B.Tech.

SIXTH SEMESTER EXAMINATION, 2008-09

COMPUTER NETWORK

(TCS-602)

Time : 3 Hours] *[Total Marks : **100***

Q. 1. Attempt any four parts of the following : **5×4 = 20**

Q. 1. (a) What is the number of cable links required for n devices connected in mesh, ring, bus and star topology ?

Ans. The number of cables for each type of network is:

a. Mesh: n (n - 1) / 2

b. Star: n

c. Ring: n - 1

d. Bus: one backbone and n drop lines

Q. 1. (b) List the various layers of OSI model. Briefly explain the working of each of them.

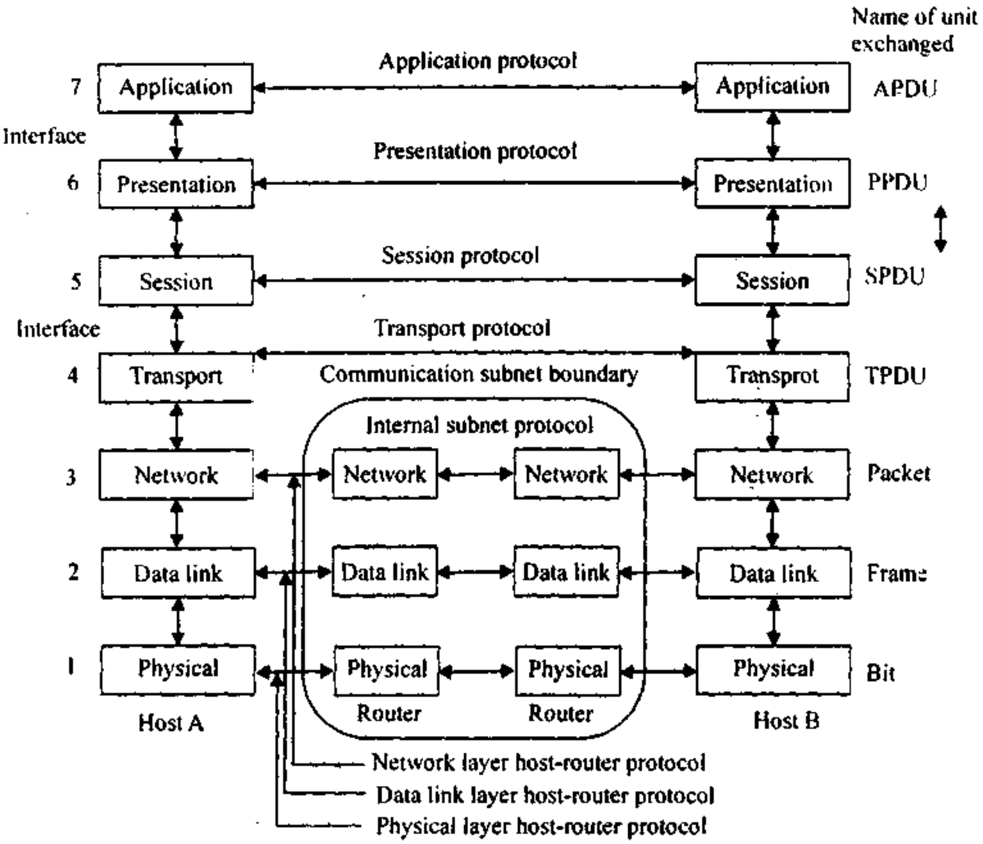
Ans. The OSI model (minus the physical medium) is shown in Fig. 1-20. This model is based on a proposal developed by the International Standards Organization (ISO) as a first step toward international standardization of the protocols used in the various layers (Day and Zimmermann, 1983). It was revised in 1995 (Day, 1995).

The model is called the ISO OSI (Open Systems Interconnection) Reference Model because it deals with connecting open systems-that is, systems that are open for communication with other systems. We will just call it the OSI model for short.

The OSI model has seven layers. The principles that were applied to arrive at the seven layers can be briefly summarized as follows:

1. A layer should be created where a different abstraction is needed.
2. Each layer should perform a well-defined function.
3. The function of each layer should be chosen with an eye toward defining internationally standardized 37 protocols.
4. The layer boundaries should be chosen to minimize the information flow across the interfaces.
5. The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity and small enough that the architecture does not become unwieldy.

Below we will discuss each layer of the model in turn, starting at the bottom layer. Note that the OSI model itself is not a network architecture because it does not specify the exact services and protocols to be used in each layer. It just tells what each layer should do. However, ISO has also produced standards for all the layers, although these are not part of the reference model itself. Each one has been published as a separate international standard.



The OSI reference model.

The Physical Layer : The physical layer is concerned with transmitting raw bits over a communication channel. The design issues have to do with making sure that when one side sends a 1 bit, it is received by the other side as a 1 bit, not as a 0 bit. Typical questions here are how many volts should be used to represent a 1 and how many for a 0, how many nanoseconds a bit lasts, whether transmission may proceed simultaneously in both directions, how the initial connection is established and how it is torn down when both sides are finished, and how many pins the network connector has and what each pin is used for. The design issues here largely deal with mechanical, electrical, and timing interfaces, and the physical transmission medium, which lies below the physical layer.

The Data Link Layer : The main task of the data link layer is to transform a raw transmission facility into a line that appears free of undetected transmission errors to the network layer. It accomplishes this task by having the sender break up the input data into data frames (typically a few hundred or a few thousand bytes) and transmits the frames sequentially. If the service is reliable, the receiver confirms correct receipt of each frame by sending back an acknowledgement frame.

