3—the POP3 protocol does not provide any means for a user to create remote folders and assign messages to folders.

To solve this and other problems, the IMAP protocol, defined in [RFC 3501], was invented. Like POP3, IMAP is a mail access protocol. It has many more features than POP3, but it is also significantly more complex. An IMAP server will associate each message with a folder; when a message first arrives at the server, it is associated with the recipient's INBOX folder. The recipient can then move the message into a new, user-created folder, read the message, delete the message, and so on. The IMAP protocol provides commands to allow users to create folders and move messages from one folder to another. IMAP also provides commands that allow users to search remote folders for messages matching specific criteria. Note that, unlike POP3, an IMAP server maintains

user state information across IMAP sessions—for example, the names of the folders and which messages are associated with which folders. Another important feature of IMAP is that it has commands that permit a user agent to obtain components of messages. For example, a user agent can obtain just the message header of a message or just one part of a multipart MIME message. This feature is useful when there is a low-bandwidth connection between the user agent and its mail server. With a low bandwidth connection, the user may not want to download all of the messages in its mailbox, particularly avoiding long messages that might contain, for example, an audio or video clip.

b. Explain FTP Commands and replies? **(06 Marks)**

Ans. The commands, from client to server, and replies, from server to client, are sent across the control connection in 7-bit ASCII format. FTP commands are readable by people. A carriage return and line feed end each command. Each command consists of four uppercase ASCII characters, some with optional arguments. Some of the more common commands are given below:

- **USER username:** Used to send the user identification to the server.
- **PASS password**: Used to send the user password to the server.
- **LIST:** Used to ask the server to send back a list of all the files in the current remote directory.

The list of files is sent over a data connection rather than the control TCP connection.

- **RETR filename:** Used to retrieve (that is, get) a file from the current directory of the remote host. This command causes the remote host to initiate a data connection and to send the requested file over the data connection.
- **STOR filename:** Used to store a file into the current directory of the remote host.

There is typically a one-to-one correspondence between the command that the user issues and the FTP command sent across the control connection. Each command is followed by a reply, sent from server to client. The replies are three-digit numbers, with an optional message following the number. This is similar in structure to the status code and phrase in the status line of the HTTP response message.

Some typical replies, along with their possible messages, are as follows:

- 331 Username OK, password required
- 125 Data connection already open; transfer starting
- 425 Can't open data connection
- 452 Error writing file

MODULE - 2

3. a. Explain Go-Back-N? **(10 Marks)**

Ans. The sender is allowed to transmit multiple packets without waiting for an acknowledgment, but is constrained to have no more than some maximum allowable number, N, of unacknowledged packets in the pipeline.

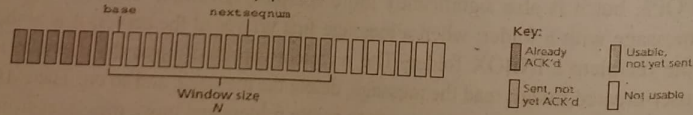


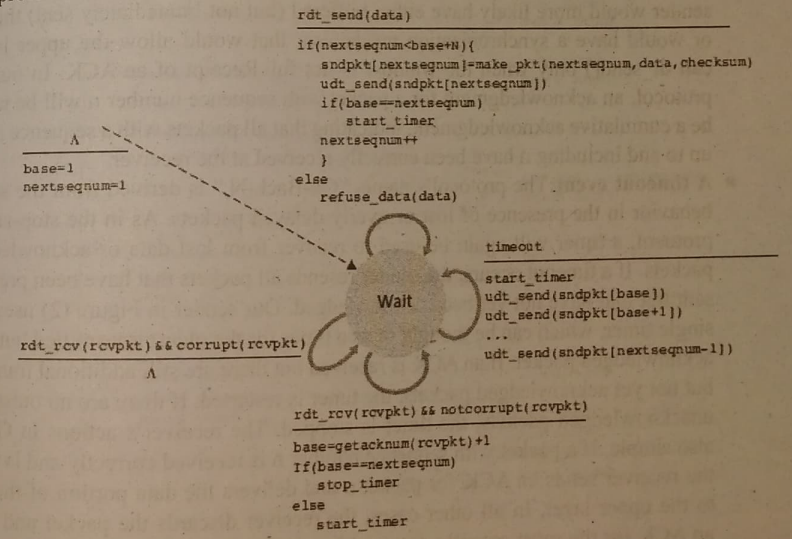
Figure (1) shows the sender's view of the range of sequence numbers in a GBN protocol. If we define base to be the sequence number of the oldest unacknowledged packet and

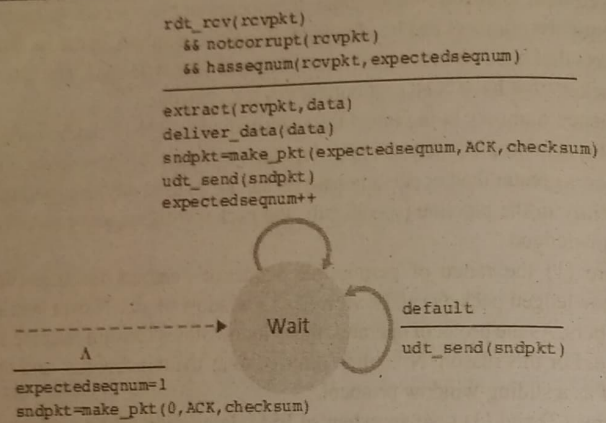
nextseq num to be the smallest unused sequence number then four intervals in the range of sequence numbers can be identified. Sequence numbers in the interval correspond to packets that have already been transmitted and acknowledged. The interval corresponds to packets that have been sent but not yet acknowledged.

Sequence numbers in the interval [nextseqnum,base+N-1] can be used for packets that can be sent immediately, should data arrive from the upper layer. Finally, sequence numbers greater than or equal to base+N cannot be used until an unacknowledged packet currently in the pipeline (specifically, the packet with sequence number base) has been acknowledged.

Figure (1) the range of permissible sequence numbers for transmitted but not yet acknowledged packets can be viewed as a window of size N over the range of sequence numbers. As the protocol operates, this window slides forward over the sequence number space. For this reason, N is often referred to as the window size and the GBN protocol itself as a sliding-window protocol.

Figures (2) and (3) give an extended FSM description of the sender and receiver sides of an ACK-based, NAK-free, GBN protocol. We refer to this FSM description as an extended FSM because we have added variables for base and nextseqnum, and added operations on these variables and conditional actions involving these variables. Note that the extended FSM specification is now beginning to look somewhat like a programming language specification. provides an excellent survey of additional extensions to FSM techniques as well as other programming-language-based techniques for specifying protocols.





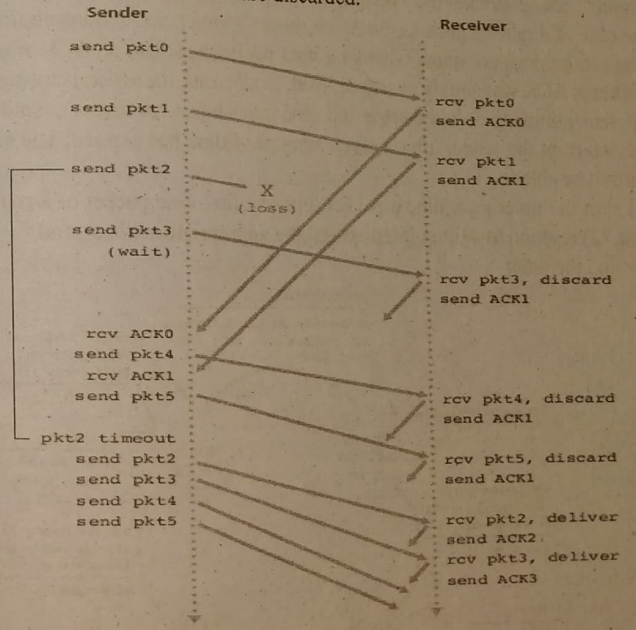
The GBN sender must respond to three types of events:

- **Invocation from above**: When rdt_send() is called from above, the sender first checks to see if the indow is full, that is, whether there are N outstanding, unacknowledged packets. If the window is not full, a packet is created and sent, and variables are appropriately updated. If the window is full, the sender simply returns the data back to the upper layer, an implicit indication that the window is full. The upper layer would presumably then have to try again later. In a real implementation, the sender would more likely have either buffered (but not immediately sent) this data, or would have a synchronization mechanism that would allow the upper layer to call dt_send() only when the window is not full.Receipt of an ACK. In our GBN protocol, an acknowledgment for a packet with sequence number n will be taken to be a cumulative acknowledgment, indicating that all packets with a sequence number up to and including n have been correctly received at the receiver.
- **A timeout event:**The protocol's name, "Go-Back-N," is derived from the sender's behavior in the presence of lost or overly delayed packets. As in the stop-and-wait protocol, a timer will again be used to recover from lost data or acknowledgment packets. If a timeout occurs, the sender resends all packets that have been previously sent but that have not yet been acknowledged. Our sender in Figure (2) uses only a single timer, which can be thought of as a timer for the oldest transmitted but not yet acknowledged packet. If an ACK is received but there are still additional transmitted but not yet acknowledged packets, the timer is restarted. If there are no outstanding, unacknowledged packets, the timer is stopped. The receiver's actions in GBN are also simple. If a packet with sequence number n is received correctly and is in order the receiver sends an ACK for packet n and delivers the data portion of the packet to the upper layer. In all other cases, the receiver discards the packet and resends an ACK for the most recently received in-order packet. Note that since packets are delivered one at a time to the upper layer, if packet k has been received and delivered, then all packets with a sequence number lower than k have also been delivered. Thus, the use of cumulative acknowledgments is a natural choice for GBN. In our GBN protocol, the receiver discards out-of-order packets. The receiver must deliver data in order to the upper layer. Suppose now that packet n is expected, but packet n + 1

arrives. Because data must be delivered in order, the receiver could buffer received and delivered packet n.if packet n is lost, both it and packet n + 1 will eventually be retransmitted as a result of the GBN retransmission rule at the sender. Thus, the receiver can simply discard packet n + 1.

The advantage of this approach is the simplicity of receiver buffering—the receiver need not buffer any out-of-order packets. Thus, while the sender must maintain the upper and lower bounds of its window and the position of nextseqnum within this window, the only piece of information the receiver need maintain is the sequence number of the next in-order packet. This value is held in the variable expectedseqnum, shown in the receiver FSM in Figure (3). The disadvantage of throwing away a correctly received packet is that the subsequent retransmission of that packet might be lost or garbled and thus even more retransmissions would be required.save) packet n + 1 and then deliver this packet to the upper layer after it had later

Figure (4) shows the operation of the GBN protocol for the case of a window size of four packets. Because of this window size limitation, the sender sends packets 0 through 3 but then must wait for one or more of these packets to be acknowledged before proceeding. As each successive ACK is received, the window slides forward and the sender can transmit one new packet. On the receiver side, packet 2 is lost and thus packets 3, 4, and 5 are found to be out of order and are discarded.



b. Explain Reliable data transfer over a lossy channel with bit errors? (06 Marks)

Ans. Two additional concerns must now be addressed by the protocol: how to detect packet loss and what to do when packet loss occurs. The use of check summing, sequence numbers, ACK packets, and retransmissions the techniques already developed in rdt2.2 will allow us to answer the latter concern. Handling the first concern will require adding

a new protocol mechanism. There are many possible approaches toward dealing with packet loss detecting and recovering from lost packets on the sender Suppose that the sender transmits a data packet and either that packet, or the receiver's ACK of that packet, gets lost. In either case, no reply is forthcoming at the sender from the receiver. If the sender is willing to wait long enough so that it is certain that a packet has been lost, it can simply retransmit the data packet. You should convince yourself that this protocol does indeed work. The sender must clearly wait at least as long as a round-trip delay between the sender and receiver plus whatever amount of time is needed to process a packet at the receiver.

In many networks, this worst-case maximum delay is very difficult even to estimate, much less know with certainty. Moreover, the protocol should ideally recover from packet loss as soon as possible; waiting for a worst-case delay could mean a long wait until error recovery is initiated. The approach thus adopted in practice is for the sender to judiciously choose a time value such that packet loss is likely, although not guaranteed, to have happened. If an ACK is not received within this time, the packet is retransmitted. Note that if a packet experiences a particularly large delay, the sender may retransmit the packet even though neither the data packet nor its ACK have been lost. This introduces the possibility of duplicate data packets in the sender-to-receiver channel. Happily, protocol rdt2.2 already has enough functionality (that is, sequence numbers) to handle the case of duplicate packets. From the sender's viewpoint, retransmission is a panacea. The sender does not know whether a data packet was lost, an ACK was lost, or if the packet or ACK was simply overly delayed. In all cases, the action is the same: retransmit. Implementing a time-based retransmission mechanism requires a countdown timer that can interrupt the sender after a given amount of time has expired. The sender will thus need to be able to

(1) start the timer each time a packet (either a first-time packet or a retransmission) is sent, (2) respond to a timer interrupt (taking appropriate actions), and

(3) stop the timer

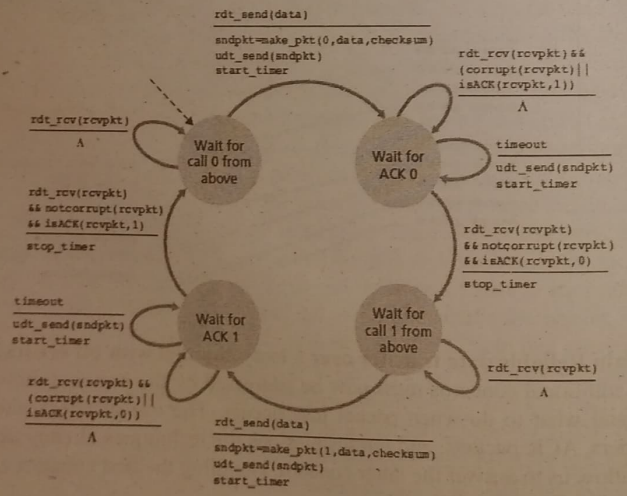


Figure (1) shows the sender FSM for rdt3.0, a protocol that reliably transfers data over a channel that can corrupt or lose packets; in the homework problems, you'll be asked to provide the receiver FSM for rdt3.0. Figure 3.16 shows how the protocol operates with no lost or delayed packets and how it handles lost data packets. In Figure time moves forward from the top of the diagram toward the bottom of the diagram; note that a receive time for a packet is necessarily later than the send time for a packet as a result of transmission and propagation delays. In Figures the send-side brackets indicate the times at which a timer is set and later times out. Several of the more subtle aspects of this protocol are explored in the exercises at the end of this chapter. Because packet sequence numbers alternate between 0 and 1, protocol rdt3.0 is sometimes known as the alternating-bit protocol. We have now assembled the key elements of a data transfer protocol. Checksums, sequence numbers, timers, and positive and negative acknowledgment packets each play a crucial and necessary role in the operation of the protocol. We have a working reliable data transfer protocol.

OR

4. a. What is Pipelining? What are the consequences of pipelining for reliable data transfer protocols? (0 4 Marks)

Ans. If the sender is allowed to transmit three packets before having to wait for acknowledgments, the utilization of the sender is essentially tripled. Since the many in-transit sender-to-receiver packets can be visualized as filling a pipeline, this technique is known as **pipelining.**

Pipelining has the following consequences for reliable data transfer protocols:

- The range of sequence numbers must be increased, since each in-transit packet must have a unique sequence number and there may be multiple, in-transit, unacknowledged packets.
- The sender and receiver sides of the protocols may have to buffer more than one packet. Minimally, the sender will have to buffer packets that have been transmitted but not yet acknowledged. Buffering of correctly received packets may also be needed at the receiver.
- The range of sequence numbers needed and the buffering requirements will depend on the manner in which a data transfer protocol responds to lost, corrupted, and overly delayed packets. Two basic approaches toward pipelined error recovery can be identified: Go-Back-N and selective repeat.

b. Explain Selective Repeat Sender and Receiver events and actions? (06 Marks)

Ans. Figure (1) details the various actions taken by the SR sender. The SR receiver will acknowledge a correctly received packet whether or not it is in order. Out-of-order packets are buffered until any missing packets (that is, packets with lower sequence numbers) are received, at which point a batch of packets can be delivered in order to the upper layer. Figure (1) itemizes the various actions taken by the SR receiver. It is important to note that in Step 2 in Figure (1), the receiver reacknowledges already received packets with certain sequence numbers below the current window base. You should convince yourself that this reacknowledgment is indeed needed.

1. Data received from above. When data is received from above, the SR sender checks the next available sequence number for the packet. If the sequence number is within the sender's window, the data is packetized and sent; otherwise it is either buffered or returned to the upper layer for later transmission, as in GBN.
2. Timeout. Timers are again used to protect against lost packets, However, each packet must now have its own logical timer, since only a single packet will be transmitted on timeout. A single hardware timer can be used to mimic the operation of multiple logical timers [Varghese 1997].
3. ACK received. If an ACK is received, the SR sender inarks that packet as having been received, provided it is in the window. If the packet's sequence number is equal to send_ base, the window base is moved forward to the unacknowledged packet with the smallest sequence number, [f the window moves and there are untransmitted packets with sequence numbers that now fall within the window, these packets are transmitted.

Figure (1) SR sender events and actions

1. Packet with sequence number in [rcv_ba.se, rcv_base+N-l] is correctly received. In this case, the received packet falls within the receiver's window and a selective ACK packet is returned to the sender. If the packet was not previously received, it is buffered. If this packet has a sequence number equal to the base of the receive window (rcv_ba.se in Figure 3.22), then this packet, and any previously buffered and consecutively numbered (beginning with rcv_base) packets are delivered to the upper layer. The receive window is then moved forward by die number of packets delivered to the upper layer. As an example, consider Figure 3.26. When a packet with a sequence number of rcv_base=2 is received, it and packets 3, 4, and 5 can be delivered to die upper layer.
2. Packet with sequence number iti [rcv_base-S, rcv_base-l] is correctly received. In this case, an ACK must be generated, even though this is a packet that the receiver has previously acknowledged.
3. Otherwise. Ignore the packet.

Figure (2)SR receiver events and actions

c. Explain the three way Handshake for establishing a TCP Connection?

(06 Marks)

Ans. TCP connection is established. Suppose a process running in one host (client) wants to initiate a connection with another process in another host (server). The client application process first informs the client TCP that it wants to establish a connection to a process in the server. The TCP in the client then proceeds to establish a TCP connection with the TCP in the server in the following manner:

- **Step 1.** The client-side TCP first sends a special TCP segment to the server-side TCP. This special segment contains no application-layer data. But one of the flag bits in the segment's header, the SYN bit, is set to 1. For this reason, this special segment is referred to as a SYN segment. In addition, the client randomly chooses an initial sequence number and puts this number in the sequence number field of the initial TCP SYN segment. This segment is encapsulated within an IP datagram and sent to the server. There has been considerable interest in properly randomizing the choice of the client_isn in order to avoid certain security attacks.
- **Step 2.** Once the IP datagram containing the TCP SYN segment arrives at the server

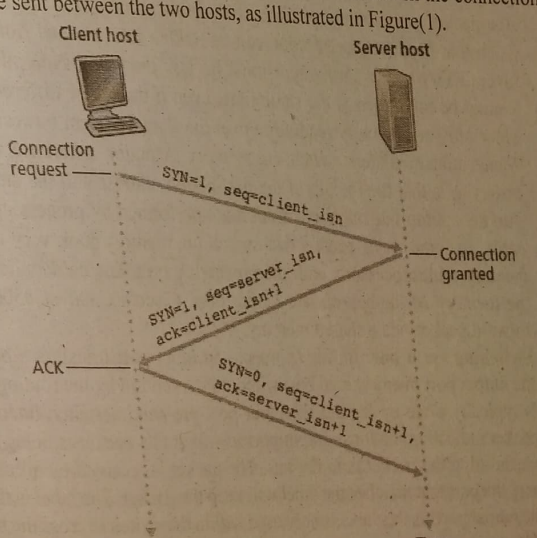
host the server extracts the TCP SYN segment from the datagram, allocates the TCP buffers and variables to the connection, and sends a connection-granted segment to the client TCP.This connection-granted segment also contains no application layer data. However, it does contain three important pieces of information in the segment header.

First, the SYN bit is set to 1.

Second, the acknowledgment field of the TCP segment header is set to client_isn+1.