The Network Layer : The network layer controls the operation of the subnet. A key design issue is determining how packets are routed from source to destination. Routes can be based on static tables that are "wired into" the network and rarely changed. They can also be determined at the start of each conversation, for example, a terminal session (e.g., a login to a remote machine). Finally, they can be highly dynamic, being determined anew for each packet, to reflect the current network load. If too many packets are present in the subnet at the same time, they will get in one another's way, forming bottlenecks. The control of such congestion also belongs to the network layer. More generally, the quality of service provided (delay, transit time, jitter, etc.) is also a network layer issue.

The Transport Layer : The basic function of the transport layer is to accept data from above, split it up into smaller units if need be, pass these to the network layer, and ensure that the pieces all arrive correctly at the other end. The transport layer also determines what type of service to provide to the session layer, and, ultimately, to the users of the network. The most popular type of transport connection is an error-free point-to-point channel that delivers messages or bytes in the order in which they were sent. However, other possible kinds of transport service are the transporting of isolated messages, with no guarantee about the order of delivery, and the broadcasting of messages to multiple destinations. The type of service is determined when the connection is established.

The Session Layer : The session layer allows users on different machines to establish sessions between them. Sessions offer various services, including dialog control (keeping track of whose turn it is to transmit), token management (preventing two parties from attempting the same critical operation at the same time), and synchronization (checkpointing long transmissions to allow them to continue from where they were after a crash).

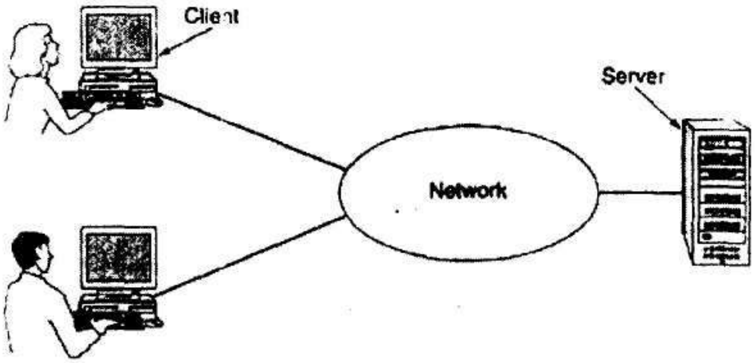
The Presentation Layer : Presentation layer is concerned with the syntax and semantics of the information transmitted. In order to make it possible for computers with different data representations to communicate, the data structures to be exchanged can be defined in an abstract way, along with a standard encoding to be used "on the wire." The presentation layer manages these abstract data structures and allows higher-level data structures (e.g., banking records), to be defined and exchanged.

The Application Layer : The application layer contains a variety of protocols that are commonly needed by users. One widely-used application protocol is HTTP (HyperText Transfer Protocol), which is the basis for the World Wide Web. When a browser wants a Web page, it sends the name of the page it wants to the server using HTTP. The server then sends the page back. Other application protocols are used for file transfer, electronic mail, and network news.

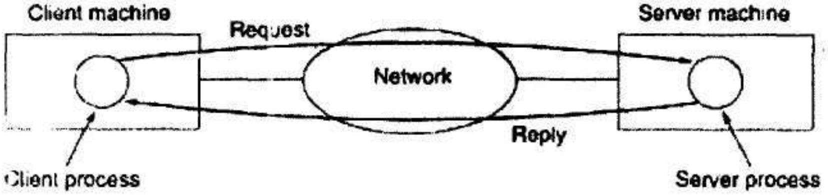
Q. 1. (c). Explain the different uses of computer networks.

Ans. 1. Business Applications : For smaller companies, all the computers are likely to be in a single office or perhaps a single building, but for larger ones, the computers and employees may be scattered over dozens of offices and plants in many countries. Nevertheless, a sales person in New York might sometimes need access to a product inventory database in Singapore. In other words, the mere fact that a user happens to be 15,000 km away from his data should not prevent him from using the data as though they were local. This goal may be summarized by saying that it is an attempt to end the "tyranny of geography."

A network with two clients and one server

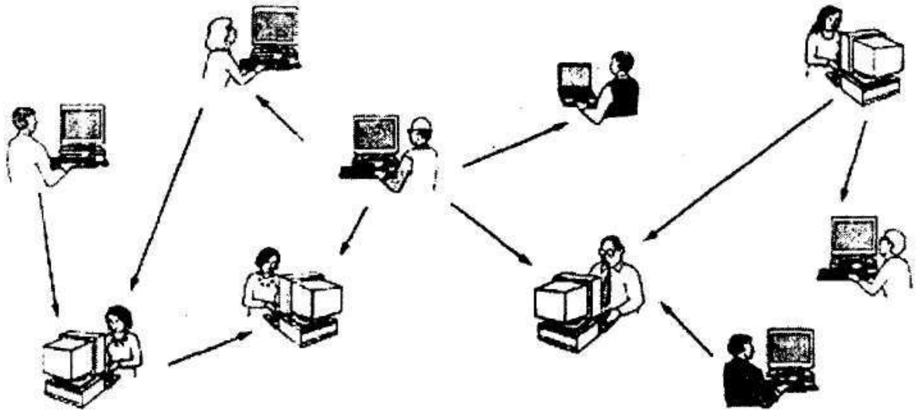


The client-server model involves requests and replies



2. Home Applications : Some of the more popular uses of the Internet for home users are as follows:

1. Access to remote information.
2. Person-to-person communication.
3. Interactive entertainment.
4. Electronic commerce.



Some forms of e-commerce.

Tag	Full name	Example
B2C	Business-to-consumer	Ordering books on-line
B2B	Business-to-consumer	Car manufacturer ordering tires from supplier
G2C	Government-to-consumer	Government distributing tax forms electronically
C2C	Consumer-to-consumer	Auctioning second-hand products on line
P2P	Peer-to-peer	File sharing

3 Mobile Users : Mobile computers, such as notebook computers and personal digital assistants (PDAs), are one of the fastest growing segments of the computer industry. Many owners of these computers have desktop machines back at the office and want to be connected to their home base even when away from home or en route. Since having a wired connection is impossible in cars and airplanes, there is a lot of interest in wireless networks

Combinations of wireless networks and mobile computing.

Wireless	Mobile	Applications
No	No	Desktop computers in offices
No	Yes	A notebook computer used in a hotel room
Yes	No	Networks in older, unwired buildings
Yes	Yes	Portable office; PDA for store inventory

Q. 1. (d) What is the total delay (latency) for a frame size of 10 million bits that is being set up on link with 15 routers each having a queuing time of 2 μs. and a processing time of 1 μs ? The length of link is 3000 km. The speed of light inside the link is 2×10^8 m/s. The link has bandwith of 6 Mbps.

Ans. Frame size = 10 million bits

15 router are used, having a queuing time of 2 μs. and a processing time of 1 μs.

Length of the link is 3000 Km = 3×10^6 m

Speed of light inside the link = 2×10^8 m/s.

Link has bandwidth = 6 mbps = $6 \times 1024 \times 1024$ bps

$$\text{Total delay time} = \frac{10^6 \times 3 \times 10^{-6} \times 3 \times 10^{-6} \times 2 \times 10^8}{6 \times 1024 \times 1024 \times 15 \times 1.5 \times 10}$$

$$= \frac{1.4305 \times 200}{15 \times 10} = 1.9073$$

Total delay time = 1.9073 seconds.

Q. 1. (e) Two networks each provide reliable connection-oriented service. One of them offers a reliable byte stream and other reliable message stream. Are these identical ? justify your answer.

Ans. Message and byte streams are different. In a message stream, the network keeps track of message boundaries. In a byte stream, it does not. For example, suppose a process writes 1024 bytes to a connection and then a little later writes another 1024 bytes. The receiver then does a read for 2048 bytes. With a message stream, the receiver will get two messages, of 1024 bytes each. With a byte stream, the message boundaries do not count and the receiver will get the full 2048 bytes as a single unit. The fact that there were originally two distinct messages is lost.

Q . 1 (f) How long does it take to transmit an 8 inch by 10 inch image by facsimile over an ISDN B channel ? The facimile digitizes the image into 300 pixel per inich and assign 4 bit per pixel.

Ans. 8 inch × 10 inch image is given –

speed of ISDN-B channel = 64 Kbit/sec

$$= 64 \times 1024 \text{ bit/sec}$$

The image is digitized into 300 pixel/inch & assign 4bit/pixel

so the total time taken $= \dfrac{8 \times 10 \times 300 \times 4}{64 \times 1024}$

$$= \frac{3000}{2 \times 1024} = \frac{1500}{1024}$$

total time taken to transmit = 1.4648 seconds

Q. 2. Attemp any four parts of the following : 10×2 = 20

Q. 2. (a) What is hamming code? Explain its working by suitable example.

Ans. A Hamming code is a linear error-correcting code. Hamming codes can detect up to two simultaneous bit errors, and correct single-bit errors; thus, reliable communication is possible when the Hamming distance between the transmitted and received bit patterns is less than or equal to one. By contrast, the simple parity code cannot correct errors, and can only detect an odd number of errors.

Parity adds a single bit that indicates whether the number of 1 bits in the preceding data was even or odd. If an odd number of bits is changed in transmission, the message will change parity and the error can be detected at this point. (Note that the bit that changed may have been the parity bit itself!) The most common convention is that a parity value of 1 indicates that there is an odd number of ones in the data, and a parity value of 0 indicates that there is an even number of ones in the data. In other words: The data and the parity bit together should contain an even number of 1s.

The key to the Hamming Code is the use of extra parity bits to allow the identification of a single error. Create the code word as follows:

1. Mark all bit positions that are powers of two as parity bits. (positions 1, 2, 4, 8, 16, 32, 64, etc.)
2. All other bit positions are for the data to be encoded. (positions 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, etc.)
3. Each parity bit calculates the parity for some of the bits in the code word. The position of the parity bit determines the sequence of bits that it alternately checks and skips.
   Position 1: check 1 bit, skip 1 bit, check 1 bit, skip 1 bit, etc. (1,3,5,7,9,11,13,15,...)
   Position 2: check 2 bits, skip 2 bits, check 2 bits, skip 2 bits, etc. (2,3,6,7,10,11,14,15,...)
   Position 4: check 4 bits, skip 4 bits, check 4 bits, skip 4 bits, etc. (4,5,6,7,12,13,14,15,20,21,22,23,...)
   Position 8: check 8 bits, skip 8 bits, check 8 bits, skip 8 bits, etc. (8-15,24-31,40-47,...)
   Position 16: check 16 bits, skip 16 bits, check 16 bits, skip 16 bits, etc. (16-31,48-63,80-95,...)

Position 32: check 32 bits, skip 32 bits, check 32 bits, skip 32 bits, etc. (32-63,96-127,160-191,...) etc.

4. Set a parity bit to 1 if the total number of ones in the positions it checks is odd. Set a parity bit to 0 if the total number of ones in the positions it checks is even.