Finally, the server chooses its own initial sequence number and puts this value in the sequence number field of the TCP segment header. This connection-granted segment is saying, in effect, "I received your SYN packet to start a connection with your initial sequence number, client_isn. I agree to establish this connection. My own initial sequence number is server_isn." The connectiong ranted segment is referred to as a SYNACK segment.

- **Step 3.** Upon receiving the SYNACK segment, the client also allocates buffers and variables to the connection. The client host then sends the server yet another segment; this last segment acknowledges the server's connection-granted segment (the client does so by putting the value server_isn+1 in the acknowledgment field of the TCP segment header). The SYN bit is set to zero, since the connection is established. This third stage of the three-way handshake may carry client-to-server data in the segment payload. Once these three steps have been completed, the client and server hosts can send segments containing data to each other. In each of these future segments, the SYN bit will be set to zero. Note that in order to establish the connection, three packets are sent between the two hosts, as illustrated in Figure(1).



For this reason, this connection establishment procedure is often referred to as a three-way handshake. It's interesting to note that a rock climber and a belayer use a three-way handshake communication protocol that is identical to TCP's to ensure that both sides

are ready before the climber begins ascent. All good things must come to an end, and the same is true with a TCP connection. Either of the two processes participating in a TCP connection can end the connection. When a connection ends, the "resources" in the hosts are deallocated. As an example, suppose the client decides to close the connection. The client application process issues a close command. This causes the client TCP to send a special TCP segment to the server process. This special segment has a flag bit in the segment's header, the FIN bit, set to 1. When the server receives this segment, it sends the client an acknowledgment segment in return. The server then sends its own shutdown segment, which has the FIN bit set to 1. Finally, the client acknowledges the server's shutdown segment. At this point, all the resources in the two hosts are now deallocated.

MODULE - 3

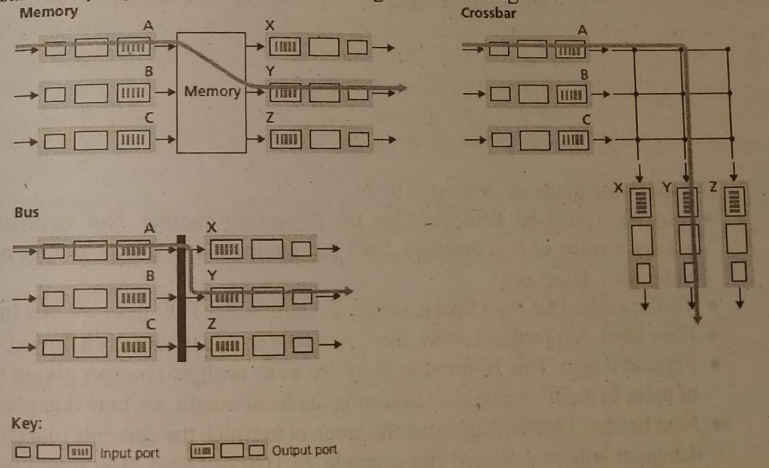
5. a. What are the 3 switching techniques? Explain? **(06 Marks)**

Ans. Switching can be accomplished in a number of ways, as shown in Figure (1).

- **Switching via memory**: The simplest, earliest routers were traditional computers, with switching between input and output ports being done under direct control of the CPU . Input and output ports functioned as traditional I/O devices in a traditional operating system. An input port with an arriving packet first signaled the routing processor via an interrupt. The packet was then copied from the input port into processor memory. The routing processor then extracted the destination address from the header, looked up the appropriate output port in the forwarding table, and copied the packet to the output port's buffers. In this scenario, if the memory bandwidth is such that B packets per second can be written into, or read from, memory, then the overall forwarding throughput must be less than B/2. Note also that two packets cannot be forwarded at the same time, even if they have different destination ports, since only one memory read/write over the shared system bus can be done at a time. Many modern routers switch via memory. A major difference from early routers, however, is that the lookup of the destination address and the storing of the packet into the appropriate memory location are performed by processing on the input line cards. In some ways, routers that switch via memory look very much like shared-memory multiprocessors, with the processing on a line card switching packets into the memory of the appropriate output port. Cisco's Catalyst 8500 series switches forward packets via a shared memory.
- **Switching via a bus:** In this approach, an input port transfers a packet directly to the output port over a shared bus, without intervention by the routing processor. This is typically done by having the input port pre-pend a switch-internal label to the packet indicating the local output port to which this packet is being transferred and transmitting the packet onto the bus. The packet is received by all output ports, but only the port that matches the label will keep the packet. The label is then removed at the output port, as this label is only used within the switch to cross the bus. If multiple packets arrive to the router at the same time, each at a different input port, all but one must wait since only one packet can cross the bus at a time. Because every packet must cross the single bus, the switching speed of the router is limited to the bus speed; in our roundabout analogy, this is as if the roundabout could only contain one car at a time.switching via a bus is often sufficient for routers that operate in small local area

and enterprise networks. The Cisco 5600 switches packets over a 32 Gbps backplane bus.

- **Switching via an interconnection network:** One way to overcome the bandwidth limitation of a single, shared bus is to use a more sophisticated interconnection network, such as those that have been used in the past to interconnect processors in a multiprocessor computer architecture. A crossbar switch is an interconnection network consisting of 2N buses that connect N input ports to N output ports, as shown in Figure (1). Each vertical bus intersects each horizontal bus at a crosspoint, which can be opened or closed at any time by the switch fabric controller.When a packet arrives from port A and needs to be forwarded to port Y, the switch controller closes the crosspoint at the intersection of busses Aand Y, and port A then sends the packet onto its bus, which is picked up by bus Y. Note that a packet from port B can be forwarded to port X at the same time, since the A-to-Y and B-to-X packets use different input and output busses. Thus, unlike the previous two switching approaches, crossbar networks are capable of forwarding multiple packets in parallel. However, if two packets from two different input ports are destined to the same output port, then one will have to wait at the input, since only one packet can be sent over any given bus at a time. More sophisticated interconnection networks use multiple stages of switching elements to allow packets from different input ports to proceed towards the same output port at the same time through the switching fabric.



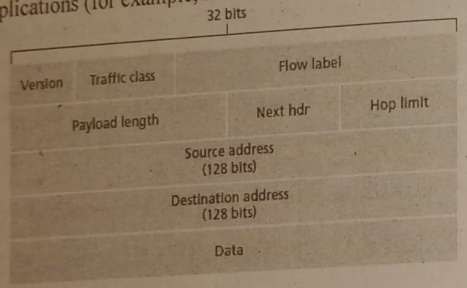
b. Explain IPV6 Header Format? **(10 Marks)**

Ans. The format of the IPv6 datagram is shown in Figure (1). The most important changes introduced in IPv6 are evident in the datagram format:

- Expanded addressing capabilities. IPv6 increases the size of the IP address from 32 to 128 bits. This ensures that the world won't run out of IP addresses. Now, every grain of sand on the planet can be IP-addressable. In addition to unicast and multicast addresses, IPv6 has introduced a new type of address, called an any cast address, which allows a datagram to be delivered to any one of a group of hosts.
- A streamlined 40-byte header. As discussed below, a number of IPv4 fields have been

dropped or made optional. The resulting 40-byte fixed-length header allows for faster processing of the IP datagram. A new encoding of options allows for more flexible options processing.

- Flow labeling and priority. IPv6 has an elusive definition of a flow. RFC 1752 and RFC 2460 state that this allows "labeling of packets belonging to particular flows for which the sender requests special handling, such as a non default quality of service or real-time service." For example, audio and video transmission might likely be treated as a flow. On the other hand, the more traditional applications, such as file transfer and e-mail, might not be treated as flows. It is possible that the traffic carried by a high-priority user might also be treated as a flow. What is clear, however, is that the designers of IPv6 foresee the eventual need to be able to differentiate among the flows, even if the exact meaning of a flow has not yet been determined. The IPv6 header also has an 8-bit traffic class field. This field, like the TOS field in IPv4, can be used to give priority to certain datagrams within a flow, or it can be used to give priority to datagrams from certain applications (for example, ICMP) over datagrams from other applications (for example, network news).



The following fields are defined in IPv6:

- Version. This 4-bit field identifies the IP version number. Not surprisingly, IPv6 carries a value of 6 in this field. Note that putting a 4 in this field does not create a valid IPv4 datagram.
- Traffic class. This 8-bit field is similar in spirit to the TOS field we saw in IPv4.
- Flow label. As discussed above, this 20-bit field is used to identify a flow of datagrams.
- Payload length. This 16-bit value is treated as an unsigned integer giving the number of bytes in the IPv6 datagram following the fixed-length, 40-byte datagram header.
- Next header. This field identifies the protocol to which the contents (data field) of this datagram will be delivered (for example, to TCP or UDP). The field uses the same values as the protocol field in the IPv4 header.
- Hop limit. The contents of this field are decremented by one by each router that forwards the datagram. If the hop limit count reaches zero, the datagram is discarded.
- Source and destination addresses. The various formats of the IPv6 128-bit address are described in RFC 4291.
- Data. This is the payload portion of the IPv6 datagram. When the datagram reaches its destination, the payload will be removed from the IP datagram and passed on to the protocol specified in the next header field.

OR

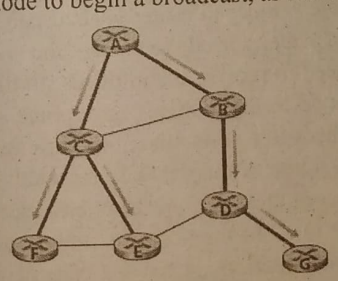
6. a. **What are the services provided by IP Sec?** **(04 Marks)**

Ans. The services provided by an IPsec session include:

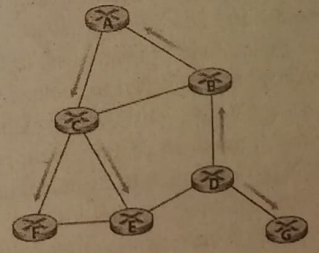
- **Cryptographic agreement--** Mechanisms that allow the two communicating hosts to agree on cryptographic algorithms and keys.
- **Encryption of IP datagram payloads--**When the sending host receives a segment from the transport layer, IPsec encrypts the payload. The payload can only be decrypted by IPsec in the receiving host.
- **Data integrity--** IPsec allows the receiving host to verify that the datagram's header fields and encrypted payload were not modified while the datagram was en route from source to destination.
- **Origin authentication--** When a host receives an IPsec datagram from a trusted source the host is assured that the source IP address in the datagram is the actual source of the datagram

b. **Explain Spanning Tree BroadCast Routing algorithms?** **(06 Marks)**

Ans. While sequence-number-controlled flooding and RPF avoid broadcast storms, they do not completely avoid the transmission of redundant broadcast packets. The tree consisting of the nodes connected by thick lines in Figure(a), you can see that if broadcast packets were forwarded only along links within this tree, each and every network node would receive exactly one copy of the broadcast packet—exactly the solution we were looking for! This tree is an example of a spanning tree—a tree that contains each and every node in a graph. More formally, a spanning tree of a graph G = (N,E) is a graph G_ = (N,E_) such that E_ is a subset of E, G_ is connected, G_ contains no cycles, and G_ contains all the original nodes in G. If each link has an associated cost and the cost of a tree is the sum of the link costs, then a spanning tree whose cost is the minimum of all of the graph's spanning trees is called a minimum spanning tree. Another approach to providing broadcast is for the network nodes to first construct a spanning tree. When a source node wants to send a broadcast packet, it sends the packet out on all of the incident links that belong to the spanning tree. A node receiving a broadcast packet then forwards the packet to all its neighbors in the spanning tree .Not only does spanning tree eliminate redundant broadcast packets, but once in place, the spanning tree can be used by any node to begin a broadcast, as shown in Figures



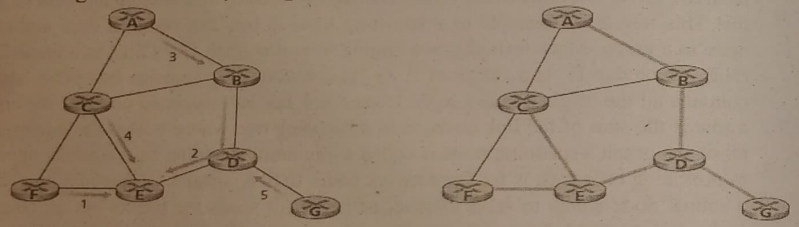
a. Broadcast initiated at A



b. Broadcast initiated at D

Note that a node need not be aware of the entire tree; it simply needs to know which of its neighbors in G are spanning-tree neighbors. The main complexity associated with the

spanning-tree approach is the creation and maintenance of the spanning tree. Numerous distributed spanning-tree algorithms have been developed. We consider only one simple algorithm here. In the center-based approach to building a spanning tree, a center node is defined. Nodes then unicast tree-join messages addressed to the center node. A tree-join message is forwarded using unicast routing toward the center until it either arrives at a node that already belongs to the spanning tree or arrives at the center. In either case, the path that the tree-join message has followed defines the branch of the spanning tree between the edge node that initiated the tree-join message and the center. One can think of this new path as being grafted onto the existing spanning tree. Figure illustrates the construction of a center-based spanning tree. Suppose that node E is selected as the center of the tree. Suppose that node F first joins the tree and forwards a tree-join message to E. The single link EF becomes the initial spanning tree. Node B then joins the spanning tree by sending its tree-join message to E. Suppose that the unicast path route to E from B is via D. In this case, the tree-join message results in the path BDE being grafted onto the spanning tree. Node A next joins the spanning group by forwarding its tree-join message towards E. If A's unicast path to E is through B, then since B has already joined the spanning tree, the arrival of A's tree-join message at B will result in the AB link being immediately grafted onto the spanning tree. Node C joins the spanning tree next by forwarding its tree-join message directly to E. Finally, because the unicast routing from G to E must be via node D, when G sends its tree-join message to E, the GD link is grafted onto the spanning tree at node D.

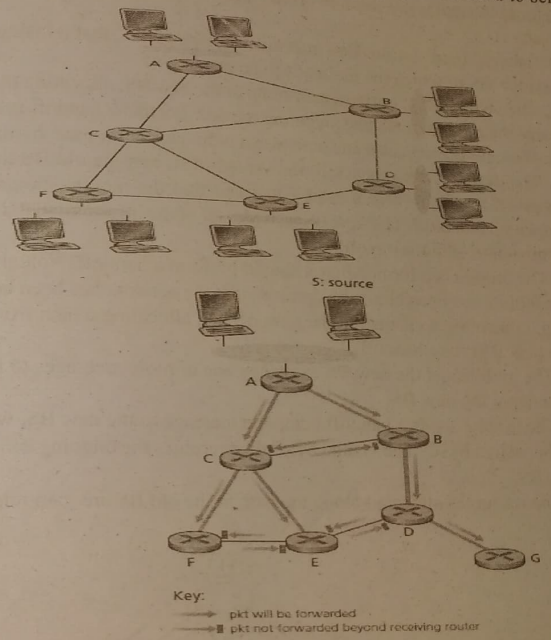


a. Stepwise construction of spanning tree b. Constructed spanning tree

c. Explain Multicast Routing algorithm? (06 Marks)

Ans. The multicast routing problem is illustrated in Figure. Hosts joined to the multicast group are shaded in color; their immediately attached router is also shaded in color. As shown in Figure, only a subset of routers actually needs to receive the multicast traffic. In Figure, only routers A, B, E, and F need to receive the multicast traffic. Since none of the hosts attached to router D are joined to the multicast group and since router C has no attached hosts, neither C nor D needs to receive the multicast group traffic. The goal of multicast routing, then, is to find a tree of links that connects all of the routers that have attached hosts belonging to the multicast group. Multicast packets will then be routed along this tree from the sender to all of the hosts belonging to the multicast tree. Of course, the tree may contain routers that do not have attached hosts belonging to the multicast group. Two approaches have been adopted for determining the multicast routing tree. The two approaches differ according to whether a single group-shared tree is used to distribute the traffic for all senders in the group, or whether a source-specific routing tree is constructed for each individual sender.

Multicast routing using a group-shared tree. As in the case of spanning-tree broadcast, multicast routing over a group-shared tree is based on building a tree that includes all edge routers with attached hosts belonging to the multicast group. A center-based approach is used to construct the multicast routing tree, with edge routers with attached hosts belonging to the multicast group sending join messages addressed to the center node. As in the broadcast case, a join message is forwarded using unicast routing toward the center until it either arrives at a router that already belongs to the multicast tree or arrives at the center. All routers along the path that the join message follows will then forward received multicast packets to the edge router that initiated the multicast join. A critical question for center-based tree multicast routing is the process used to select the center.



Multicast routing using a source-based tree. While group-shared tree multicast routing constructs a single, shared routing tree to route packets from all senders, the second approach constructs a multicast routing tree for each source in the multicast group. In practice, an RPF algorithm is used to construct a multicast forwarding tree for multicast datagrams originating at source x. The RPF broadcast algorithm we studied earlier requires a bit of tweaking for use in multicast.consider router D in Figure. Under broadcast RPF, it would forward packets to router G, even though router G has no attached hosts that are joined to the multicast group. While this is not so bad for this case where D has only a single downstream router, G, imagine what would happen if there were thousands of

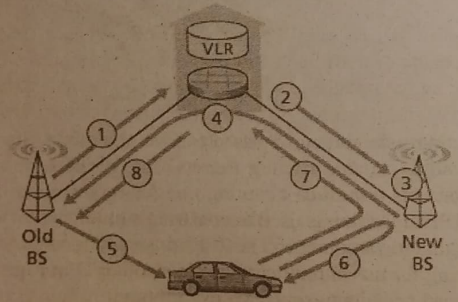
routers downstream from D! Each of these thousands of routers would receive unwanted multicast packets. The solution to the problem of receiving unwanted multicast packets and under RPF is known as pruning. A multicast router that receives multicast packets and has no attached hosts joined to that group will send a prune message to its upstream router. If a router receives prune messages from each of its downstream routers, then it can forward a prune message upstream.

MODULE - 4

7. a. List out the steps in accomplishing a handoff between base stations with a common MSC? (10 Marks)

Ans. Figure illustrates the steps involved when a base station does decide to hand off a mobile user:

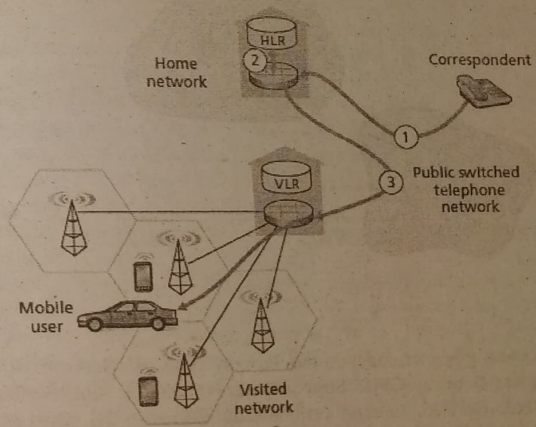
1. The old base station (BS) informs the visited MSC that a handoff is to be performed and the BS to which the mobile is to be handed off.
2. The visited MSC initiates path setup to the new BS, allocating the resources needed to carry the rerouted call, and signaling the new BS that a handoff is about to occur.
3. The new BS allocates and activates a radio channel for use by the mobile.
4. The new BS signals back to the visited MSC and the old BS that the visited- MSC-to-new-BS path has been established and that the mobile should be informed of the impending handoff. The new BS provides all of the information that the mobile will need to associate with the new BS.
5. The mobile is informed that it should perform a handoff. Note that up until this point, the mobile has been blissfully unaware that the network has been laying the groundwork (e.g., allocating a channel in the new BS and allocating a path from the visited MSC to the new BS) for a handoff.
6. The mobile and the new BS exchange one or more messages to fully activate the new channel in the new BS.
7. The mobile sends a handoff complete message to the new BS, which is forwarded up to the visited MSC. The visited MSC then reroutes the ongoing call to the mobile via the new BS.
8. The resources allocated along the path to the old BS are then released.



b. Explain how a call is placed to a mobile user in a visited network? (06 Marks)

Ans. Consider a simple example below; more complex scenarios are described in the steps, as illustrated in Figure 6.29, are as follows:

1. The correspondent dials the mobile user's phone number. This number itself does not refer to a particular telephone line or location. The leading digits in the number are sufficient to globally identify the mobile's home network. The call is routed from the correspondent through the PSTN to the home MSC in the mobile's home network. This is the first leg of the call.
2. The home MSC receives the call and interrogates the HLR to determine the location of the mobile user. In the simplest case, the HLR returns the mobile station roaming number (MSRN), which we will refer to as the roaming number. Note that this number is different from the mobile's permanent phone number, which is associated with the mobile's home network. The roaming number is ephemeral: It is temporarily assigned to a mobile when it enters a visited network. The roaming number serves a role similar to that of the care-of address in mobile IP and, like the COA, is invisible to the correspondent and the mobile. If HLR does not have the roaming number, it returns the address of the VLR in the visited network. In this case, the home MSC will need to query the VLR to obtain the roaming number of the mobile node. But how does the HLR get the roaming number or the VLR address in the first place? What happens to these values when the mobile user moves to another visited network?
3. Given the roaming number, the home MSC sets up the second leg of the call through the network to the MSC in the visited network. The call is completed, being routed from the correspondent to the home MSC, and from there to the visited MSC, and from there to the base station serving the mobile user.