Here is an example:

A byte of data: 10011010

Create the data word, leaving spaces for the parity bits: _ _ 1 _ 0 0 1 _ 1 0 1 0

Calculate the parity for each parity bit (a ? represents the bit position being set):

- Position 1 checks bits 1,3,5,7,9,11: ? _ 1 _ 0 0 1 _ 1 0 1 0. Even parity so set position 1 to a 0: 0 _ 1 _ 0 0 1 _ 1 0 1 0
- Position 2 checks bits 2,3,6,7,10,11: 0 ? 1 _ 0 0 1 _ 1 0 1 0. Odd parity so set position 2 to a 1: 0 1 1 _ 0 0 1 _ 1 0 1 0
- Position 4 checks bits 4,5,6,7,12: 0 1 1 ? 0 0 1 _ 1 0 1 0. Odd parity so set position 4 to a 1: 0 1 1 1 0 0 1 _ 1 0 1 0
- Position 8 checks bits 8,9,10,11,12: 0 1 1 1 0 0 1 ? 1 0 1 0. Even parity so set position 8 to a 0: 0 1 1 1 0 0 1 0 1 0 1 0
- Code word: 011100101010.

Q. 2. (b) a channel has a bit rate of 4Kbps and propogation delay of 20 msec. what will be the size of frame range so that stop & wait give an efficiency of at least 50 percent ?

Ans. Efficiency will be 50% when the time to transmit the frame equals the roundtrip propagation delay. At a transmission rate of 4 bits/ms, 160 bits takes 40 ms.

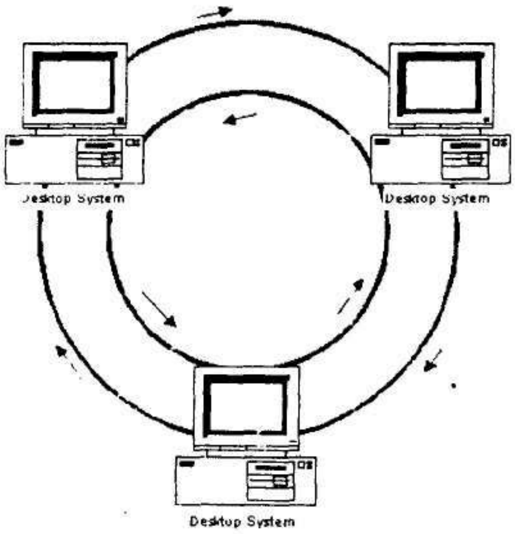
For frame sizes above 160 bits, stop-and-wait is reasonably efficient.

Q. 2. (c) How FDDI ring can be used as a back bone to connect LANs & computers? Also discuss the FDDI cabling in brief.

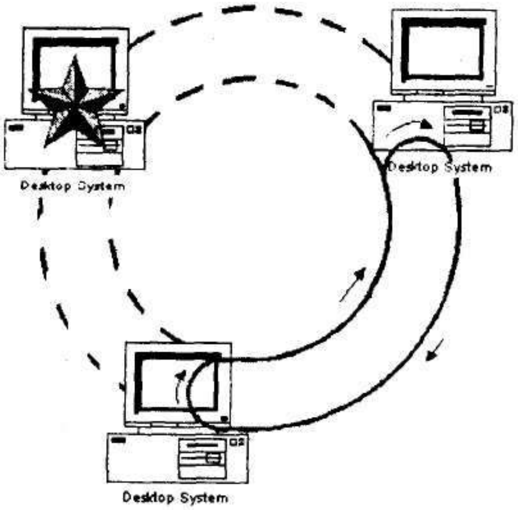
Ans. The Fiber Distributed Data Interface (FDDI) is a high speed local area network which has been defined as a standard by an American National Standards Institute committee, ANSI X3T9.5 and by ISO.

FDDI is a 100 Mbps, token-passing, single or dual ring interface that can be implemented with Fiber Optic or Unshielded Twisted-Pair (UTP) media. A Timed Token Protocol (TTP) is used to control when a station can transmit data to the network. A station can transmit a message on the network only after it has received a token. Upon receiving the token, a station begins transmitting data. The station can transmit until the message is transmitted or until the TTP timer expires. This allows all stations fair access to the ring. Once the message is sent or the timer expires, the station generates a new token and releases it on the ring. Any downstream station with data to send can capture the token and repeat the timed-transmission cycle.

A dual ring configuration for the network media provides a secondary backup ring in case of a fault on the primary ring. It is typically implemented as a campus backbone or within buildings where a failure in the primary ring would have serious consequences. A break in the primary ring causes the two stations on each side of the fault to automatically wrap the data to the secondary ring. Stations in a single ring configuration can only attach to the primary ring. There is no secondary backup path in the event of a failure.



FDDI - all stations functioning



FDDI - one station is down

Q. 2. (d) Compare the delay of pure ALOHA to slotted ALOHA at low load.

Ans. PURE ALOHA

A station can transmit whenever it has data to send.

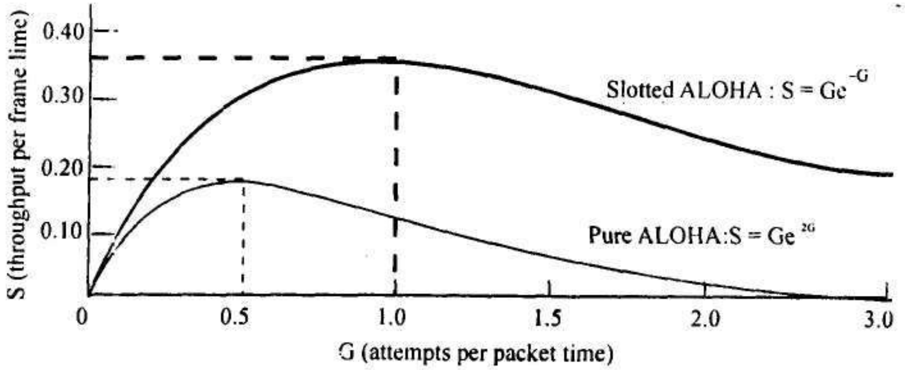
If a frame suffers collision, sender waits a random amount of time and sends again.