An unresolved question in step 2 is how the HLR obtains information about the location of the mobile user. When a mobile telephone is switched on or enters a part of a visited network that is covered by a new VLR, the mobile must register with the visited network. This is done through the exchange of signaling messages between the mobile and the VLR. The visited VLR, in turn, sends a location update request message to the mobile's HLR. This message informs the HLR of either the roaming number at which the mobile

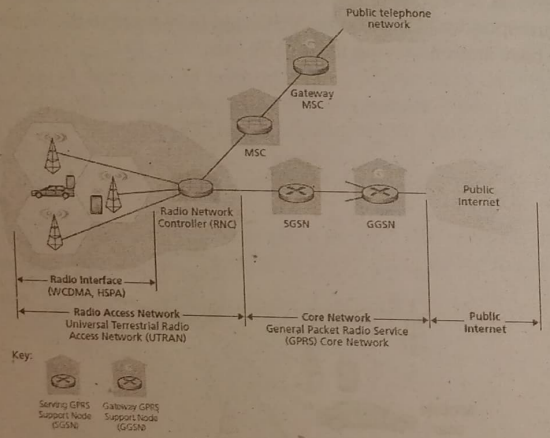
can be contacted, or the address of the VLR. As part of this exchange, the VLR also obtains subscriber information from the HLR about the mobile and determines what services (if any) should be accorded the mobile user by the visited network.

OR

8. a. What is a foreign agent? Explain GSM 2.5 G Cellular Voice and data network architecture (06 Marks)

Ans. The network in which the mobile node is currently residing is known as the foreign (or visited) network,

The 3G core cellular data network connects radio access networks to the public Internet. The core network interoperates with components of the existing cellular voice network Given the considerable amount of existing infrastructure in the existing cellular voice network, the approach taken by the designers of 3G data services is clear: leave the existing core GSM cellular voice network untouched, adding additional cellular data functionality in parallel to the existing cellular voice network. The alternative integrating new data services directly into the core of the existing cellular voice network would have raised the same challenges encountere where we discussed integrating new (IPv6) and legacy (IPv4) technologies in the Internet.



There are two types of nodes in the 3G core network: Serving GPRS Support Nodes (SGSNs) and Gateway GPRS Support Nodes (GGSNs). (GPRS stands for Generalized Packet Radio Service, an early cellular data service in 2G networks; here we discuss the evolved version of GPRS in 3G networks). An SGSN is responsible for delivering datagrams to/from the mobile nodes in the radio access network to which the SGSN is attached. The SGSN interacts with the cellular voice network's MSC for that area, providing user authorization and handoff, maintaining location (cell) information about active mobile nodes, and performing datagram forwarding between mobile nodes in the radio access network and a GGSN. The GGSN acts as a gateway, connecting multiple

SGSNs into the larger Internet. A GGSN is thus the last piece of 3G infrastructure that a datagram originating at a mobile node encounters before entering the larger Internet. To the outside world, the GGSN looks like any other gateway router; the mobility of the 3G nodes within the GGSN's network is hidden from the outside world behind the GGSN.

b. Write a short note on Cellular network architecture?(06 Marks)

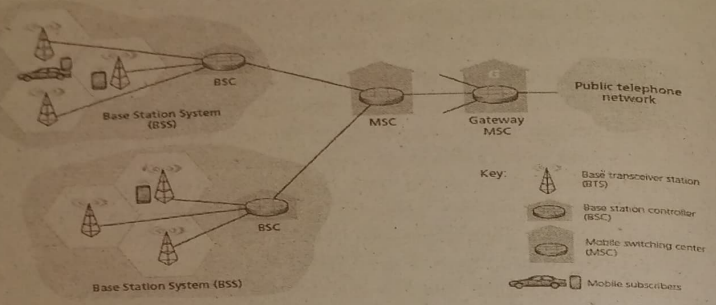
Ans. Cellular technology classify the technology as belonging to one of several "generations." The earliest generations were designed primarily for voice traffic. First generation (1G) systems were analog FDMA systems designed exclusively for voice-only communication. These 1G systems are almost extinct now, having been replaced by digital 2G systems. The original 2G systems were also designed for voice, but later extended (2.5G) to support data (i.e., Internet) as well as voice service. The 3G systems that currently are being deployed also support voice and data, but with an ever increasing emphasis on data capabilities and higher- speed radio access links.

Cellular Network Architecture, 2G: Voice Connections to the Telephone Network

The term cellular refers to the fact that the region covered by a cellular network is partitioned into a number of geographic coverage areas, known as cells, shown as hexagons on the left side of Figure. GSM has its own particular nomenclature. Each cell contains a base transceiver station (BTS) that transmits signals to and receives signals from the mobile stations in its cell. The coverage area of a cell depends on many factors, including the transmitting power of the BTS, the transmitting power of the user devices, obstructing buildings in the cell, and the height of base station antennas. Figure shows each cell containing one base transceiver station residing in the middle of the cell, many systems today place the BTS at corners where three cells intersect, so that a single BTS with directional antennas can service three cells. The GSM standard for 2G cellular systems uses combined FDM/TDM (radio) for the air interface. Recall from Chapter 1 that, with pure FDM, the channel is partitioned into a number of frequency bands with each band devoted to a call. Pure TDM, time is partitioned into frames with each frame further partitioned into slots and each call being assigned the use of a particular slot in the revolving frame. In combined FDM/TDM systems, the channel is partitioned into a number of frequency sub-bands; within each sub-band, time is partitioned into frames and slots. Thus, for a combined FDM/TDM system, if the channel is partitioned into F sub-bands and time is partitioned into T slots, then the channel will be able to support F.T simultaneous calls. we saw in that cable access networks also use a combined FDM/ TDM approach. GSM systems consist of 200-kHz frequency bands with each band supporting eight TDM calls. GSM encodes speech at 13 kbps and 12.2 kbps.

A GSM network's base station controller (BSC) will typically service several tens of base transceiver stations. The role of the BSC is to allocate BTS radio channels to mobile subscribers, perform paging (finding the cell in which a mobile user is resident), and perform handoff of mobile users. The base station controller and its controlled base transceiver stations collectively constitute a GSM base station system (BSS). The mobile switching center (MSC) plays the central role in user authorization and accounting call establishment and teardown, and handoff. A single MSC will typically contain up to five BSCs, resulting in approximately 200K subscribers per MSC. A cellular provider's network will have a number of MSCs, with special MSCs known as gateway MSCs

connecting the provider's cellular network to the larger public telephone network.



c. What are the common functional elements between mobile IP and GSM mobility? (4 Marks)

Ans.

GSM element	Comment on GSM element	Mobile IP element
Home system	Network to which the mobile user's permanent phone number belongs.	Home network
Gateway mobile switching center or simply home MSC, Home location register (HLR)	Home MSC: point of contact to obtain routable address of mobile user. HLR: database in home system containing permanent phone number, profile information, current location of mobile user, subscription information.	Home agent
Visited system	Network other than home system where mobile user is currently residing.	Visited network.
Visited mobile services switching center, Visitor location register (VLR)	Visited MSC: responsible for setting up calls to/from mobile nodes in cells associated with MSC. VLR: temporary database entry in visited system, containing subscription information for each visiting mobile user.	Foreign agent
Mobile station roaming number (MSRN) or simply roaming number	Routable address for telephone call segment between home MSC and visited MSC, visible to neither the mobile nor the correspondent.	Care-of address

MODULE-5

9. a. What are the three approaches towards providing network-level support for multimedia applications? (06 Marks)

Ans. • **Making the best of best-effort service.**-The application-level mechanisms and infrastructure can be used in a well-dimensioned network where packet loss and excessive end-to-end delay rarely occur. When demand increases are forecasted, the ISPs deploy additional bandwidth and switching capacity to continue to ensure satisfactory delay and packet-loss performance.

- **Differentiated service -** Since the early days of the Internet, it's been envisioned that different types of traffic could be provided with different classes of service, rather than a single one size-fits-all best-effort service. With differentiated service, one type of traffic might be given strict priority over another class of traffic when both types of traffic are queued at a router. For example, packets belonging to a real-time

conversational application might be given priority over other packets due to their stringent delay constraints. Introducing differentiated service into the network will require new mechanisms for packet marking, packet scheduling, and more.

- **Per-connection Quality-of-Service (QoS) Guarantees -** With per-connection QoS guarantees, each instance of an application explicitly reserves end-to-end bandwidth and thus has a guaranteed end-to-end performance. A hard guarantee means the application will receive its requested quality of service (QoS) with certainty. A soft guarantee means the application will receive its requested quality of service with high probability. For example, if a user wants to make a VoIP call from Host A to Host B, the user's VoIP application reserves bandwidth explicitly in each link along a route between the two hosts. But permitting applications to make reservations and requiring the network to honor the reservations requires some big changes. First, we need a protocol that, on behalf of the applications, reserves link bandwidth on the paths from the senders to their receivers. Second, we'll need new scheduling policies in the router queues so that per-connection bandwidth reservations can be honored. Finally, in order to make a reservation, the applications must give the network a description of the traffic that they intend to send into the network and the network will need to police each application's traffic to make sure that it abides by that description. These mechanisms, when combined, require new and complex software in hosts and routers. Because per-connection QoS guaranteed service has not seen significant deployment

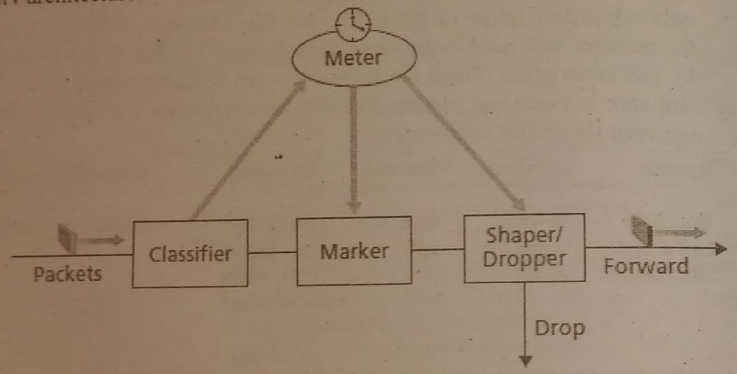
Approach	Granularity	Guarantee	Mechanisms	Complexity	Deployment to date
Making the best of best-effort service.	all traffic treated equally	none, or soft	application-layer support, CDNs, overlays, network-level resource provisioning	minimal	everywhere
Differentiated service	different classes of traffic treated differently	none, or soft	packet-marking, policing, scheduling	medium	some
Per-connection Quality-of-Service (QoS) Guarantees	each source-destination flows treated differently	soft or hard, once flow is admitted	packet marking, policing, scheduling; call admission and signaling	light	little

b. Write Short note on :1)Differenciated Service 2)Diffserv Traffic Classification and Conditioning 3) Per-connection Quality-of-Service (QoS) Guarantees? (10 Marks)

Ans. Diffserv Traffic Classification and Conditioning

Figure 7.26 provides a logical view of the classification and marking functions within the edge router. Packets arriving to the edge router are first classified. The classifier selects packets based on the values of one or more packet header fields (for example, source

address, destination address, source port, destination port, and protocol ID) and steers the packet to the appropriate marking function. As noted above, a packet's marking is carried in the DS field in the packet header. In some cases, an end user may have agreed to limit its packet-sending rate to conform to a declared traffic profile. The traffic profile might contain a limit on the peak rate, as well as the burstiness of the packet flow, as we saw previously with the leaky bucket mechanism. As long as the user sends packets into the network in a way that conforms to the negotiated traffic profile, the packets receive their priority marking and are forwarded along their route to the destination. On the other hand, if the traffic profile is violated, out-of profile packets might be marked differently, might be shaped (for example, delayed so that a maximum rate constraint would be observed), or might be dropped at the network edge. The role of the metering function, shown in Figure 7.26, is to compare the incoming packet flow with the negotiated traffic profile and to determine whether a packet is within the negotiated traffic profile. The actual decision about whether to immediately remark, forward, delay, or drop a packet is a policy issue determined by the network administrator and is not specified in the Diffserv architecture.



Differenciated Service

Since the early days of the Internet, it's been envisioned that different types of traffic could be provided with different classes of service, rather than a single one-size-fits-all best-effort service. With differentiated service, one type of traffic might be given strict priority over another class of traffic when both types of traffic are queued at a router. For example, packets belonging to a real-time conversational application might be given priority over other packets due to their stringent delay constraints. Introducing differentiated service into the network will require new mechanisms for packet marking (indicating a packet's class of service), packet scheduling, and more

Per-connection Quality-of-Service (QoS) Guarantees

With per-connection QoS guarantees, each instance of an application explicitly reserves end to-end bandwidth and thus has a guaranteed end-to-end performance. A hard guarantee means the application will receive its requested quality of service (QoS) with certainty. A soft guarantee means the application will receive its requested quality of service with high probability. For example, if a user wants to make a VoIP call from Host A to Host B, the user's VoIP application reserves bandwidth explicitly in each link

along a route between the two hosts. But permitting applications to make reservations and requiring the network to honor the reservations requires some big changes. First, we need a protocol that, on behalf of the applications, reserves link bandwidth on the paths from the senders to their receivers. Second, we'll need new scheduling policies in the router queues so that per-connection bandwidth reservations can be honored. Finally, in order to make a reservation, the applications must give the network a description of the traffic that they intend to send into the network and the network will need to police each application's traffic to make sure that it abides by that description. These mechanisms, when combined, require new and complex software in hosts and routers. Because per-connection QoS guaranteed service has not seen significant deployment,

OR

10. a. Write Short note on :1 RTSP 2) Internet Telephony? (10 Marks)

Ans. Internet Telephony

Real-time conversational voice over the Internet is often referred to as Internet telephony, since, from the user's perspective, it is similar to the traditional circuitswitched telephone service. It is also commonly called Voice-over-IP (VoIP). Conversational video is similar, except that it includes the video of the participants as well as their voices. Today's voice and video conversational systems allow users to create conferences with three or more participants. Conversational voice and video are widely used in the Internet today, with the Internet companies Skype, QQ, and Google Talk boasting hundreds of millions of daily users. In our discussion of application service requirements, we identified a number of along which application requirements can be classified. Two of these axes timing considerations and tolerance of data loss are particularly important for conversational voice and video applications. Timing considerations are important because audio and video conversational applications are highly delay-sensitive. For a conversation with two or more interacting speakers, the delay from when a user speaks or moves until the action is manifested at the other end should be less than a few hundred milliseconds. For voice, delays smaller than 150 milliseconds are not perceived by a human listener, delays between 150 and 400 milliseconds can be acceptable, and delays exceeding 400 milliseconds can result in frustrating, if not completely unintelligible, voice conversations. On the other hand, conversational multimedia applications are loss-tolerant occasional loss only causes occasional glitches in audio/video playback, and these losses can often be partially or fully concealed. These delay-sensitive but loss-tolerant characteristics are clearly different from those of elastic data applications such as Web browsing, e-mail, social networks, and remote login. For elastic applications, long delays are annoying but not particularly harmful; the completeness and integrity of the transferred data, however, are of paramount importance.

RTSP

The Real-Time Streaming Protocol (RTSP) is a popular open protocol for such a control connection. UDP streaming has been employed in many open-source systems and proprietary products, it suffers from three significant drawbacks.

First, due to the unpredictable and varying amount of available bandwidth between server and client, constant-rate UDP streaming can fail to provide continuous playout. For example, consider the scenario where the video consumption rate is 1 Mbps and the

serverto- client available bandwidth is usually more than 1 Mbps, but every few minutes the available bandwidth drops below 1 Mbps for several seconds. In such a scenario, a UDP streaming system that transmits video at a constant rate of 1 Mbps over RTP/UDP would likely provide a poor user experience, with freezing or skipped frames soon after the available bandwidth falls below 1 Mbps.

The second drawbackof UDP streaming is that it requires a media control server, such as an RTSP server, to process client-to-server interactivity requests and to track client state for each ongoing client session. This increases the overall cost and complexity of deploying a large-scale video-on-demand system.

The third drawback is that many firewalls are configured to block UDP traffic, preventing the users behind these firewalls from receiving UDP video.

b. What is link scheduling discipline? Write a note on Streaming Live audio/Video (06 Marks)

Ans. Packets belonging to various network flows are multiplexed and queued for transmission at the output buffers associated with a link. The manner in which queued packets are selected for transmission on the link is known as the link-scheduling discipline.

This third class of applications is similar to traditional broadcast radio and television, except that transmission takes place over the Internet. These applications allow a user to receive a live radio or television transmission such as a live sporting event or an ongoing news event transmitted from any corner of the world. Thousands of radio and television stations around the world are broadcasting content over the Internet. Live, broadcast-like applications often have many users who receive the same audio/video program at the same time. The distribution of live audio/video to many receivers can be efficiently accomplished using the IP multicasting techniques. multicast distribution is more often accomplished today via application-layer multicast (using P2P networks or CDNs) or through multiple separate unicast streams. As with streaming stored multimedia, the network must provide each live multimedia flow with an average throughput that is larger than the video consumption rate. Because the event is live, delay can also be an issue, although the timing constraints are much less stringent than those for conversational voice. Delays of up to ten seconds or so from when the user chooses to view a live transmission to when playout begins can be tolerated.

Fifth Semester B.E. Degree Examination, CBCS - Dec 2017 / Jan 2018
Computer Networks

Time: 3 hrs. **Max. Marks: 80**

Note : Answer any FIVE full questions, selecting ONE full question from each module.

Module - 1

1. a. **Compare client server and Perl-to-Peer architecture.** **(05 Marks)**

Ans. In client server architecture, there is always on host called server, it services requests from many other hosts called clients. Ex : Web server server a find well known address called IP address.

In P2P architecture, there is minimal reliance on dedicated servers in data centers. The application exploits direct communication between pairs of intermittently connected hosts called peers. Because the peers communicate without passing through a dedicated server communicate without passing through a dedicated server the architecture is called peer to peer. Ex : Bit-torrent one of the feature of P2P is self - scalability.

In client server architecture, a data center housing a large no. of hosts, in often used to create a visual server.

b. Describe HTTP with persistent and non-persistent connections. **(08 Marks)**

Ans. All of the requests and their corresponding responses to be sent over same TCP connection is called HTTP persistent connection. If the requests are sent separate over TCP requests are called HTTP non-persistent connection.

With persistent connections, the server leaves the TCP connection open after sending a response subsequent requests and responses between the same client and server can be sent over the same connection. It uses pipelining concept. The HTTP server closes the connection when it is not used for a certain time.

For non-persistent : Refer Q.No. 2.a. of MQP - 2.

c. What are the services provided by DNS? **(03 Marks)**

Ans. Domain name system (DNS) translates hostname to IP address.

Refer Q.no 1(b) of MQP 2

OR

2. a. **Demonstrate socket implementation using TCP.** **(08 Marks)**

Ans. Three way handshake protocol is implemented in TCP. While the server process is running, the client can initiate a TCP connection to the server.

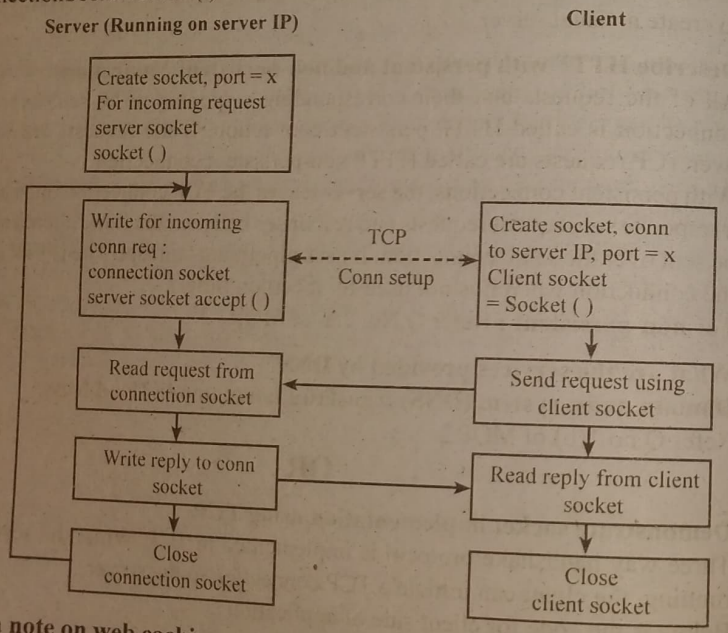
Below is the code for client side of application

```
from socket import *
ServerName = 'Servername'
Serverpost = 12000
ClientSocket = Socket (AF_INET, SOCK_STREAM)
```

```
ClientSocket.Connect((ServerName, ServerPost))
Sentence = raw_input('Input lowercase sentence:')
ClientSocket.send(Senctence)
modifiedSentence = ClientSocket.recv(1024)
Print 'From Server:' , modifiedSentence
ClientSocket.Close ( )
```

Below is the code for server side of application

```
from Socket import *
ServerPort = 12000
ServerSocket = Socket (AF_INET,SOCK_STREAM)
ServerSocket.bind (',ServerPost)
ServerSocket .listen(1)
print 'The server is ready to receive'
While 1 :
ConnectionSocket,addr = ServerSocket.accept ( );
Sentence = ConnectionSocket recv(1024)
Capitialized Sentence = Sentence.upper ( )
Connection Socket Send (CapitalizedSentence)
ConnectionSocket.Close ( )
```



. Write a note on web caching. **(04 Marks)**

Web cache is also called Proxy server satisfies request on behalf of original web server. Ex : Suppose a browser is requesting the object http : //www.SomeSchool.edu/Campus.gif

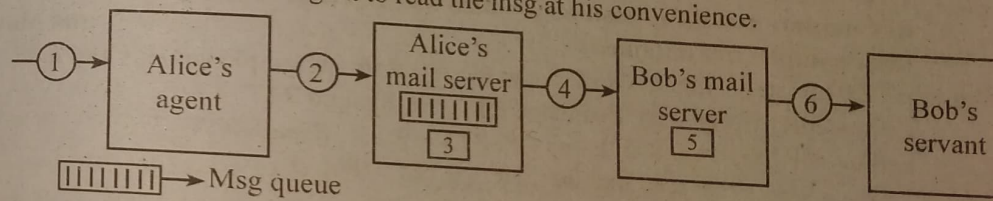
Below shows the steps of occurance

1. The browser establishes a TCP connection to the web cache and sends an HTTP request for the object to the web cache.
2. The web cache checks to see if it has a copy of the object stored locally. If it does, the web cache returns the object within an HTTP response message to the client browser.
3. If the web cache does not have the object, the web cache opens a TCP connection to the origin server i.e., www.someschool.edu. The web cache then sends an HTTP req for the object into the cache - to - server TCP connection. After receiving this request, the origin server sends the object within an HTTP response to the web cache.
4. When the web cache receives the object, it stores a copy in its local storage and sends a copy , within an HTTP response message , to the client browser.

c. Illustrate the basic operation of SMTP with an example. **(04 Marks)**

Ans. SMTP transfer messages from sender's mail server to the reception mail servers. Below ex illustrates the basic of SMTP suppose Alice wants to send a Bob a simple ASCII mesg

1. Alice invokes her user agent for e-mail provides Bob's e-mail address (en bob@someschool.edu) composes a msg and instructs the user agent to send the msg.
2. Alice's user agent sends the msg to mail server, whose it is placed in a msg Queue
3. Client side of SMTP, running on Alice's mail server, sees the msg in msg Queue. It opens a TCP connection to an SMTP server, running on Bob's mail server.
4. After some initial SMTP handshaking , SMTP client sends Alice msg into TCP connection.
5. At Bob's mail server, server side of SMTP receives msg. Bob's mail server then places msg in Bob's mailbox.
6. Bob invokes his user agent to read the msg at his convenience.



Module-2

3. a. Elaborate the three way handshaking in TCP. **(05 Marks)**

Ans. TCP is connection oriented, because the two processes must first "handshake" with each other. The client application process first inform the client transport layer that it wants to establish TCP connection to a process in server.

Socket ClientSocket = new Socket ("host name", postNumber);

where hostname is the name of the server. Transport layer in the client then proceeds to establish TCP connection with TCP in the server. The server responds with a second special TCP segment and finally client responds again with a third segment. The first two segment carry no payload, third of these segment may carry payload. Because three segments are sent between two hosts , the connection establishment

procedure is called three way handshake. Once a TCP connection is established, the two application processes can send data to each other.

b. Discuss Go-Back N protocol. **(06 Marks)**

Ans. Refer Q.No. 3.a. of MQP - 3

c. Explain the connection-oriented multiplexing and de-multiplexing. (05 Marks)

Ans. Refer Q.No. 3.a. of MQP - 1

OR

4. a. State congestion and discuss the cause of congestion. **(04 Marks)**

Ans. Network congestion is too many sources attempting to send data at too high rate. The causes of congestion are two senders, a router with infinite buffer, two senders and a router with finite buffers, four senders routers with finite buffers and multi hop paths. Congestion control techniques are available bit rate (ABR) service in asynchronous transfer mode (ATM). It also has per - connection throughput, offered load. Large queuing delays are experienced as the packet arrival rate nears the link capacity. Delays may cause a router to use its link capacity. Delays may cause a router to use its link bandwidth to forwarded unneeded copin of a packet.

b. With a neat diagram, explain the TCP segment structure. **(08 Marks)**

Ans. Refer Q.No. 3.a. of MQP - 2

c. Suppose that two measured sample RTT values are 106 ms and 120 ms. Compute:

i) Estimated RTT after each of these sample RTT value is obtained. Assume a = 0.125 and estimated RTT is 100 msec just before first of the samples obtained.

ii) Compute DevRTT. Assume p = 0.25 and DevRTT was 5 msec before first of these samples are obtained. **(04 Marks)**

Ans. i) TCP updates estimated RTT acc to the foll formula sample value is 106 ms.

EstimatedRTT = $(1-\alpha)$. Estimated RTT + α. Sample RTT

EstimatedRTT = 0.875. Estimated RTT + 0.125. SampleRTT

= $0.875 \times 100 \times 10^3 + 0.125 \cdot 106(10^3)$

= 193,500.125 = 193 ms

Sampled value is 120 ms

Estimated RTT = $0.875 \times 100 \times 10^3 + 0.125 \times 120 \times 10^3$

= 2,07,500 = 2.07 m/sec

ii. DevRTT = $(1-\beta)$. DevRTT + B./Sample RTT . Estimated RTT)

= $(1 - 0.25) .5 \times 10^3 + 0.25 / 106 \times 10^3 - 193 \times 10^3)$

Module-3

5. a. Write the link-state routing algorithm. Solve the following graph using link-state algorithm with source node 'u'. Fig.Q5(a) **(08 Marks)**

Ans. Algorithm of link state routing is shown as below

1. Initialization
2. $N^1 = \{u\}$

3. For all nodes V
4. If v is a neighbour of u
5. then S(v) = c(u,v)
6. else D(v) = ∞
7.
8. Loop
9. Find w not in N^1 such that D(w) is minimum
10. add W to N^1
11. Update D(v) for each neighbour v of W and not in N^1 =
12. D(v) = min(D(v), D(w) + C(w,v))
13. /* new cost to v is either old cost to v or known
14. path cost to W plus cost from W to v*/
15. Until $N^1 = N$

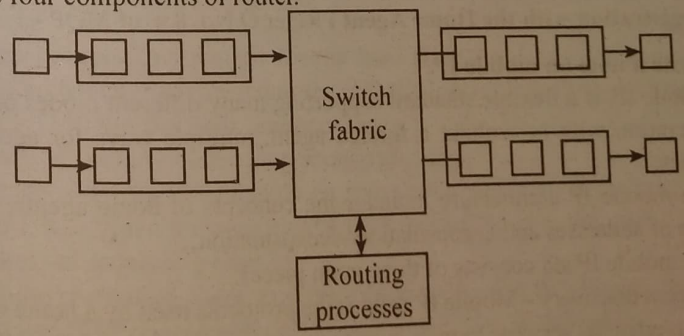
Running the also on the network in the fig.

Step	N^1	D(v), p(v)	D(W), p(W)	D(x), p(s)	D(y), p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	u x	2,u	4,x		2,x	∞
2	u x y	2,u	3,y			4,y
3	u x yv					4,y
4	u x yvw					4,y
5	u x yvwz					4,y

1. In the initialization step, currently known least cost paths from u to its directly attached neighbours v,x & w are initialized to 2,1 & 5 respectively. The costs to y & z are set to infinity became they are not directly connected to u.
2. In the second iteration, nodes v & y are found to have the least cost path (2) & we break the tie arbitrarily and add y to the set N^1 now contains u,x & y. The costs to the remaining nodes not yet in N^1 i.e., nodes v,w and z are updated via line 12 of the LS also, yielding the results shown in the third row.

b. What is routing? Explain the structure of a router. **(08 Marks)**

Ans. Router is a networking device that forwards data packets between computer networks. There are four components of router.



1. Input ports : It performs physical layer functions of terminating an incoming physical link to a router. It perform the data link layer functions needed to interperate with the data link layer functions at the remote side of the link. Multiple ports are gathered together on a single line card within a router.

2. Switching fabric connects the router's input ports to its output ports. Switching fabric is contained within the router - a network inside of a network router.

3. Output ports : An output port stores the packets that have been forwarded to it through the switching fabric and then transmits the packets on the outgoing link. The output port performs the reverse data link and physical layer functionality of the input port.

4. Routing processor - Executes the routing protocols maintains the routing info and forwarding tables and performs network management functions, within the router.

OR

6. a. Discuss the IPV6 packet format. **(05 Marks)**

Ans. Refer Q.No. 5.b. of MQP - 3

b. Elaborate the path attributes in BGP and steps to select the BGP routes. **(05 Marks)**

Ans. Refer Q.No. 6.b. of MQP - 1

c. List the broadcast routing algorithms. Explain any one of them. **(06 Marks)**

Ans. The broadcast routing algorithms are

1. Uncontrolled flooding
2. Controlled flooding
3. Spanning tree broadcast

For spanning tree broadcast routing also explanation refer 6(B) MQP 3.

Module - 4

7. a. Show the components of GSM 2G cellular network architecture with a diagram. **(07 Marks)**

Ans. Refer Q.No. 7.b. of MQP - 2

b. Illustrate the steps involved in mobile IP registration with home agent. **(05 Marks)**

Ans. Registration with the Home Agent : Refer Q.No. 8.a. of MQP - 1

c. Write a note on mobile IP. **(04 Marks)**

Ans. Mobile IP, is a flexible standard supporting many different modes of operation. Ex: Operation with or without a foreign agent, multiple ways for agents and mobile nodes.

The mobile IP architecture includes the concepts of home agents, foreign agents, case of addresses and encapsulation/ decapsulation.

The mobile IP std consists of three main pieces.

i. Agent discovery - Mobile IP defines the protocols used by a home or foreign agent to advertise its services to mobile nodes and protocols for mobile nodes to solicit the services of a foreign or home agent.

ii. Regitration with home agen - Mobile IP defines the protocols used by the mobile node and / or foreign agent to register and deregister COA with a mobile node's home agent.

iii. Indirect routing of datagrams - The std also defines the manner in which datagrams are forwarded to mobile nodes by a home agent, including rules for forwarding datagram, rules for handling error conditions.

OR

8. a. Define Handoff. Explain the steps accomplishing a handoff. **(07 Marks)**

Ans. Refer Q.No. 7.a. of MQP - 3

b. Bring out the mechanism of direct routing to mobile node in mobility management. **(06 Marks)**

Ans. Refer Q.No. 8.b. of MQP - 1

c. Compare the 4G LTE standard to 3G systems. (03 Marks)

Ans. 4G means long term evolution (LE) and is the dominant framework of cellular system.

3G cellular systems are required to provide telephone service as well as data communications at significantly higher speed than their 2G counterparts.

These are two major stds in 3G UMTS (universal mobile telecommunication service and CDMA 2000.

In 4G, in heterogenous environment access technology is switched from one to another automatically and transparently.

Module-5

9. a. Elaborate the features of streaming stored video. **(03 Marks)**

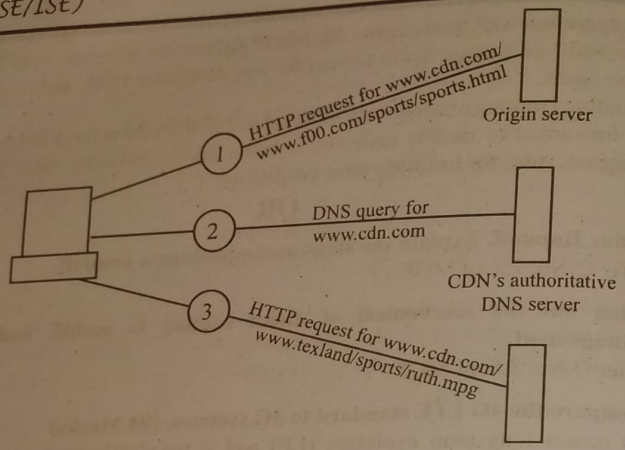
Ans. Refer Q.No. 10.b. of MQP - 2

b. With a neat diagram, explain the CDN operation. **(08 Marks)**

Ans. If the client can't come to the content, the content should be brought to the client. The CDN company typically places the CDN servers in data center. The CDN replicates its customers content in the CDM server. CDN pushes content provider's tagged object to its CDN servers.

When a browser requests a web page containing the image ruth. mpg following actions occur.

1. The browser sends its request for the base HTML object to the origin server, www.foo.com which sends the requested HTML objects to the browser. The browser parses the HTML file and finds reference to
http : //www.cdn.com/www.too.com/sports/ruth.mpg.
2. Browser then does a DNS lookup on www.cdn.com which is the host name for referenced URL. When the authoriative DNS server receives the query, it extracts the IP address of requesting browser.
3. DNS in the requesting client receives a DNS reply with IP address. The browser then sends its HTTP request to the CDN server with that IP address.



c. Summarize the limitations of Best-effort IP service. (05 Marks)

Ans. The limitations are

1. Packet loss - The UDP segment in encapsulated in an IP datagram. As the datagram wanders through the network, it passes through buffers in the routers in order to access outbound links. In this case, the IP datagram is discarded never to arrive at the receiving application.
2. End to end delay is the accumulation of transmission, processing and queuing delay in routers, propagation delays in links and system processing delays. Thus, packets that are delayed by more than the threshold are effectively lost.
3. Packet jitter - End to End delay is the random queuing delays in the routers. The time from when a packet is generated at the source until it is received at the receiver can fluctuate from packet to packet. This phenomenon is called jitter. Jitter can be removed by using sequence nos, timestamps and a payout delay.

OR

10. a. Explain the diffserv internet architecture. (05 Marks)

Ans. Refer Q.No. 10.b. of MQP - 3

b. Describe the leaky bucket policing mechanism. (06 Marks)

Ans. Refer Q.No. 10.a. of MQP - 1

c. Discuss the round-robin and waited fair queuing scheduling mechanism. (05 Marks)

Ans. Refer Q.No. 9.b. of MQP - 2

Fifth Semester B.E. Degree Examination, CBCS - June/July 2018
Computer Networks

Time: 3 hrs. **Max. Marks: 80**

Note : Answer any FIVE full questions, selecting ONE full question from each module.

Module - 1

1. a. What are the different types of transport services provided by internet.

Ans. Refer Q.no.1(b) of MQP - I. **(08 Marks)**

b. Compose logical note on proxy - server with suitable diagram. (08 Marks)

Ans. Refer Q.no.2(b) of Dec 17/ Jan 18.

OR

2. a. Discuss how files are distributed in peer-peer application

Ans. Let us consider a simple quantiative model for distributing a file to a fixed set of peers. The upload rate of the server's access in denoted by u_s, upload rate of ith link by u and download rate of ith peer access links by di. the size of the file to be distributed is F bits and no. of peers that want to obtain a copy of the file by N. Distribution time is the time it takes a copy of file to all N peers.

Following observations are made

i. At the beginning of file distribution, only the server has the file. To get this file into the community of peers, the server send each bit of the file atleast once access link. Minimum distribution time is atleast F/us.

ii. The peer with the lowest download rate cannot obtain all F bits of the file in less F/d_{min} seconds. Thus, minimum distribution is atleast F/d_{min}.

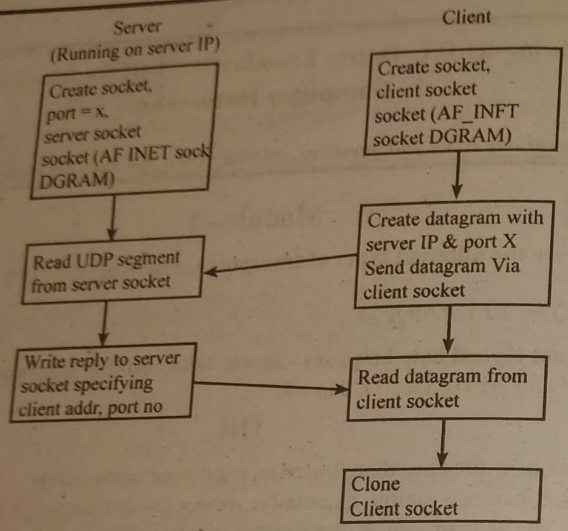
iii. The total upload capacity of the system as a whole is equal to the upload rate of server plus upload rates of each of the individual peers i.e., $u_{total} = u_s + u_1 + ... + u_N$. System must deliver (upload) F bits to each of the N peers, thus delivering a total of NF bits. This cannot be done at a rate faster than u_{total}. Thus, min distribution is also at least $NF/ (u_s + u_1 + ... + u_N)$

The minimum distribution time for P2P, denoted by D_{P2P}.

$$D_{P2P} \geq \max\left\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_s + \sum_{i=1}^{N} u_i}\right\}$$

b. Design network application using socket programming with UDP. (08 Marks)

Ans. When a socket is created, an identifier called port number is assigned.



Below is the code for client side of application

```
from socket import *;
ServerName = 'hostname'
Server Port = 12000
ClientSocket = Socket(Socket, AF_INET,Socket, SOCK_DGRAM)
message = raw _input (Input lowercase sentence:')
ClientSocket.Sendto (message(ServerName, Serverport))
modified message, Server Address=ClientSocket. recvfrom (2048)
print modified message
ClientSocket. Close( )
```

Here is the code for server side of application

```
from Socket import *
Serverport = 12000
ServerSocket = Socket (AF_INET, SOCK_DGRAM)
ServerSocket bind(' ' ,ServerPort)
print " The server is ready to receive"
While 1 :
message, ClientAddress = ServerSocket, recv from (2048)
modifie Message = message. upper( )
ServerSocket, Sendto (modifiedMessage,ClientAddress)
ClientSocket = Socket(Socket.AF_INFT,Socket.Sock_DGRAM)
```

This line creates the clients socket called clientsocket. First parameter indicates address family. AF_INFT indicates underlying network in IPv4.Second parameter indicates socket is of type

SOCK_DGRAM meaning it is a UDP socket raw_input( ) is a built in function in

python. When this command is executed, user at the client is prompted with words " Input data".

Module - 2

3. a. Describe the various fields of UDP segment. Explain how checksum is calculated. **(08 Marks)**

Ans. UDP segment structure is defined in RFC 768.

Source port #	Dest port #
Length	Checksum
Application data (message)	

Application data occupies the data field of UDP segment. Ex DNS data field contains either a query or response message. UDP header has four fields, each consisting of two bytes. The port # header has four fields, each consisting of two bytes. The port # allows the destination to the application data to the correct process running on the destination end system. Checksum is used by the receiving host to check whether errors have been introduced into the segment.

UDP Checksum :

UDP checksum provides error detection. Checksum is used to determine whether bits within the UDP segment have been altered as it moved from source to destination. UDP at the sender side performs 1's complement of the sum of all 16 bit words in segment, with anyoverflow encounter the sum being wrapped around.