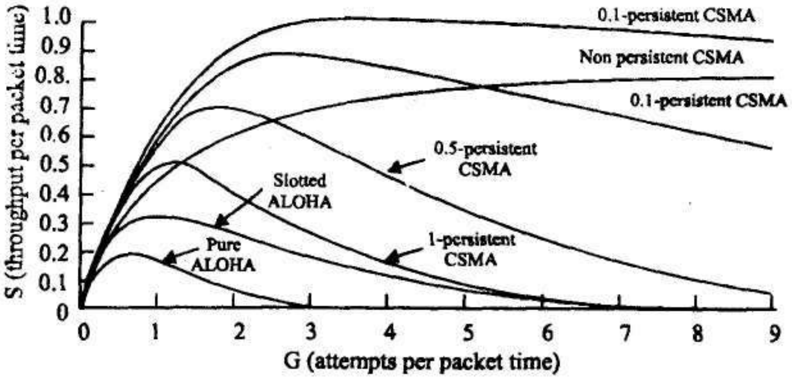
Throughput maximized when all frames have the same length.

Maximum throughput = 0.184.

Slotted ALOHA

Time divided into slots, slot length equals frame transmission time.





When frame ready for transmission, wait until start of next slot.

Maximum throughput = 0.368.

Q. 2. (f) Explain the following protocols:

(i) Adaptive Tree Walk Protocol

The stations are organized as the leaves of a binary tree.

In slot 0, node 0 is searched (all stations under node 0 are permitted to try to acquire the channel).

If a collision occurs in a slot, search the node's left and right children.

If the slot is idle or only cne station transmits in the slot, stop the search of the node.

Adaptive Tree Walk Algorithm

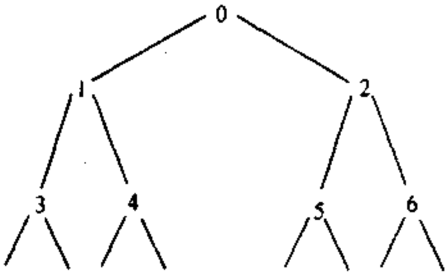
- Each node at level I has N. 2-I station under it.
- Q ready stations -uniform.y distributed at level I 2-IQ
- level at which search begins

-2-IQ=1

-i= log2Q

Adaptive Tree Walk Protocol

Example



Slot 0: C*, E*, F*, H* (all nodes under node 0 can try), conflict

slot 1: C* (all nodes under node 1 can try), C sends

slot 2: E*, F*, H*(all nodes under node 2 can try), conflict

slot 3: E*, F* (all nodes under node 5 can try), conflict

slot 4: E* (all nodes under E can try), E sends

slot 5: F* (all nodes under F can try), F sends

slot 6: H* (all nodes under node 6 can try), H sends.

The number of stations permitted to transmit can be dynamically adjusted, depending upon traffic.

o If high traffic, start the search at a higher level in the tree.

o If low traffic, start the search at a lower level in the tree.

(ii) Binary exponential back off algorithms

Exponential backoff is an algorithm that uses feedback to multiplicatively decrease the rate of some process, in order to gradually find an acceptable rate. It is often used in network congestion avoidance to help determine the correct sending rate.

An example of an exponential backoff algorithm

This example is from the Ethernet protocol, where a sending host is able to know when a collision has occurred (that is, another host has tried to transmit), when it is sending a frame. If both hosts attempted to retransmit as soon as a collision occurred, there would be yet another collision - and the pattern would continue forever. The hosts must choose a random value within an acceptable range to ensure that this situation doesn't happen. An exponential backoff algorithm is therefore used. The figure 51.2?s has been given here as an example. However, 51.2?s could be replaced by any positive value, in practice.

1. When a collision first occurs, send a `Jamming signal' to prevent further data being sent.
2. Resend a frame after either 0 seconds or 51.2μs, chosen at random.
3. If that fails, resend the frame after either 0s, 51.2μs, 102.4μs, or 153.6μs.
4. If that still doesn't work, resend the frame after k .51.2μs , where k is a random number between 0 and $2^3 - 1$.
5. In general, after the nth failed attempt, resend the frame after k .51.2μs , where k is a random number between 0 and $2^n - 1$.

In a variety of computer networks, binary exponential backoff or truncated binary exponential backoff refers to an algorithm used to space out repeated retransmissions of the same block of data.

Examples are the retransmission of frames in carrier sense multiple access with collision avoidance (CSMA/CA) and carrier sense multiple access with collision detection (CSMA/CD) networks, where this algorithm is part of the channel access method used to send data on these network. In Ethernet networks, the algorithm is commonly used to schedule retransmissions after collisions. The retransmission is delayed by an amount of time derived from the slot time and the number of attempts to retransmit.

After i collisions, a random number of slot times between 0 and 2i ? 1 is chosen. For the first collision, each sender might wait 0 or 1 slot times. After the second collision, the senders might wait 0, 1, 2, or 3 slot times, and so forth. As the number of retransmission attempts increases, the number of possibilities for delay increases.

The 'truncated' simply means that after a certain number of increases, the exponentiation stops; i.e. the retransmission timeout reaches a ceiling, and thereafter does not increase any further. For example, if the ceiling is set at i=10, then the maximum delay is 1023 slot times.

Because these delays cause other stations who are sending to collide as well, there is a possibility that, on a busy network, hundreds of people may be caught in a single collision set. Because of this possibility, after 16 attempts at transmission, the process is aborted.