Ex : Suppose following three 16 bit words are there

0110011001100000
0101010101010101
1000111100001100

Sum of the first two of these 16 bit words is

0110011001100000
0101010101010101
1011101110110101

Adding the third word to the above sum gives

1011101110110101
1000111100001100
0100101011000010

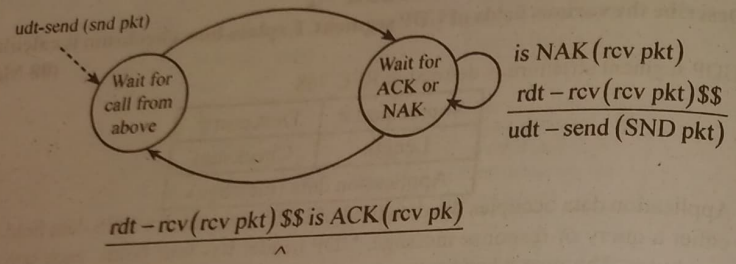
Last addition has over flow, which is wrapped around. At the receiver , all four 16 bit words are added including checksum. If no errors are introduced , then sum at receiver will be 1111111111111111. If one of the bits is 0, then errors has been introduced into the packet.

1's complement is obtained by converting all the 0's to 1's and converting an to 1's to 0's.

b. Design rdt 2.0 protocol. **(08 Marks)**

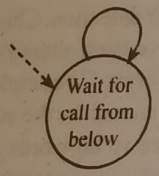
Ans. The send side of rdt 2.0 has two states. In the leftmost state, send side protocol is waiting for data to be paused down from upper layer. When the rdt _ send (data) event occurs, sender will create a pallet (snd pkt) containing data to be sent along with a packet checksum.

rdt_send(data)
Sndpkt = make_pkt(datachecksum)



a. rdl 2.0 sending side

rdt_rcv(rcvpkt) && corrupt(rvcpkt)
Sndpkt = make_pkt(NAK)
udt_send(sndpkt)



rdt_rcv(rcvpkt) && corrupt(rvcpkt)
Extract(rcvpkt, data)
deliver_data(data)

b. rdt 2.0 receiving side

Sndpkt = make_pkt (Ack)
udt_send(sndpkt)

rdt 2.0 - A protocol for a channel with bit errors.

When the sender is in the wait for ACK_NAK it cannot get more data from upper layer. Receiver side has rdt 2.0 has a single state. On packet arrival, receiver replies with either an ACK or NAK, depending on whether or not the received packet is corrupted. The notation rdt_rcv (rcvpkt) && corrupt (rcvpkt) corresponds to the event in which a packet is received and is found to be in error.

If an ACk packet is received rdt_rcv(rcvpkt) && in Ack(rcv/pkt) the sender known that the mostly recently transmitted packet has been received correctly & thus the protocol returns to the state of waiting for data from upper layer. The sender will not end a new piece of data until it is sure that the receiver has correctly received the rrent packet. Because of this behavioue, protocol rdt 2.0 are known as stop and it protocols.

OR

4. a. **With a neat sketch, explain TCP segment and its services.** **(08 Marks)**

Ans. Refer Qno.3(a) MQP - 2.

b. **Explain how connection is established and tear down in TCP.** **(08 Marks)**

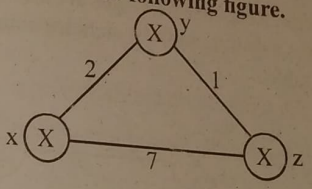
Ans. Refer Q.no.4(c) of MQP - 3.

Module - 3

5. a. **Draw Ipv6 datagram format, mention the significance of each field.** **(08 Marks)**

Ans. Refer Q.no.6(a) of Dec 17/ Jan 18.

b. **Apply distance vector algorithm for following figure.** **(08 Marks)**



Ans.

Node x table

Cost to

From	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

Cost to

From	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

Cost to

From	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

Node y table

Cost to

From	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

Cost to

From	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

Cost to

From	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

Node z table

Cost to

From	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

Cost to

From	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

Cost to

From	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

The leftmost column of fig displays three initial routing tables for each of the three nodes. The second and third rows in this table are the most recently received distance vectors from nodes y & z respectively. Because at initialization , node x has not

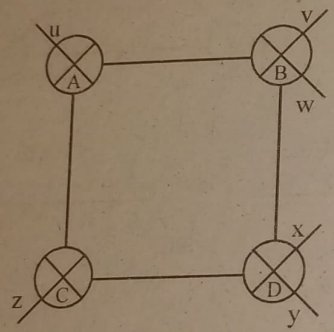
received anything from node y or 2, the entries in second and third row are initialized to infinity.
After initialization, each node send its distance vector to each of its two neighbors.
After receiving the updates, each node recomputes its own distance vector.

OR

6. a. **Illustrate routing info protocol (RIP) with suitable diagram.** **(08 Marks)**

Ans. Routing info protocol (RIP) is intra as routing in internet. RIP is a distance vector protocol. RIP uses hop i.e., no. of subnets traversed along the shortest path from source router to destination subnet, including destination subnet. In RIP, routing updates are exchanged between neighbors approximately every 30 seconds using RIP response message. Response messages are also known as RIP advertisements. Each router maintains a routing table, it includes both the routers distance vector and routers forwarding table.

Ex :



No of hops from sources A to various subnets

Destination	Hops
u	1
v	2
w	2
x	3
y	3
z	2

If a router does not hear from its neighborer every 180 seconds, that is considered to be n longer reachable i.e., either neighbor has died or connecting link has gone down. When this happens, RIP modifies the local routing table and then propagates this info by sending advertisements to its neighboring routers.

b. Explain spanning tree algorithm. **(08 Marks)**

Ans. Refer Q.no.6(b) of MQP - 3.

Module - 4

7. a. **Define cellular network. Give the overview of GSM cellular network architecture.**

Ans. Refer Q.no.7(b) of MQP - 2 **(08 Marks)**

b. **Explain the two different types of routing approaches to mobile node.**

Ans. Refer Q.no.8(b) of MQP - 1 and Refer Q.no.7(a) of MQP - 2. **(08 Marks)**

OR

8. a. **Explain the following concepts of mobile IP**
i. Agent discovery
ii. Registration with home agent **(08 Marks)**

Ans. i. Agent discovery
ii. Registration with home agent : Refer Q.no.8(a) of MQP - 1.

b. **Illustrate the steps involved when a base station does decide to hand-off mobile user.**

Ans. Refer Q.no.7(a) of QMP - 3. **(08 Marks)**

Module - 5

9. a. **Brief out three broad categories of multimedia network application.** **(08 Marks)**

Ans. Refer Q.no.9(a) of MQP - 3.

b. **Discuss the following :**
i. Adaptive streaming
ii. DASH **(08 Marks)**

Ans. i. Adaptive streaming

OR

10. a. **With a general format, explain the various fields of RTP.** **(08 Marks)**

Ans. RTP runs on UDP. Sending side encapsulates an media chunk within an RTP packet, then encapsulates the packet in UDP segment and then hands the segment to IP. The sending side proceeds each chunk of data with an RTP header. RTP header form RTP packet. At the receiver side, application receiver its RTP packet from socket.

Payload type	Sequence number	Timerstamp	Synchronization source identifier	Miscellaneous field

RTP header fields is shown above
The payload field is 7 bits long. For ex in audio stream payload type field is used to indicate the type of audio encoding.

- Sequence no field is 16 bits long. Sequence no increments by one for each RTP

packet sent. The receiver uses sequence no to detect packet loss and to restore packet sequence.

- Timestamp field is 32 bits long. It reflects the sampling instant of the first byte in the RTP data packet. The receiver can use timestamp to remove packet fitter. The timestamp clock continues to n cease at a constant rate even if the source is inactive.
- Synchronization source identifier (SSRC) : Field is 32 bits long . It identifies source of RTP stream. Each stream in an RTP session has a distance SSRC. SSRC is a number that the source assigns randomly when new stream is started.

b. Explain the working procedure of leaky bucket algorithm **(08 Marks)**

Ans. Refer Q.no. 1.(a) of MQP - 1.

Fifth Semester B.E. Degree Examination, CBCS - Dec 2018 / Jan 2019

Computer Networks

Time: 3 hrs. **Max. Marks:** 80

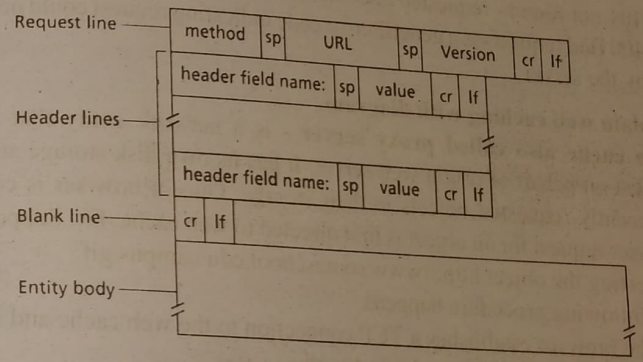
Note : Answer any FIVE full questions, selecting ONE full question from each module.

Module-1

1. a. Explain HTTP messages. **(08 Marks)**

Ans. There are two types of messages HTTP request message and restore messages. HTTP/1.0 allows three types of methods GET, POST and HEAD. The first line of an HTTP request message is called request time the subsequent times are called the header lines. The request line has three fields, method field, URL field and HTTP version field. Majority of HTTP request messages uses GET method. Header line specifies the user agent that is the browser type that is making request to the server.

Below is the general format of a request message

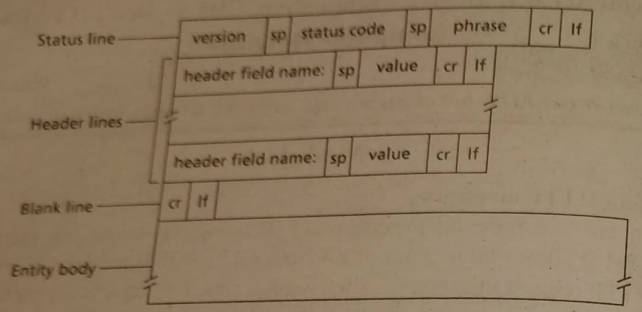


HTTP1.1 allows for additional methods PUT and DELETE. The DELETE method allow a user or an application to delete on a web server.

HTTP Response Message

It has three sections - an initial status line, sin header lines and entity body. The entity body contains the required object. Status line has three fields - protocol version, status code and corresponding status message. Server uses connection : close header line to tell client is going to close the connection. Data header line indicates time and date when HTTP response was created and sent by the server. Last-modified indicates date and time when the object was created or last modified. Content length indicates the number of bytes in the object being sent. Content type indicates the object in the entity body is HTML text.

Below is the form at of response message.



Some common status code and associated phrases include

- 2000k - request succeeded and the info is returned in the response.
- 404 not found - requested document does not exist on the server.
- 400 Bad request - is a genetic error code indicating request could not be understood by the server.

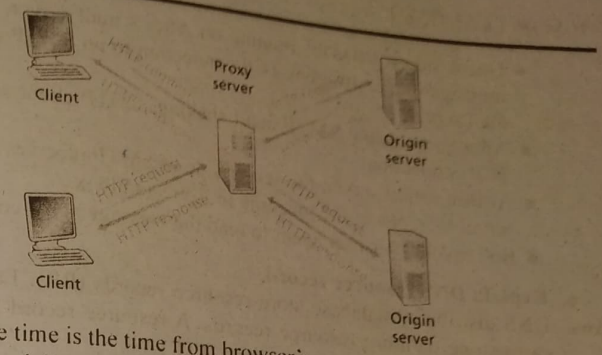
b. Explain web caching with diagram. **(08 Marks)**

Ans. Web cache also called proxy server - is a network entity that satisfies HTTP request on behalf of origin web server. It has its own disk storage and keeps copies of recently requested objects in their storage. Once a browser is configured, each browser request for an object is first directed to web cache. Ex: Suppose a browser is requesting the object http://www.someschool.edu/campus.gif.

The following procedure happens

- The browser establishes a TCP connection to the web cache and sends an HTTP request for the object to the web cache.
- The web cache checks to see if it has a copy of the object stored locally. If it has, it forwards the object within an HTTP response message to client browser.
- If it does not have, web cache opens TCP connection to the origin server i.e, www.school.edu. Web cache then sends an HTTP request for the object into the TCP connection. After receiving their request, origin server sends the object within an HTTP response to web cache.
- When the web cache receives the object, it stores a copy in the local storage and few as a copy.

Cache is both a server and a client at same time.



The total response time is the time from browser's request of an object is the sum of LAN delay, access delay and Internet delay.

OR

2. a. Explain FTP with its commands and replies. **(08 Marks)**

Ans. In FTP, the commands from client to server and replies from server to client are sent across control connection in 7 bit ASCII format. Each command consists of four upper case ASCII characters with optional arguments. Some of the commands are shown below.

- USER Username : Used to send user identification to the server.
- PASS Password : Used to send the password to the server.
- List : Used to ask the server to send back a list of all files in the current remote directory.
- RETR filename : Used to retrieve file from current directory.
- Stor file name : used to store a file into the current directory of remote host.

There is one to one correspondence between the user issued and FTP across control connection. Each common and is followed by a reply sent from server to client. The replies are three digit numbers with an optional message following number.

Some replies with the possible messages are shown below.

- 331 - User name ok, password required
- 125 - Data connection already open, transfer starting
- 425 - Can't open data connection
- 452 - Error writing file.

b. Explain SMTP. **(04 Marks)**

Ans. Simple mail transfer protocol(SMTP) transfers message from sender's mail to recipient's mail server. It uses 7-bit ASCII. Suppose Alice wants to send Bob a simple ASCII message following steps happens.

- Alice's invokes her user agent for e-mail, provides Bob's e-mail address, compose a message and instructs user agent to send the message.
- Alice's user agent sends the message to her mail server, where it is placed in a message queue.

- Client side of SMTP, running on Alice's mail server, sees the message in the message queue. It opens TCP connection to an SMTP server, running on Bob's mail server.
- After some initial SMTP hand shaking; SMTP client sends Alice message into TCP connection.
- At Bob's mail server, the server side of SMTP receives the message. Bob's mail server then places the message in Bob's mailbox.
- Bob invokes his user agent to read the message at his convenience.

c. Explain DNS resource record. **(04 Marks)**

Ans. DNS distributed database store resource records (RR). Each DNS reply message carries one or more resource records. A resource record is four types containing following fields (Name, Value, Type, TTL)

TTC is the time to live of resource record. It determines when a resource should be removed from cache. The remaining of Name and value depend on Type:

If type = A, then Name is a host name and value is the IP address for the host name.

If type = NS, then name is domain and value is the host name of an authoritative DNS server.

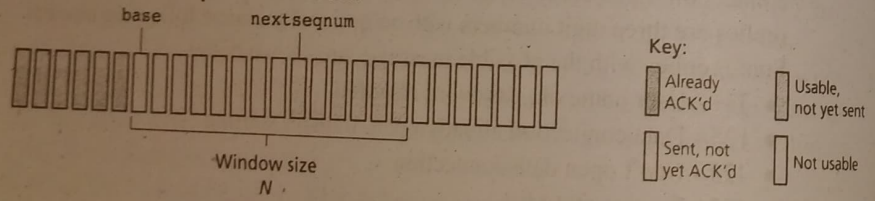
If type = Name , then value is a canonical host name for the alias host name Name.

If type = MX, then value is canonical name of a mail server that has an alias hast name Name.

Module-2

3. a. Explain Sender's view of sequence numbers and its operation in Goback N protocol. **(08 Marks)**

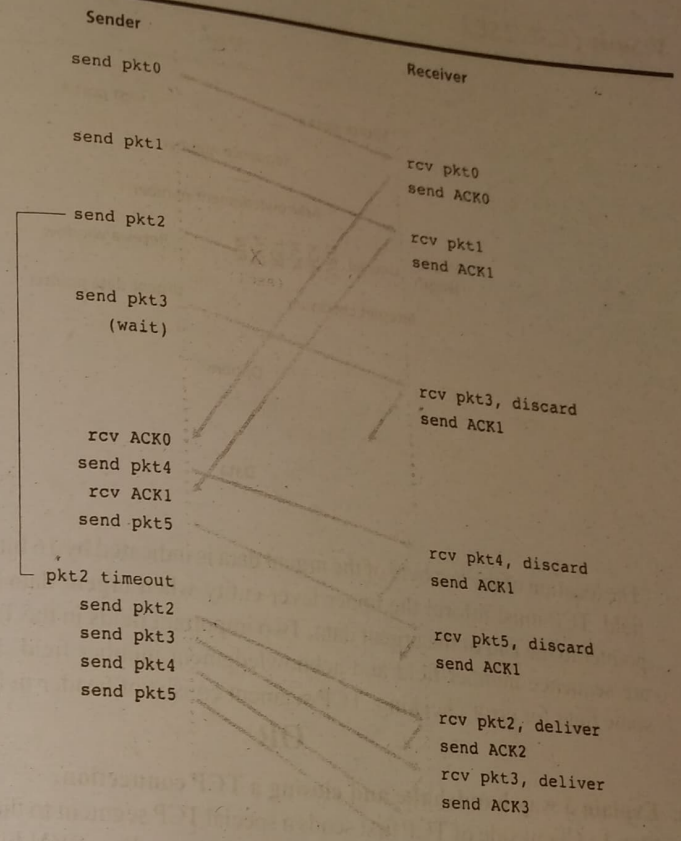
Ans. In Go - back - N (GBN) protocol, the sender is allowed to transmit multiple packets (when available) without waiting for an acknowledgment. It cannot have more than maximum value N of unacknowledged packets in the pipeline. Figure shown the sender's view os sequence numbers.



The range of permissible sequence numbers for transmitted but not yet acknowledged packets are viewed as window size N over the range of sequence numbers. As the protocol operates the window slides forward over the sequence number space.

In GBN protocol, receiver discards out of order packets. Suppose packet n is expected, n+1 packet arrives, because the data must be delivered in order, the receiver buffers n+1 packet and then deliver the packet to upper layer after it has received packet N.

Figure shows the operation of GBN protocol for the case of four packets.

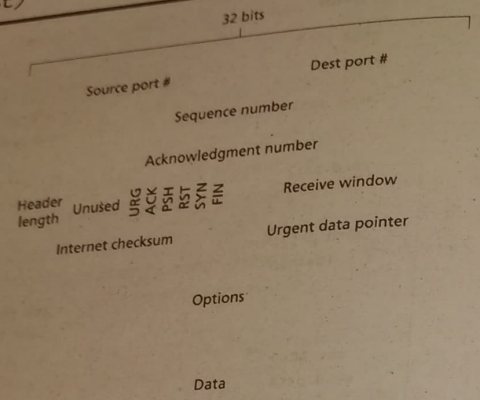


A packet's sequence number is carried in a fixed length field in the packet header. If K is the number of bits in the packet sequence number field, range of sequence numbers is $(0.2^k - 1)$

b. Draw TCP segment structure and explain. **(08 Marks)**

Ans. TCP structure includes source and destination post numbers used for multiplexing and de-multiplening.

- 32 bit sequence number field and 32 bit acknowledgment number field are used by TCP sender and receiver in implementing a reliable data transfer service.
- 16bit receive window field is used for flow control.
- 4 bit header length specifies the length of TCP header in 32 bit words.
- Optional and variable length options field is used when a sender and a receiver negotiate maximum segment size(MSS)
- Flag field contains 6 bits. ACK but is used to indicate that the value carried in the acknowledgement field is valid. RST, SYN, FW bits are used for connection setup and tear down. Setting PSH bit indicates receiver should pass data to the upper layer immediately. URG bit is used to indicate that there is data in the segment that the sending side upper layer entity has marked urgent.



The location of the last byte of the urgent data is indicated by 16 bit urgent data pointer field. TCP must inform the upper layer entity when urgent data exists and pass it a pointer to the end of the urgent data. Two important fields in the TCP segment header are sequence number field and acknowledgment number field. It includes a check some field for error checking. TCP segment consist of header fields and data field.

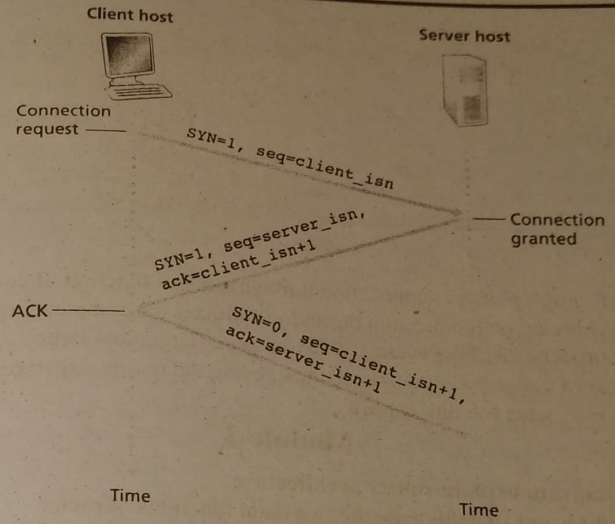
OR

4. a. Explain 3 way handshake and closing a TCP connection. **(08 Marks)**

Ans. **Step 1 :** Client side of TCP first sends a special TCP segment to the server - side TCP. This special segment contains no application layer data. SYN bit is set to 1. Client randomly chooses an initial sequence number and puts this number in the sequence number field of initial TCP SYN segment. This segment is encapsulated within an IP datagram and sent to the server.