Q. 3. Attempt any two parts of the following : 10×2 = 20

Q.3. (a). (i) Differentiate between adaptive & non adaptive routing algorithms.

Ans. Non-Adaptive Routing Algorithms

Non-adaptive routing algorithms do not base their routing decisions on the current state of the network

This Procedure is sometimes called static routing

do not base their routing decisions on measurements or estimates of the current traffic and topology. Instead, the choice of the route to use to get from I to J (for all I and J) is computed in advance, off-line, and downloaded to the routers when the network is booted. This procedure is sometimes called static routing

Flooding

Shortest Path Routing.

Problems non-adaptive Algorithms

Problems with non-adaptive algorithms

If traffic levels in different parts of the subnet change dramatically and often, non adaptive routing algorithms are unable to cope with these changes

Lots of computer traffic is bursty, but non-adaptive routing algorithms are usually based on average traffic conditions

Adaptive routing algorithms can deal with these situations.

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

Q.3. (a). (ii) What are the limitations of leaky bucket algorithm? How these are overcomed ?

Ans. The Token Bucket Algorithm : The leaky bucket algorithm enforces a rigid output pattern at the average rate, no matter how bursty the traffic is. For many applications, it is better to allow the output to speed up somewhat when large bursts arrive, so a more flexible algorithm is needed, preferably one that never loses data. One such algorithm is the token bucket algorithm. In this algorithm, the leaky bucket holds tokens, generated by a clock at the rate of one token every ?T sec. In Fig. 5-34(a) we see a bucket holding three tokens, with five packets waiting to be transmitted. For a packet to be transmitted, it must capture and destroy one token. In Fig. 5-34(b) we see that three of the five packets have gotten through, but the other two are stuck waiting for two more tokens to be generated.

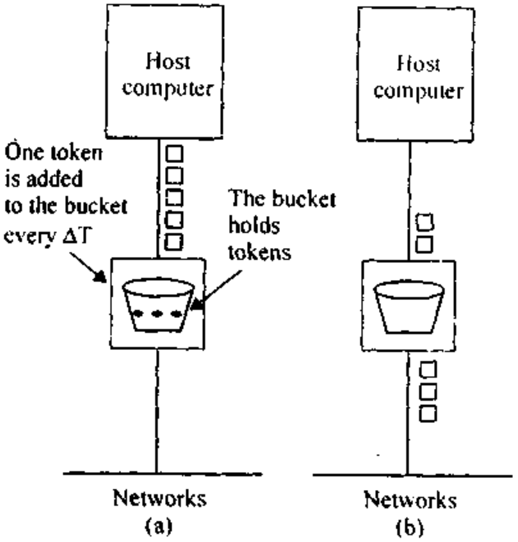


Fig. The token bucket algorithm (a) Before (b) After

The token bucket algorithm provides a different kind of traffic shaping than that of the leaky bucket algorithm. The leaky bucket algorithm does not allow idle hosts to save up permission to send large bursts later. The token bucket algorithm does allow saving, up to the maximum size of the bucket, n. This property means that bursts of up to n packets can be sent at once, allowing some burstiness in the output stream and giving faster response to sudden bursts of input.

Another difference between the two algorithms is that the token bucket algorithm throws away tokens (i.e., transmission capacity) when the bucket fills up but never discards packets. In contrast, the leaky bucket algorithm discards packets when the bucket fills up.

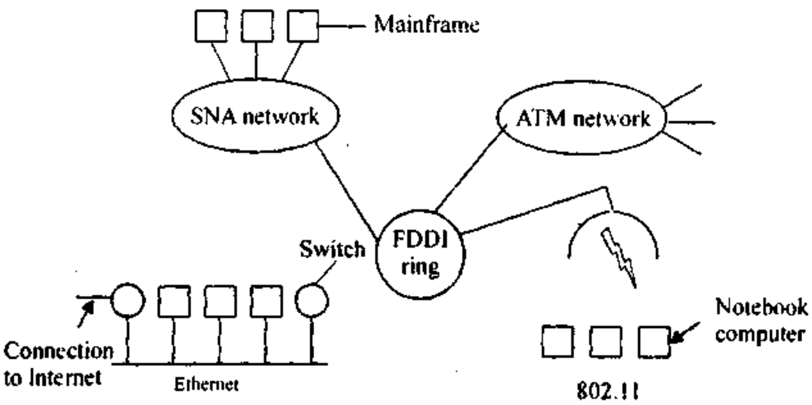
Here, too, a minor variant is possible, in which each token represents the right to send not one packet, but k bytes. A packet can only be transmitted if enough tokens are available to cover its length in bytes. Fractional tokens are kept for future use.

The leaky bucket and token bucket algorithms can also be used to smooth traffic between routers, as well as to regulate host output as in our examples. However, one clear difference is that a token bucket regulating a host can make the host stop sending when the rules say it must. Telling a router to stop sending while its input keeps pouring in may result in lost data.

Q. 3. (b). (i) what do you understand by internetworking ? Discuss the parameters on which the networks differs.

Ans. We believe that a variety of different networks (and thus protocols) will always be around, for the following reasons. First of all, the installed base of different networks is large. Nearly all 317 personal computers run TCP/IP. Many large businesses have mainframes running IBM's SNA. A substantial number of telephone companies operate ATM networks. Some personal computer LANs still use Novell NCP/IPX or AppleTalk. Finally, wireless is an up-and-coming area with a variety of protocols. Th:s trend will continue for years due to legacy problems, new technology, and the fact that not all vendors perceive it in their interest for their customers to be able to easily migrate to another vendor's system.

A collection of interconnected networks.



The purpose of interconnecting all these networks is to allow users on any of them to communicate with users on all the other ones and also to allow users on any of them to access data on any of them. Accomplishing this goal means sending packets from one network to another

Some of the many ways networks can differ.

Item	Some Possibilities
Service offered	Connection oriented versus connectionless
Protocols	IP, IPX, SNA, ATM, MPLS, Apple Talk, etc.
Addressing	Flat (802) versus hierarchical (IP)
Multicasting	Present or absent (also broadcasting)
Packet size	Every network has its own maximum
Quality of service	Present or absent; many differetn kinds
Error handling	Reliable, ordered, and unodered delivery
Flow control	Sliding window, rate control, other, or none
Congestion Control	Leaky bucket, token bucke, RED, choke packets, etc.
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, by packet, by byte, or not at all

Q. 3. (b). (ii) If fragmentation needed in concatenated virtual circuit internets, or only in datagram systems ? Explain.

Ans. Fragmentation is needed in both. Even in a concatenated virtual-circuit network, some networks along the path might accept 1024-byte packets, and others might only accept 48-byte packets. Fragmentation is still needed.

Issue	Datagram subnet	Virtual-circuit subnet
Circuit	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State inormation	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are erminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficutl	Easy if enough resources can be allocated in advance for each VC

Q. 3. (c). What are the defeciencies of IPv4 ? How IPv6 was modified to overcome these defeciencies ? What are the advantages of using IPv6 ?

Ans. IPv6

Its major goals were:

1. Support billions of hosts, even with inefficient address space allocation.
2. Reduce the size of the routing tables.
3. Simplify the protocol, to allow routers to process packets faster.
4. Provide better security (authentication and privacy) than current IP.
5. Pay more attention to type of service, particularly for real-time data.
6. Aid multicasting by allowing scopes to be specified.
7. Make it possible for a host to roam without changing its address.
8. Allow the protocol to evolve in the future.
9. Permit the old and new protocols to coexist for years.

IPv6 maintains the good features of IP, discards or deemphasizes the bad ones, and adds new ones where needed. In general, IPv6 is not compatible with IPv4, but it is compatible with the other auxiliary Internet protocols, including TCP, UDP, ICMP,

IGMP, OSPF, BGP, and DNS, sometimes with small modifications being required (mostly to deal with longer addresses). The main features of IPv6 are discussed below.

First and foremost, IPv6 has longer addresses than IPv4. They are 16 bytes long, which solves the problem that IPv6 set out to solve: provide an effectively unlimited supply of Internet addresses. We will have more to say about addresses shortly.

The second major improvement of IPv6 is the simplification of the header. It contains only seven fields (versus 13 in IPv4). This change allows routers to process packets faster and thus improve throughput and delay. We will discuss the header shortly, too.

The third major improvement was better support for options. This change was essential with the new header because fields that previously were required are now optional. In addition, the way options are represented is different, making it simple for routers to skip over options not intended for them. This feature speeds up packet processing time.

A fourth area in which IPv6 represents a big advance is in security. IETF had its fill of

newspaper stories about precocious 12-year-olds using their personal computers to break into banks and military bases all over the Internet. There was a strong feeling that something had to be done to improve security. Authentication and privacy are key features of the new IP. These were later retrofitted to IPv4, however, so in the area of security the differences are not so great any more.

Finally, more attention has been paid to quality of service. Various half-hearted efforts have been made in the past, but now with the growth of multimedia on the Internet, the sense of urgency is greater.

Q. 4. Attempt any two parts of the following :

10×2 = 20

Q. 4. (a) Discuss the transport service primitives. What do you understand by the term : "three way handshake"? Explain the problem which is solved by this three way hand shake.

Ans. Transport Service Primitives

To allow users to access the transport service, the transport layer must provide some operations to application programs, that is, a transport service interface. Each transport service has its own interface.

The transport service is similar to the network service, but there are also some important differences. The main difference is that the network service is intended to model the service offered by real networks, warts and all. Real networks can lose packets, so the network service is generally unreliable.

The (connection-oriented) transport service, in contrast, is reliable. Of course, real networks are not error-free, but that is precisely the purpose of the transport layer-to provide a reliable service on top of an unreliable network.

As an example, consider two processes connected by pipes in UNIX. They assume the connection between them is perfect. They do not want to know about acknowledgements, lost packets, congestion, or anything like that. What they want is a 100 percent reliable connection. Process A puts data into one end of the pipe, and process B takes it out of the other. This is what the connection-oriented transport service is all about-hiding the imperfections of the network service so that user processes can just assume the existence of an error-free bit stream.

Primitive	Packet sent	Meaning
LISTEN	(none)	Block untill some process tries to connect
CONNECT	CONNCETION REQ.	Actively attemp to establisha conncetion
SEND	DATA	Send information
RECEIVE	(none)	Block untill a DATA packet arrives
DISCONNECT	DISCONNECTION REQ.	This side wants to release the connection