Step 2 : Once the IP datagram containing TCP SYN segment arise at the server host, server extracts the TCP SYN segment from datagram, allocates TCP buffers and variables to the connection and sends a connection granted segment to the client TCP. It contains three important piece of info. First SYN bit is set to 1, second acknowledgement field of TCP segment header is set to client_isn+1. Finally, server chooses its own initial sequence number (server_isn). The connection granted segment is also called referred to as SYNACK segment.

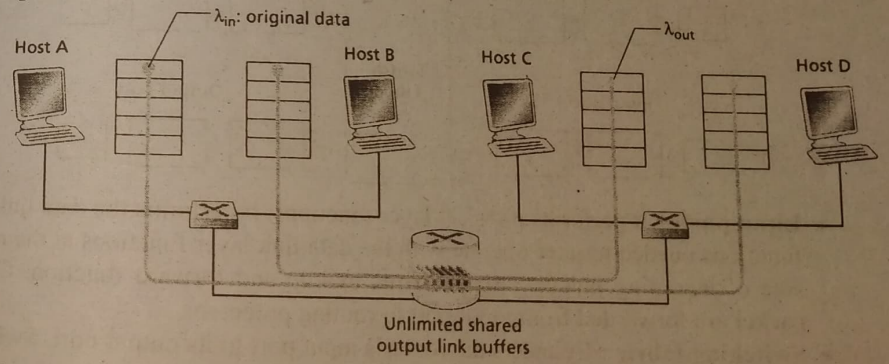
Step 3 : UPon receiving SYNACK segment, client also allocates buffers and variables to the connection. The client host then sends the server another segment. This last segment acknowledges servers connection granted segment. The SYN bit is set to 0, since the connection is established.



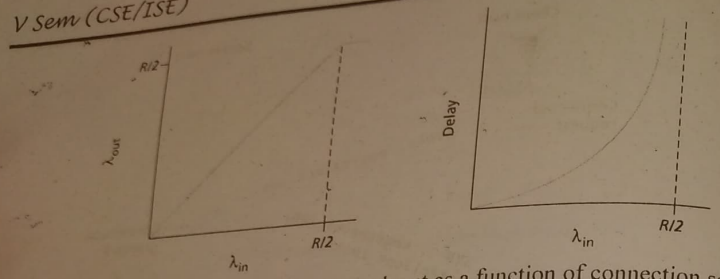
Once these three steps are completed , client and server host can send segments containing data to each other.

b. Explain the causes and costs of congestion. **(08 Marks)**

Ans. Causes and cost of congestion can be explained by taking any one of the scenario's Scenario : To senders, a router with infinite buffers two host (A and B) each have a connection that shares a single hop between source and destination as shown in figure.



Application in host A sends data into the connection at an average rate of λ in bytes/second. Each unit of data is sent only once into the socket. Data is encapsulated and sent no error recovery is included. Host A offers traffic to the router is λ in bytes/second. Host B also operate in similar manner. Packets from host A and B ran through a router and share outgoing link of capacity R. The router has buffers that allows it to store in coming packets when packet arrival exceeds outgoing link.

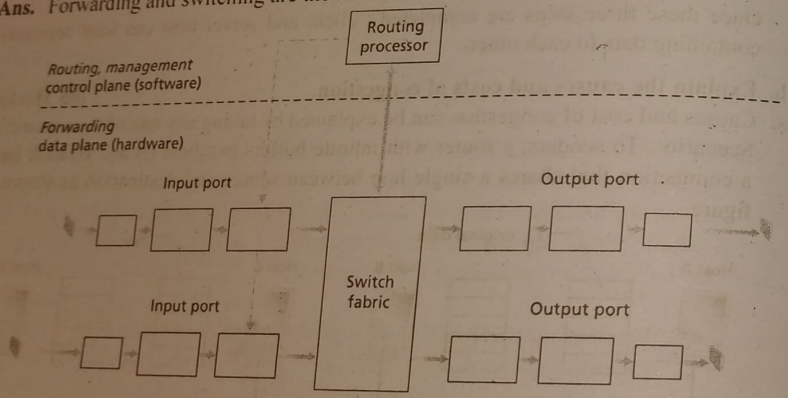


The left graph plots per connection through out as a function of connection sending rate. Achieving per connection through out utilizes the complete link. As the sending rate approaches R/2 the average delay becomes larger and larger.
The cost of congested network/- large queuing delays are experienced as the packet arrival rate nears the link capacity.

Module-3

5. a. With diagram explain router architecture. **(08 Marks)**

Ans. Forwarding and switching are the two main functions of router.



- **Input ports :** It performs physical layer functions. It performs the data link layer functions needed to inter operate with the data link layer functions at the remote side of incoming link. It also performs lookup and forward function. Control packet are forwarded from input port to routing processor.
- **Switching fabric :** Connects the router's input port to its output port. Switching fabric is completely contained within the router.
- **Output ports :** Stores the packets that have been forwarded to it through the switching fabric and then transmits the packets on the outgoing link. It performs the reverse data link layer function of the input port. When the link is bidirectional. an output port to the link will be paired with input port on same line card.

- **Routing processor :** Executes the routing protocol maintains the routing information and forwarding tables and performs network management functions within the router.

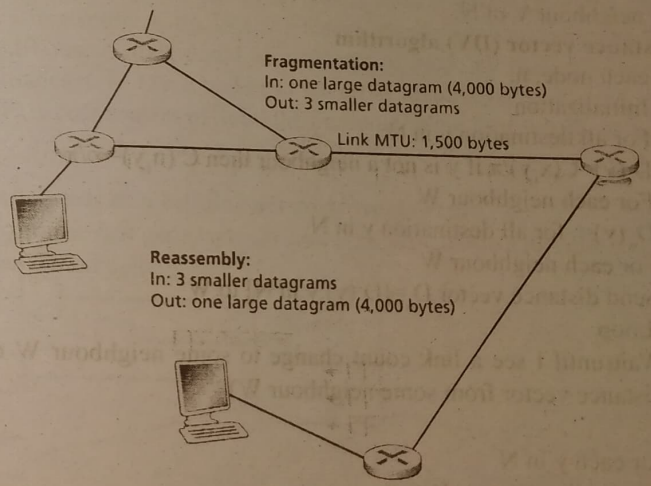
Multiple ports are often gathered together on a single line card within a router.

b. Explain IP fragmentation. **(08 Marks)**

Ans. The maximum amount of data a link layer frame can carry is called maximum transmission unit (MTU). Because each IP data gram is encapsulated within link layer format, transport from one router to next router places a limit on the length of IP packet. The solution is to fragment the data in the IP data gram into two or more smaller IP data gram, then send these smaller data gram over outgoing link. Each of these referred as fragment.

Fragment needs to be reassemble at transport layer before they reach destination. If some data gram are fragments, it needs to determined when it as received last fragment. To allow reassembling identification, flag and fragment offset fields are present in IP data gram. When a data gram is created, sending host stamps the data gram with an identification number and source and destination address. When a router needs to fragment the data gram, each resulting data gram is stamped with source address and destination address.

Example : A data gram of 4000 bytes arrives at a router and must be forwarded to a link with an MTU of 1.500 bytes. This implies that 3.980 bytes in the original data gram can be allocated to three separate fragments.



Fragment	Bytes	ID	Offset	Flag
1st	1480bytes in data field of IP datagram	777	Offset=0 (data should be inserted at beginning)	Flag=1 (there are more fragments)
2nd	1480bytes of data	777	Offset=185	Flag=1
3rd	1020bytes (3980-1480-1480) of data	771	Offset=370	Flag=0 (no more fragments)

OR

6. a. Explain distance vector algorithm. **(08 Marks)**

Ans. Distance vector algorithm is inter active, asynchronous and distributed. Each node receives some information from one or more of its directly attached neighbours. It is interactive in that this process continues on until no information is exchanged between neighbours. It is asynchronous in that it does not receive all of the nodes to operate in lock setup with each other.

Each node X begins with $D_x(Y)$, an estimate of the cost of the least - cost path from itself to node y, for all nodes in N. Let $D_x=(D_x(y):y \text{ in } N)$ be the node x's distance vector. Each node x maintains following routing data.

- For each neighbour V, cost C(x,v) from x to directly attached neighbour V.
- Node x's distance vector that is $D_x=(D_x(y):y \text{ in } N)$, containing x's estimate of its cost to all destinations y in N.
- The distance vector of each of its neighbour that is $D_v=(D_v(y):y \text{ in } N)$ for each neighbour V of N.

Distance vector (DV) algorithm

At each node, n:

- Initialization
- For all destination y in N:
- $D_n(y)=C(x,y)/\pi$ if y is not a neighbour then C $(n,y)=\infty/\pi$
- For each neighbour W
- $D_w(y)=$ for all destination y in N
- For each neighbour W
- Send distance vector $D_n=[D_n(y):y \text{ in } N]$ to W
- Loop
- Wait(until I see a link count change to some neighbour W or until I receive a distance vector from some neighbour W)
-
- For each y in N
- $D_n(y)=\min_x(C(n,v)+D_x(y))$
-
- If $D_x(y)$ changed for any destination y
- Send distance vector $D_x=[D_x(y)=y \text{ in } N]$ to all neighbour
-
- Forever

b. Explain 4 types of hierarchical OSPF routers. **(04 Marks)**

Ans. An OSPF is configured into areas. Within each area one more border router are responsible for routing. Four types of OSPF router are

- Internal routers : are in non-backbone areas and perform only intra AS routing.
- Area border routers - These routers belong to both an area and the backbone.
- Backbone routers (non border routers) : Perform routing within the backbone. Within a non-backbone area, internal router learn of the existence of routers to the other areas from information broadcast within the area by its backbone routers.
- Boundary routers : Exchanges routing information with routers belonging to other autonomous system. This router for example use BGP to perform inter as routing. It is through such a boundary router that other routers learn about paths to external networks.

Exactly one OSPF area in the AS is configured to be the backbone area.

c. Compare link state with distance vector algorithm. **(04 Marks)**

Ans. **Message complexity :** LS requires each node to know the cost of each link in the network. Whenever a link cost changes, new link cost must be sent to all nodes. BV algorithm requires messages exchanges between directly connected neighbours at each iteration. DV will propagate results of the changed link cost only if the new link cost result is changed.

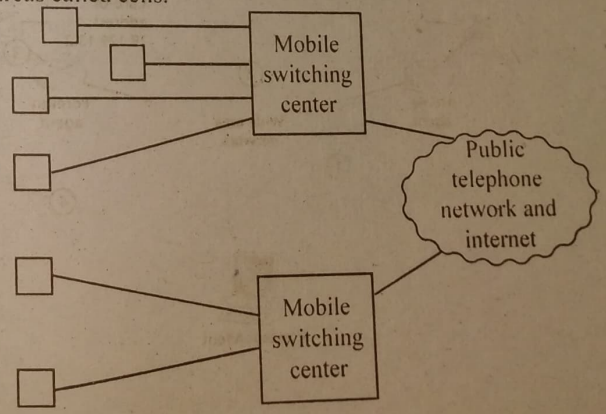
Speed of Convergence : Implementation of LS is an O (N^2) algorithm requiring O(NI/EI) messages. DV algorithm converges slowly and can have routing loops while the algorithm in is converging. DV refers from count to infinity problem.

Robustness : If a router fails, in LS a router broadcast an incorrect cost for one of its attached links. A node could also corrupt or drop any packet it received as a part of LS broadcast. In DV a node can advertise incorrect least paths to any or all destination. This cost routers to flood the malfunctioning router with traffic.

Module-4

7. a. Explain components of a cellular network architecture. **(08 Marks)**

Ans. Cellular refers to the fact geographical area is portioned into a number of geographic coverage areas called cells.



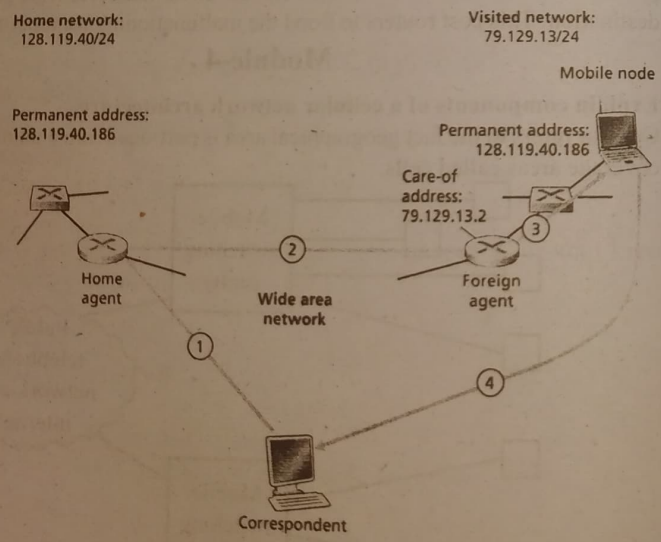
Each base station is connected to wide area network such as public switch telephone network (PSTN) or directly to internet via wired infrastructure. Each base station is connected to mobile switching centre (MSC) which manages call establishment and tear down to and from mobile users. An MSC contains much of functionality, calls need to share the portion of the radio spectrum that is allocated to cellular service. Two approaches are used - combination of frequently division multiplexing (FDM) and time division multiplexing (TDM)

Code division multiple access does not partition in frequently or in time. All users share the same radio frequency at same time. Each user in a cell is allocated a distinct sequence of bits called chipping sequence. When using FDM/TDM system, the receivers are sensitive to interference from other signals in same frequency band. A given frequency can be reused in FDM/TDM system only in cells that are located far to avoid such interference. Such frequency reused is major concern when designing CDMA systems.

b. Explain direct routing of a mobile node. **(08 Marks)**

Ans. Direct routing overcomes inefficiency of indirect routing, a correspondent agent in the correspondent's network first learns COA of mobile node. This is done by having correspondent agent query the home agent. mobile node has an up-to-date value for its COA registered with home agent. It introduces two challenges

- Mobile-users location protocol is needed for the correspondent agent to query home agent to obtain mobile node's COA.
- When the mobile node moves from one foreign network to another home agent is queried for the COA by correspondent agent only once at the beginning of session.

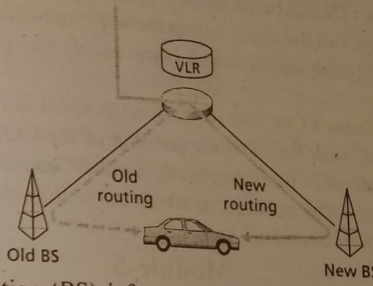


If the data is being forwarded to the mobile node in the foreign network, then foreign agent is identify (step1) as anchor foreign agent. When the mobile node moves to new foreign network (step2) the mobile node registers with new foreign agent (step3) and the new foreign agent provides the anchor foreign agent with mobile node's new COA (step4). When anchor foreign agent receives an encapsulated datagram for a departed mobile node it can then re-encapsulate the datagram and forward it to mobile node

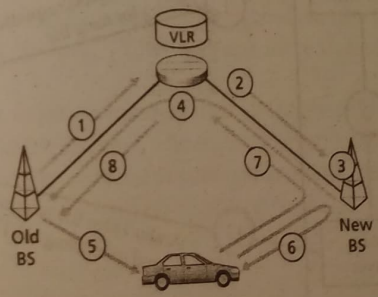
OR

8. a. Explain steps of handoff a mobile user. **(08 Marks)**

Ans. A hand of occurs when a mobile station changes its association from one base station to another base station.



- The old base station (BS) informs the visited MSC that a hand off is to be performed and the BS to which the mobile is to be handed off.
- Visited MSC initiates path setup to the new BS that a handoff is able to occur.
- The new BS allocates and activates a radio channel for use by the mobile.
- The new BS signal base to the visited MSC and the old BS that the visited MSC to new BS path that is established and the mobile needs to be handoff.
- The mobile is informed that it should perform a handoff..
- The mobile and the new BS exchange one or more messages to fully activate the new channel in BS.
- The mobile then sends a handoff complete message to the new BS which is forwarded upto the visited MSC. The visited MSC then reroutes the on-going call to the mobile via the new BS.
- The resources allocated along the path to the old BS are then released.



b. Explain HLR, VLR, home address, care-of-address. **(08 Marks)**

Ans. **HLR :** The home network maintains a database known as the home location register (HLR), which contains permanent cell phone number and subscriber profile info for each of its subscriber. HLR also contains info about the current location of these subscribers. HLR contains info to obtain an address in the visited network to which a call to the mobile user should be routed.

VLR : The visited network maintains a database known as the visitor location register (VLR). It contains an entry for each mobile user that is currently in the position of the network served by VLR. VLR entries come and go as mobile users enter and leave the network. VLR is co-located with the mobile switching centre(MSC).

Home Address : IN a network setting, permanent home of a mobile node is known as home network and the entity within the home network that performs the mobility management function and is called home agent. The permanent address is called home address.

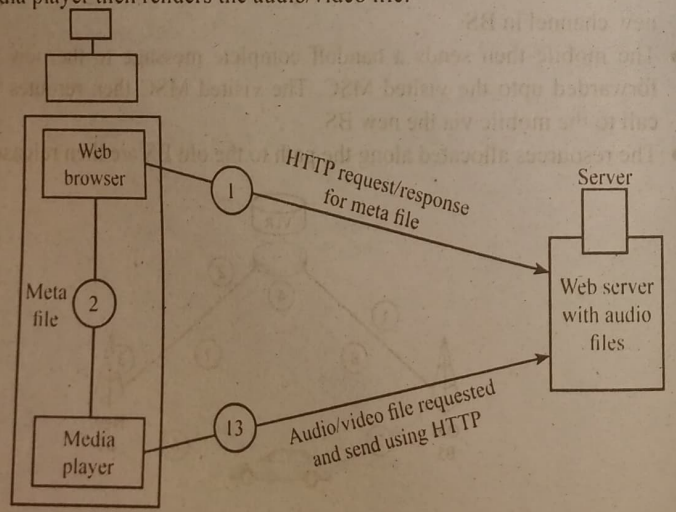
Case - of - Address : One role of the foreign agent is t create care-of -address for the mobile node, with the network partition of the COA matching that of foreign network. The address of foreign network is called Care - of - address. It is also known as foreign address. The network in which the mobile node is currently residing is known as foreign or united network.

Module-5

9. a. With diagram, explain naive architecture for audio/video streaming. (08 Marks)

Ans. In the naive architecture

- The browser process establishes TCP connection with web server and requests audio/video file with an HTTP request message.
- Web server sends the audio/video file to the browser in an HTTP response message.
- Content type header line in HTTP response message indicates a specific audio/ video encoding.
- Media player then renders the audio/video file.



- The user clicks on a hyperlink for audio/video file.
- Hyperlink does not point directly to audio/video file but instead into a meta file. It contains URL of actual audio/video file.
- Client browser examines the content type header line of the response message. launches associated media player and passes entire body of the response message.
- Media player after TCP connection directly with the HTTP server. The media player sends an HTTP request message for the audio/video file into the TCP connection.
- The audio/video file is sent within an HTTP response message to the media player. The media player streams out the audio/video file.

b. Explain audio compression in internet. **(08 Marks)**

Ans. Before audio and video can be transmitted over a computer network it must be digitised and compressed. All transmitted info must be represented as a sequence of bits.

A continuously varying analog audio signal is converted to digital signal as follows

- The analog signal is first sampled at some fixed rate, for example at 8000 samples per second. The value of each sample is an arbitrary real number.
- Each of the samples is rounded to one of a finite number of values. This operation is referred to as Quantization. The number of finite values called Quantization values - is power of two for En 256 Quantization values.
- Each Quantization value is represented by a fixed number of bits. EX : If there are 256 Quantization values then each value is represented by 1 byte. Each of the samples is converted into its bit representation. Bit representation of all samples are concatenated together to form digital representation of signal.

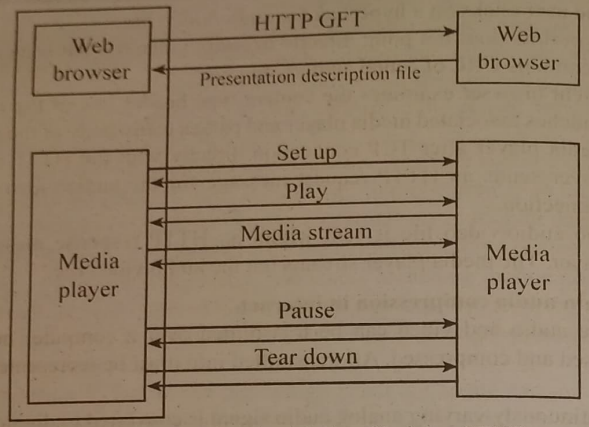
The basic encoding technique is called pulse code modulation (PCM) with a sampling rate of 8000 samples per second.

OR

10. a. With diagram, explain interaction between client and server using RTSP. **(08 Marks)**

Ans. Real Time Streaming Protocol(RTSP) allows a media player to control the transmission of media stream. Control actions include pause/resume, repositioning of playback, fast-forward and rewind RTSP is an out band protocol. RTSP messages are sent out of band where as the media stream, whose packet structure is not defined by RTSP is considered "in-band". RTSP message use port number 544 from media stream.

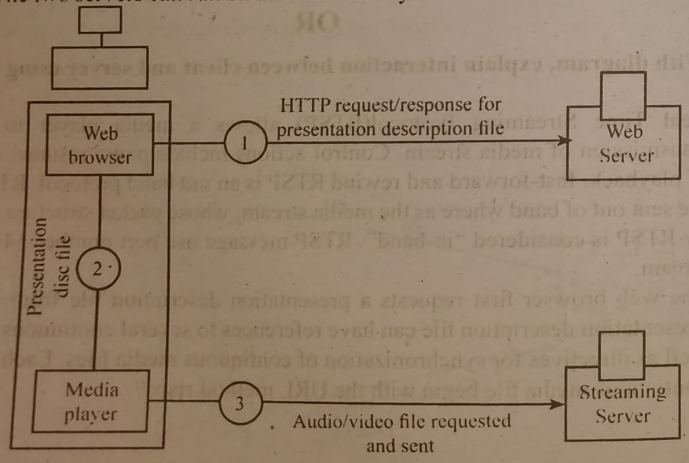
The web browser first requests a presentation description file from a web server. Presentation description file can have references to several continuous media files as well as directives for synchronization of continuous media files. Each reference to a continuous media file begin with the URL method rtsp://.



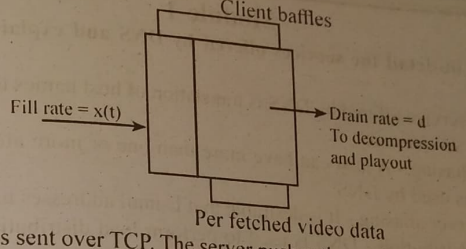
Web server encapsulates presentation description file in an HTTP response message and sends message to the browser. When the browser receives message, it invokes media player based on content type field of the message. The player sends an RTSP SETUP RTSP play request and server responds with ok message. Later the RTSP sends an RTSP PAUSE request and server responds with ok message. Finally tear down phase happens and connection is terminated.

b. Explain how streaming from streaming server to a media player is done.
(08 Marks)

Ans. The architecture requires two servers. One server, the HTTP server, server web pages (including meta files). The second server, the streaming server serves the audio/video files. The two servers can run on the same end system or on two distinct end systems.



- The audio/video is sent over UDP at a constant rate equal to the drain rate at the receiver.
- This is same as the first option, but the media player delays playout for two to five seconds in order to eliminate network induced jitter. Client accomplishes this task by placing compressed media it received from the network into a client buffer as shown in figure.



- The media is sent over TCP. The server pushes the media file into the TCP socket as quickly it can. Client reads from the TCP socket as quickly as it can and places the compressed video into the media player buffer.

Fifth Semester B.E. Degree Examination, CBCS - June / July 2019

Computer Networks

Time: 3 hrs. **Max. Marks:** 80

Note : *Answer any FIVE full questions, selecting ONE full question from each module.*

Module-1

1. a. Describe in detail the services offered by DNS and explain the DNS message format. **(08 Marks)**

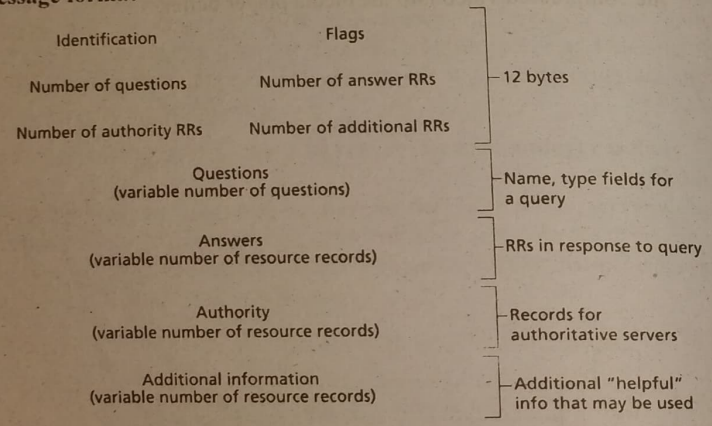
Ans. The main service offered by DNS is translation of host names to IP addresses. Other additional services are

(1) Host aliasing - A host can have more than one or more alias names. Canonical hostname is used by DNS.

(2) Mail server aliasing - It is desirable that E-mail addresses must be mnemonic.

(3) Load distribution - DNS is used to perform load distribution among replicated server.

DNS message format



- The first 12 bytes is the header section, First field is 16 bit no that identifies query. Identifies is copied into reply message to a query.
- The question section contains info about the query that is being made. This section includes a name field that contains name that is being queried and a type field that indicates the type of question being asked.
- In a reply from DNS server, answer section contains the resource records for the name that was originally queried.
- The authority section contains records of other authoritative servers.
- The additional section contains other helpful records. For ex, answer field in a reply to an MX query contains a resource record providing canonical ostname of a mail server. Additional section contains a Type A record providing IP address for Canonical hostname of mail server.

b. Illustrate the basic operation of SMTP and FTP.

Ans. Refer Q.No. 2.a. and 2.b. of Dec 18 / Jan 19 **(08 Marks)**

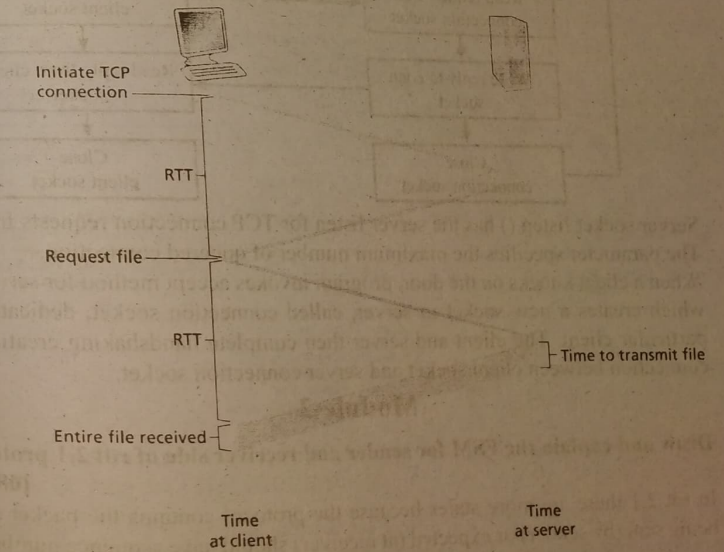
OR

2. a. Explain the persistent and non-persistent connection of HTTP. **(08 Marks)**

Ans. **Non-persistent connection :-** Web page needs to be transferred from server to client, the following steps take place

1. The HTTP client process initiates a TCP connection to the server www.school.edu on port number 80.
2. The HTTP client sends an HTTP request message to the server via its socket associaed with TCP connection. Request message includes the path name / someDepartment/home.index.
3. HTTP server process receives the request message via its socket associated with connection retrieves the object.
4. HTTP server process tells TCP to close the TCP connection (But TCP doen't actually terminate the connection until it knows for sure that the client has received response message intact)
5. HTTP client receives the response message. TCP connection terminates. Message indicates an encapsulated object is an HTML file. The client extracts the file from the response message, examines the HTML file, and finds references to JPEG objects.
6. The first four steps are then repeated for each of the referenced JPEG objects.

Round trip time is the time it takes for a small packet to travel from client to server and then back to client



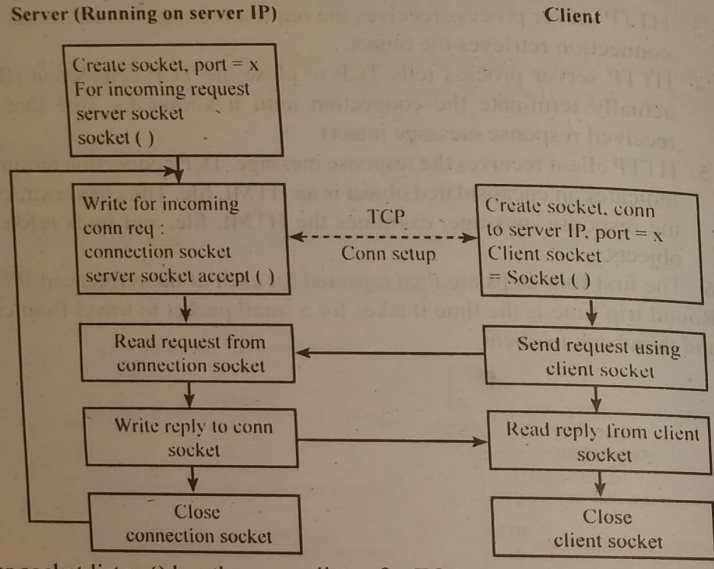
With persistent connections, server leaves the TCP connection open often sending a response. Sub request and responses between the same client and server can be sent over same connection.

There are two versions of persistent connection - without pipe lining and with pipe lining. For version without pipe lining client issues a new request only when the previous response has been received.

Default mode of HTTP uses persistent connections with pipe lining. With pipe lining, HTTP client issues a request as soon as it encounter a reference. Client can make back to back requests for referenced objects.

b. Define a socket. Describe the socket programming using TCP. (08 Marks)

Ans.



Server socket listen () has the server listen for TCP connection requests from client. The parameter specifies the maximum number of queered connection.

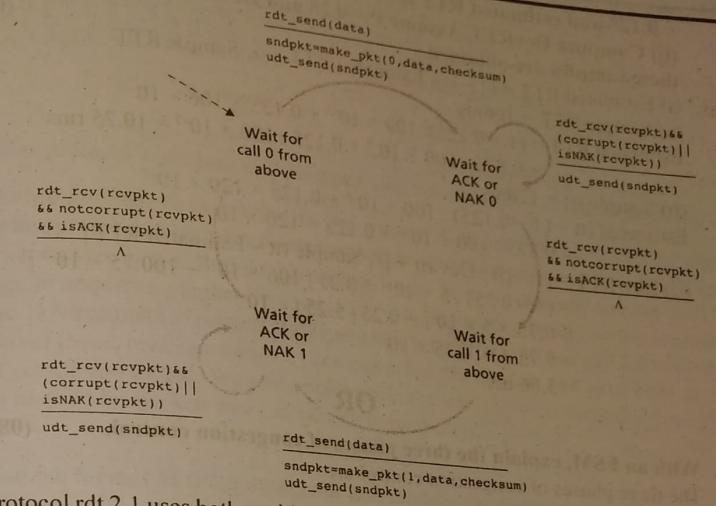
When a client knocks on the door, program invokes accept method for server socket, which creates a new socket in server, called connection socket, dedicated to this particular client. The client and server then complete handshaking creating a TCP connection between client socket and server connection socket.

Module-2

3. a. Draw and explain the FSM for sender and receiver side of rdt 2.1 protocol. (08 Marks)

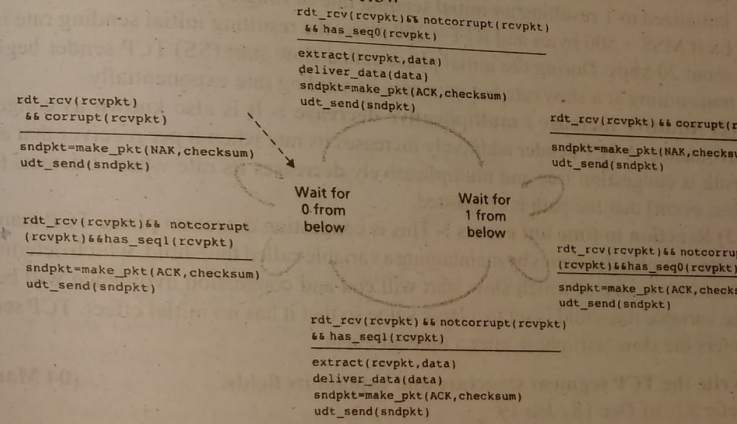
Ans. In rdt 2.1 there are more states because the protocol contains the packet currently being sent (by sender) or expected (at receiver) should have sequence number 0 or 1.

FSM of sender is shown below.



Protocol rdt 2.1 uses both positive and negative acknowledgements from the receiver to the sender. When an out of order packet is received, the receiver sends a positive acknowledgment for the packet is has received.

FSM of rdt 2.1 receives is shown below



b. Elaborate the three-way handshaking procedure used in TCP. (04 Mark

Ans. Refer Q.No. 4.a. of Dec 18 / Jan 19

c. Suppose that 2 measured sample RTT values are 106 ms and 120 ms. Compu

(i) Estimated RTT after each of these sample RTT value is obtained, Assume

= 0.125 and estimated RTT is 100 ms just before first of the sample obtained. (ii) Compute DevRTT, Assume P = 0.25 and DevRTT was 5 msec before first of these samples are obtained. (04 Marks)

Ans. (i) Estimated RTT = $(1-\alpha)$. Estimated RTT _ α. Sample RTT

(a) Sample RTT = 106ms

Estimated RTT = $(1-0.125).\ 100\times10^{-3} + 0.125\times106\times10^{-3}$

$= 0.875\ .\ 100\times10^{-3} + 0.125\ .\ 106\times10^{-3} = 10.75$ rms

(b) Sample rtt = 120ms

Estimated rtt = $(1-0.125)\ .\ 100\times10^{-3} + 0.125\times120\times10^{-3}$

$= 0.875\ .\ 100\times10^{-3} + 0.125\times120\times10^{-3}$

(ii) Dev RTT = $(1-\beta)$. Dev rtt + β | Sample rtt – Estimated RTT |

$= (1-0.25)\ .\ 5\times10^{-3} + 0.25\ |\ 106\times10^{-3} - 100.75\times10^{-3}\ |$

$= 0.75\times5\times10^{-3} + 0.25\ |\ 5.25\ |\times10^{-3}$

$= 3.75 + 0.3125\times10^{-3}$

$= 5.06$ ms

OR

4. a. With an FSM, explain the three phases of congestion control. (08 Marks)

Ans. The three phases of congestion control are

(1) Slow start phase

(2) Congestion avoidance phase

(3) Congestion detection phase

(1) Slow start phase :- When a TCP connection begins, value of congwin is initialized to 1 resulting an initial sending rate of roughly MSS/RTT.

Ex if MSS = 500 bytes and RTT = 200 msec, the resulting initial sending rate is only about 20 kbps. During the initial phase called slow start (SS) TCP sender begins by transmitting at a slow rate but increases its sending rate exponentially.

(2) Additive increase / multiplicative decrease :- It is also known as congestion avoidance. A TCP sender additively increases its rate when it per receiver that end of path is congestion free and multiplicatively decreases its rate when it detects (via a loss event) that the path is congested.

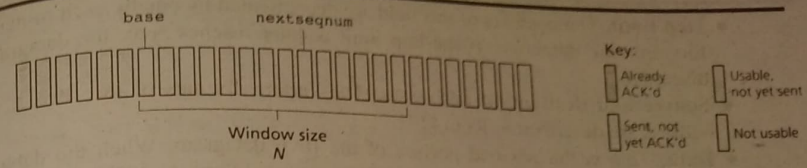
(3) Reaction to time out events :- This is congestion detection phase TCP manages more complex dynamics by maintaining a variable called threshold, which determined the window size at which slow start will end and congestion avoidance will begin. the variable threshold is set to a large value so that it has no initial effect. TCP sender enters the slow start phase after a timeout event.

b. Write the TCP segment structure and explain its fields. (04 Marks)

Ans. Refer 3.b. of Dec 18 / Jan 19

c. Elaborate the working of Go-Back N protocol. (04 Marks)

Ans. In Go-back-N protocol, the sender is allowed to transmit multiple packets (When available) without waiting for an acknowledgment, and is constrained to have no more than some maximum allowable number N, of unacknowledged packets in pipeline.

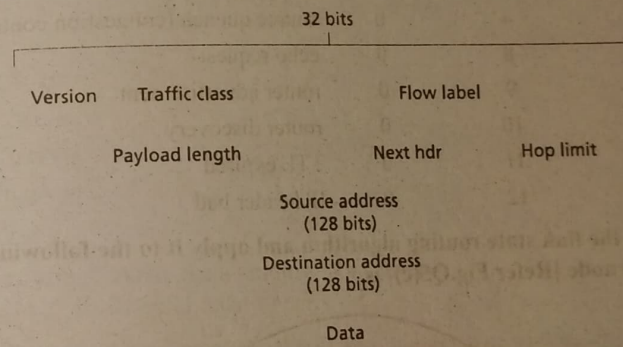


As the protocol operates, the windows slides forward [ ] - Not usable over the sequence number. N is refereed as the window size and GBN protocol as sliding window protocol. A packet's sequence number is carried in a fixed-length field in the packet header. If k is the number of bits in the packet sequence number field, the range of sequence numbers is thus [0,2k – 1].Sequence number in the interval (0, base-1) correspond to packets that have already been transmitted and acknowledged. Interval [base, next sequence - 1] correspond to the packets that have been sent but not yet acknowledged. Sequence numbers in the interval [next sequence, base+N–1] can be used for packets that can be sent immediately.

Module-3

5. a. Give the format of IPV6 datagram and explain the fields. (06 Marks)

Ans.



- **Expanded addressing capabilities.** IPv6 increases the size of the IP address from 32 to 128 bits and it has streamlined 40 byte header.
- **Flow labeling and priority.** IPV6 has an state that allows labeling of packets belonging to particular flows
- **Version.** This 4-bit field identifies the IP version number.
- **Traffic class.** This 8-bit field is similar to type of service.
- **Flow label.** is used to identify a flow of datagrams and is 20 bits
- **Payload length.** Is a 16-bit value i.e., an unsigned integer giving the number of bytes in the IPv6 datagram following the fixed-length, 40-byte datagram header.
- **Next header.** identifies the protocol to which the contents of this datagram will be delivered

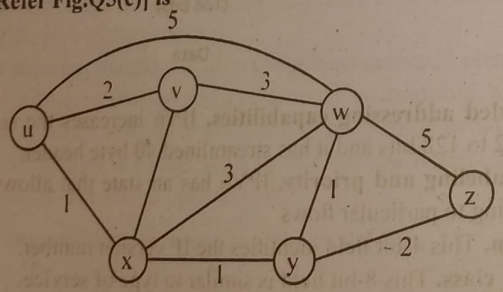
- **Hop limit.** The contents of this field are decremented by one by each router that forwards the datagram. If the hop limit counter reaches zero, the datagram is discarded.
- **Source and destination addresses.** The various formats of the IPv6 128-bit address are described in RFC 4291.
- **Data.** This is the payload portion of the IPv6 datagram. When the datagram reaches its destination, the payload will be removed from the IPV6 datagram and passed on to the protocol specified in the next header field.

b. What are the message types used in IGMP? **(03 Marks)**

Ans.

ICMP Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

c. Write the link state routing algorithm and apply it to the following graph with source node [Refer Fig.Q5(c)] is **(07 Marks)**



Ans. The algo is as below

1. Initialization:
2. $N' = \{u\}$
3. for all nodes v
4. if v is a neighbor of u

5. then D(v) = c(u,v)
6. else D(v) = ∞
7.
8. Loop
9. find w not in N' such that D(w) is a minimum
10. add w to N'
11. update D(v) for each neighbor v of w and not in N':
12. D(v) = min(D(v), D(w) + c(w,v))
13. /* new cost to v is either old cost to v or known least path cost to w plus cost from w to v */
14. until $N' = N$

- In the initialization step, the currently known least-cost paths from u to its directly attached neighbors, v, x, and w.
- In the first iteration, we look among those nodes not yet added to the set N' and find that node with the least cost as of the end of the previous iteration.
- In the second iteration, nodes v and y are found to have the least-cost path. The cost to the remaining nodes not yet in N' i.e., nodes v, w and z are updated via line 12 of the algorithm.

Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

OR

6. a. What is routing? Write the structure of a router. **(07 Marks)**

Ans. Refer Q.No. 5.a. of Dec 18 / Jan 19

b. List the broadcast routing algorithms? Explain any one of them. **(04 Marks)**

Ans. The broadcast algorithms are

(1) Uncontrolled flooding
(2) Controlled flooding
(3) Spanning tree broadcast

Uncontrolled flooding :- The source node sends a copy of the packet to all of its neighbors. When a node receives a broadcast packet, it duplicates the packet and forwards it to all of its neighbors. If the graph is connected, this scheme will eventually deliver a copy of the broadcast packet to all nodes in the graph. This simple scenario results in the endless cycling of two broadcast packets, one clockwise, and one counterclockwise. This broadcast storm, resulting from the endless multiplication of broadcast packets. The most obvious technique for achieving broadcast is flooding approach. In that two floodings types are there uncontrolled and controlled.

c. Describe the intra-AS routing protocols in detail **(05 Marks)**

Ans. Thre are two intra-AS routing protocols routing information protocol (RIP) and open shortest path first (OSPF). In RIP routing updates are enchanged between neighbours approximately every 30 seconds using a RIP response message. The response message sent by a router or a host contains a list upto 25 destination subnets within an AS, and senders distance to each of those subnets. Response messages are also known as RIP advertisements. RIP is a distance vectors protocol and version is specified in RFC 1058. RIP uses term hop which is the number of subnets traversed along the shortest path from source router to destination.

OSPF is a links state protocol that uses flooding of link state information. With OSPF, a router constructs a complete topological map (graph) of the entire autonomous system. The router then locally run Dijsktra's shortest path algorithm to determine a shortest path tree to all subnets. A router broadcasts routing information to all other routers in the autonomous system and also to its neighbouring routers. A router broadcasts link state information whenever there is a change in a link's state.

Module-4

7. a. Illustrate the two different approaches for routing to a mobile node. (08 Marks)

Ans. The two different approaches are (a) Indirect routing (b) Direct routing

(a) Indirect routing

The correspondent simply addresses the datagram to the mobile node's permanent address and sends the datagram into the network. Home agent responsible for interacting with a foreign agent to track the mobile node's COA. Its second job is to be on the lookout for arriving datagrams addressed to nodes whose home network is that of the home agent but currently resident in a foreign network.

The home agent intercepts these datagrams and then forwards in two-step process. The datagram is first forwarded to the foreign agent, using the mobile node's COA (Case of address) and then forwarded from the foreign agent to the mobile node

Indirect - routing provides following functions

(a) A mobile node to foreign agent protocol.
(b) A foreign agent to home agent registration protocol.
(c) A home agent datagram encapsulation protocol.
(d) A foreign agent decapsulation protocol.

(b) Direct routing to mobile node

In the direct routing approach, a correspondent agent in the correspondent's network first learns the COA of the mobile node. This can be done by having the correspondent agent query the home agent the mobile node has an up-to-date value for its COA registered with its correspondent agent. Two additional challenges are

(1) mobile-user location protocol

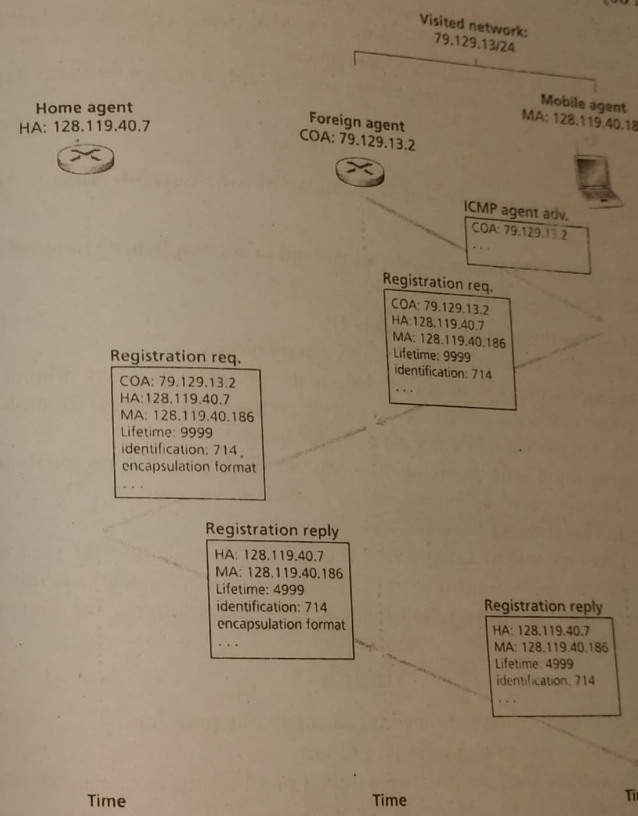
(2) When the mobile node moves from one foreign network to another, how will data now be forwarded to the new foreign network

Solution is to find the anchor foreign agent. When the mobile node moves to a new foreign network, the mobile node registers with the new foreign agent, and the new

foreign agent provides the anchor foreign agent with the mobile node's new COA). When the anchor foreign agent receives an encapsulated datagram for a departed mobile node, it can then re-encapsulate the datagram and forward it to the mobile node using the new COA. If the mobile node later moves yet again to a new foreign network, the foreign agent in that new visited network would then contact the anchor foreign agent in order to set up forwarding to this new foreign network.

b. With a neat diagram, bring out the steps for mobile node registration to home agent.

Ans. Four steps are involved **(08 Marks)**



1. Following the receipt of foreign agent advertisement, a mobile node sends a mobile IP registration message to the foreign agent. The registration message is carried within a UDP datagram and sent to port 434. The registration agent carries a COA advertised by the foreign agent, the address of the home agent, the

permanent address of the mobile node (MA), the requested lifetime of registration, and a 64-bit registration identification.

2. The foreign agent receives the registration message and records the mobile node's permanent IP address. The foreign agent now knows that it should be looking for datagrams containing an encapsulated datagram whose destination address matches the permanent address of the mobile node.
3. The home agent receives the registration request and checks for authenticity and correctness. The home agent binds the mobile node's permanent IP address with the COA. In future, datagrams arriving at the home agent and addressed to the mobile node will now be encapsulated. The home agent sends a mobile IP registration reply containing the HA, MA, actual registration lifetime, and the registration identification of the request.
4. The foreign agent receives the registration reply and then forwards it to the mobile node.

OR

8. a. Bring out the components of 3G Cellular Network architecture. (08 Marks)

Ans. Refer Q.No. 7.a. of Dec 18 / Jan 19

b. State handoff? What are the steps involved in accomplishing handoff.(05 Marks)

Ans. Refer Q.No. 8.a. of Dec 18 / Jan 19

c. Explain the three phases of mobile IP. (03 Marks)

ns. The mobile IP standard consists of three main pieces

- **Agent discovery -** Mobile IP defines the protocols used by a home or foreign agent to advertise its services to mobile nodes, and protocols for mobile nodes to solicit the services of a foreign or home agent.
- **Registration with home agent -** Mobile IP defines the protocols used by the mobile node and/or foreign agent to register and deregister COAs with a mobile node's home agent.
- **Indirect routing of datagrams -** The standard also defines the manner in which datagrams are forwarded to mobile nodes by a home agent, including rules for forwarding datagrams, rules for handling error conditions, and several forms of encapsulation.

Module-5

9. a. Bring out the leaky bucket mechanism for traffic policing. (07 Marks)

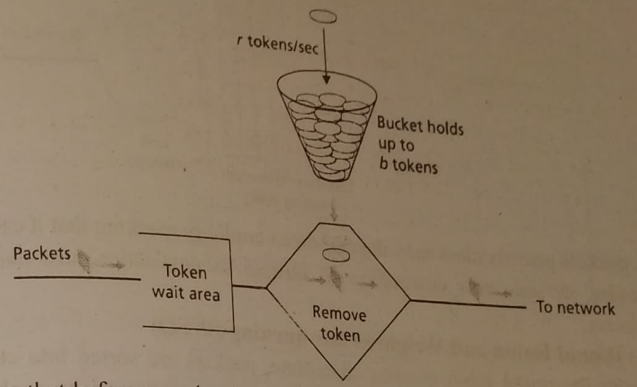
Ans. Three important policing criteria are present

1) Average rate - The network may wish to limit the long-term average rate at which a flow's packets can be sent into the network.

2) Peak rate - Peak rate constraint limits the maximum number of packets that can be sent over a shorter period of time.

3) Burst size - The network may wish to limit the maximum number of packets that can be sent into the network over an extremely short interval of time

The leaky bucket mechanism is an abstraction that can be used to characterized these policing limits



Suppose that before a packet is transmitted into the network, it must first remove a token from the token bucket. If the token bucket is empty, the packet must wait for a token. These can be atmost b tokens in the bucket, minimum burst size for a leaky bucket period flow is b packets. Because the token generation rate is r, maximum number of packets that can enter the network of any interval of time length t is rtt+5. Thus, token generation rate, r serves to limit the long - term average rate at which packets can enter the network. Two leaky bucket can be used to police a flow's peak rate in addition to the long term average rate.

b. Classify the multimedia network applications. (03 Marks)

Ans. These are three classes of multimedia applications

(1) Streaming stored audio/video - Client requests on demand compressed audio or video files that are stored on servers.

(2) Streaming - In streaming stored audio/video application, a client typically begins playout of the audio/video a few seconds after it begins receiving the file from the server.

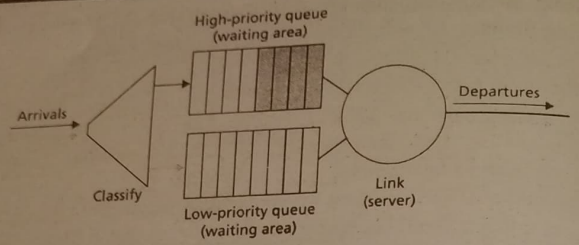
(3) Real time interaction audio/video - This allows people to use audio/video to communicate with each other in real time. Real time interactive audio is often referred to as internet phone.

c. Describe the link scheduling mechanisms. (06 Marks)

Ans. Following are the scheduling mechanisms

(1) First in First out - Packets arriving at the link output queue wait for transmission if the link is currently busy transmitting another packet. If there is not sufficient buffering space to hold the arriving packet, the queue's packet-discarding policy then determines whether the packet will be dropped (lost) or whether other packets will be removed. It is first come first serve basis.

(2) Priority queuing - Packets arriving at the output link are classified into priority classes at the output queue, as shown in Figure



A packets priority class may depend on an enplicity masking that it carries in packet header, its source or destination IP address its destination post number ot other criteria.

(3) Round Robin and Weighted fair queuing (WFQ)

Under the round robin queuing discipline, packets are sorted into classes as with priority queuing. A class 1 packet is transmitted, followed by a class 2 packet, followed by a class 1 packet, followed by a class 2 packet, and so on. A work serving round robin discipline looks for a packet of a given class but finds none will immediately check the next class in round robin sequence.

A generalized abstraction of round robin queuing is weighted for queuing. Arriving packets are classified and queued in the appropriate per clan waiting area. A WFQ scheduler will serve classes in a circular manner first serving class 1 then serving class 2. WFQ will immediately move on to the next class in the service sequence when it finds an empty class queue.

OR

10. a. List the categories of streaming stored video. Explain any one of them.(08 Marks)

Ans. The categories of stores video are

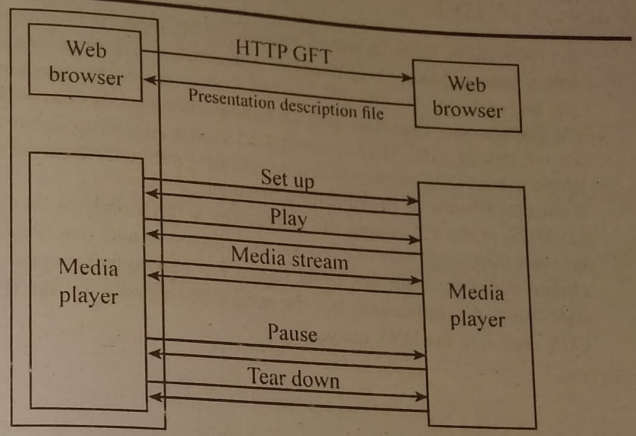
(1) Accessing video through a web server

(2) Sending multimedia from a streaming server to a helper application

(3) Real time streaming protocol

Real time streaming protocol allows a media player to control the transmission of media stream. Control actions include pause/resume, repositioning of playback, fast forward and rewind. RTSP is an out of band protocol. RTSP message are sent out of band where as media stream, whose packet is not defined by RTSP is considered "in-band". RTSP messages use a different port no 544 from media stream.

The web browser first requests a presentation description file from a web server. Each reference to the continues media file begins with URL method RTSP 11.



The web server encapsulates the presentation description file in an HTTP response message and sends the messages to the browser. When the browser receives the HTTP response message, the browser involves a media player based on content type field of the message. The presentation description file includes references to media stream using URL method rtsp 11. The player and the server then send each other a series of RTSP messages. The player sends an RTSP SETUP PLAY request and server responds with RTSP ok message. When the uses is finished, media player sends an RTSP TEAR DOWN request and the server confirms with RTSP ok response.

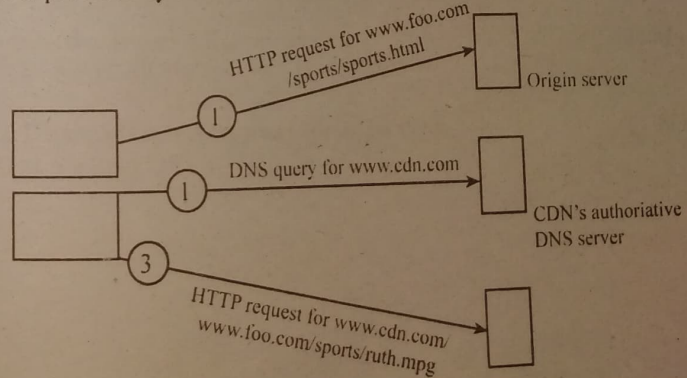
b. Explain the working of CDN. **(08 Marks)**

Ans. A CDN company typically provides its content distribution services as follows.

1) The CDN company installs hundreds of CDN servers throughout the internet. CDN servers are placed in data center.

2) The CDN replicates its customer's content in CDN server, whenever a customer updates its content, CDN redistributes the fresh content to CDN servers.

3) CDN Company provides a mechanism so that when a client requests content, the content is provided by CDN server that can but deliver content to specific client.



(1) The browser sends its request for base HTML object to origin server. www.foo.com which sends requested HTML object to the browse. The browser passes the HTML file and finds reference to http : \\www.cdn.com\www.foo.com/sports/ruth.mpg
(2) The browser then does a DNS lookup on www.cdn.com with is the host name for referenced URL. DNS is configured so that all queries about www.cdn.com that arrive to root DNS server are sent to an authoritative DNS server receives the query, it entracts IP address of requesting browser.
(3) DNS in the requesting client receives a DNS reply with the IP address. The browser then sends in HTTP request to CDN server with that IP address. The browser obtains ruth.mpg from this CDN server. For subsequent requests from www.CDN. com. the client continues to use the same CDN server since the IP address for www. CDN.com is in the DNS cache.

Fifth Semester B.E. Degree Examination, CBCS - Dec 2019 / Jan 2020
Computer Networks

Time: 3 hrs. **Max. Marks:** 100

Note : Answer any FIVE full questions, selecting ONE full question from each module.

Module-1

1. a. Which protocol can be used for fetching web pager? Explain its working with request and response message formats. **(10 Marks)**

Ans. Refer Q.No. 1.b. of Dec 2018 / Jan 2019

b. Explain the services offered by DNS and also explain the DNS record and massage formal. **(10 Marks)**

Ans. Refer Q.No. 1.a. of June / July 2019

OR

2. a. Explain the working FTP along with its commands. **(08 Marks)**

Ans. Refer Q.No. 2.a. of Dec 2018 / Jan 2019

b. Compare HTTP and SMTP. **(04 Marks)**

Ans. Refer Q.No. 2.a. of MQP - 1

c. Illustrate how P2P architecture can be adopted in file. Sharing application like bit torrentz. **(08 Marks)**

Ans. Refer Q.No. 2.a. of June / July 2018

Module-2

3. a. Draw and explain the FSM for sender site and receiver site of rdt 2.0 protocol. **(07 Marks)**

Ans. Refer Q.No. 3.a. of June / July 2019

b. Explain selective repeat ARQ protocol. **(06 Marks)**

Ans. Refer Q.No. 3.c. of MQP - 1

c. Draw TCP segment structure and explain its fields. **(07 Marks)**

Ans. Refer Q.No. 3.a. of MQP - 2

OR

4. a. Suppose that two measured sample RTT values are 106ms and 120ms. i) Compute Estimated RTT after each of these Sample RTT value is obtained. Assume a = 0.125 and Estimated RTT is 100ms. Just before first of the samples obtained. ii) Compute DeVRTT. Assume p = 0.25 and DeVRTT is 5ms before first of the samples obtained. (06 Marks)

Ans. Refer Q.No. 3.c. of June / July 2019

b. Explain how connection establishment and termination is handled by TCP. (07 Marks)

Ans. Refer Q.No. 4.c. of MQP - 3

c. What is congestion in network? Explain how TCP handles congestion. (07 Marks)

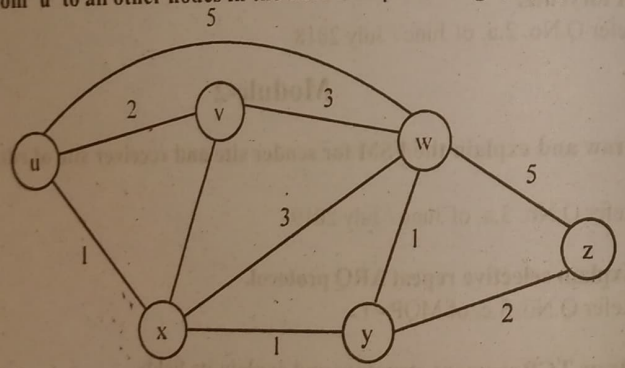
Ans. Refer Q.No. 4.a. of MQP - 1

Module-3

5. a. What is routing? With a neat diagram, explain the structure of a router. (10 Marks)

Ans. Refer Q.No. 5.b. of MQP - 2

b. Write link state routing algorithm, consider the following network with the indicated link costs. Apply link state routing algorithm to compute the shortest path from 'u' to all other nodes in the network. [Refer Fig.Q5(b)]. (10 Marks)



Ans. Refer Q.No. 5.c. of June / July 2019

OR

6. a. Draw IPV6 datagram format. Explain its fields. (06 Marks)

Ans. Refer Q.No. 5.a. of June / July 2019

b. Illustrate the working of RIP protocol. (07 Marks)

Ans. Refer Q.No. 6.a. of June / July 2019

c. List the broadcast routing algorithm. Explain any one of them. (07 Marks)

Ans. Refer Q.No. 6.b. of June / July 2019

Module-4

7. a. With a neat diagram, explain the components of 3G cellular network architecture. (10 Marks)

Ans. Refer Q.No. 7.a. of Dec 2018 / Jan 2019

b. Explain two different types of routing approaches to mobile nodes. (10 Marks)

Ans. Refer Q.No. 7.a. of June / July 2019

OR

8. a. Explain the three phases of mobile IP. (10 Marks)

Ans. Refer Q.No. 8.c. of June / July 2019

b. What is handoff? What are the steps involved in accomplishing handoff. (10 Marks)

Ans. Refer Q.No. 8.a. of Dec 2018 / Jan 2019

Module-5

9. a. Explain three different types of streaming stored video. (10 Marks)

Ans. Refer Q.No. 10.a. of June / July 2019

b. Explain the working of CDN. (10 Marks)

Ans. Refer Q.No. 10.b. of June / July 2019

OR

10. a. Describe the leaky bucket policing mechanism. (06 Marks)

Ans. Refer Q.No. 9.a. of June / July 2019

b. Explain the various packet scheduling mechanism. (08 Marks)

Ans. Refer Q.No. 9.c. of June / July 2019

c. Explain the properties of Video. (06 Marks)

Ans. Refer Q.No. 10.b. of MQP - 2