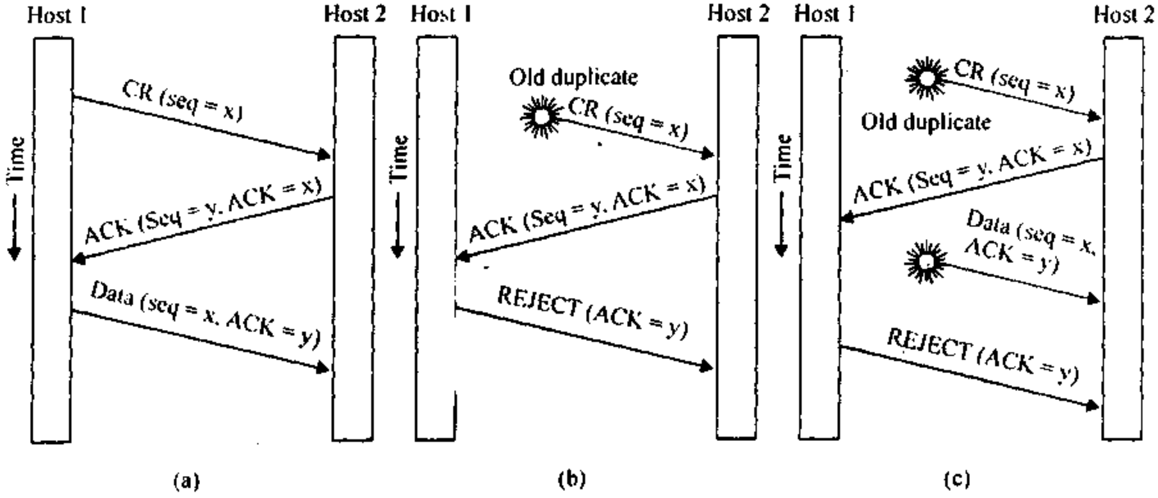
To see how these primitives might be used, consider an application with a server and a number of remote clients. To start with, the server executes a LISTEN primitive, typically by calling a library procedure that makes a system call to block the server until a client turns up. When a client wants to talk to the server, it executes a CONNECT primitive. The transport entity carries out this primitive by blocking the caller and sending a packet to the server. Encapsulated in the payload of this packet is a transport layer message for the server's transport entity.

Three-way handshake

the three-way handshake works in the presence of delayed duplicate control TPDUs. In Fig, the first TPDU is a delayed duplicate CONNECTION REQUEST from an old connection. This TPDU arrives at host 2 without host 1's knowledge. Host 2 reacts to this TPDU by sending host 1 an ACK TPDU, in effect asking for verification that host 1 was indeed trying to set up a new connection. When host 1 rejects host 2's attempt to establish a connection, host 2 realizes that it was tricked by a delayed duplicate and abandons the connection. In this way, a delayed duplicate does no damage.

The worst case is when both a delayed CONNECTION REQUEST and an ACK are floating around in the subnet. This case is shown in Fig. (c). As in the previous example, host 2 gets a delayed CONNECTION REQUEST and replies to it. At this point it is crucial to realize that host 2 has proposed using y as the initial sequence number for host 2 to host 1 traffic, knowing full well that no TPDUs containing sequence number y or acknowledgements to y are still in existence. When the second delayed TPDU arrives at host 2, the fact that z has been acknowledged rather than y tells host 2 that this, too, is an old duplicate. The important thing to realize here is that there is no combination of old TPDUs that can cause the protocol to fail

and have a connection set up by accident when no one wants it.



Q. 4. (b) Explain the TCP segment header. Also discuss the TCP connection management.

Ans. The TCP Segment Header : Figure 6-29 shows the layout of a TCP segment. Every segment begins with a fixed-format, 20-byte header. The fixed header may be followed by header

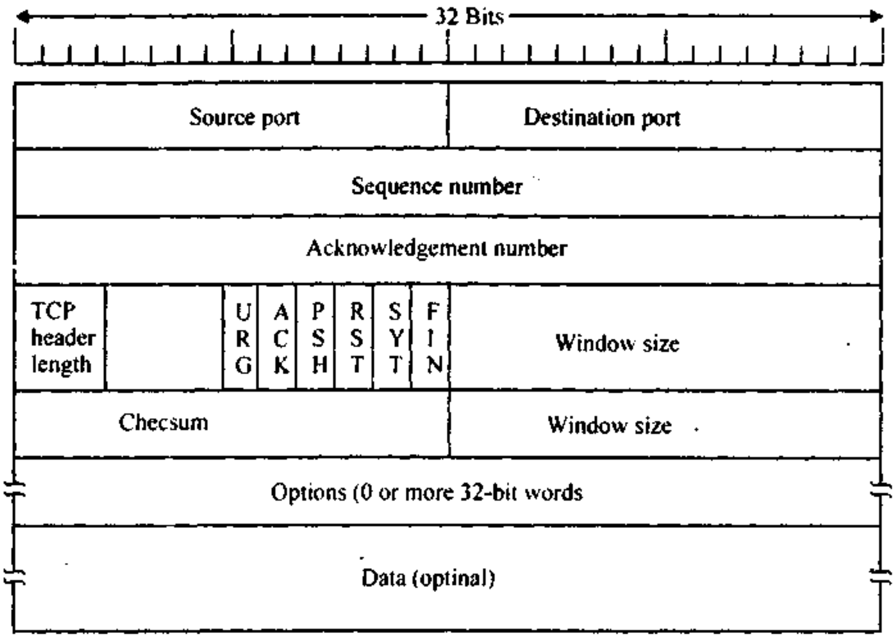


Fig. The TCP header

options. After the options, if any, up to 65,535 - 20 - 20 = 65,495 data bytes may follow, where the first 20 refer to the IP header and the second to the TCP header. Segments without any data are legal and are commonly used for acknowledgements and control messages.

Let us dissect the TCP header field by field. The Source port and Destination port fields identify the local end points of the connection. A port plus its host's IP address forms a 48-bit unique end point. The source and destination end points together identify the connection.

The Sequence number and Acknowledgement number fields perform their usual functions. Note that the latter specifies the next byte expected, not the last byte correctly received. Both are 32 bits long because every byte of data is numbered in a TCP stream.

The TCP header length tells how many 32-bit words are contained in the TCP header. This information is needed because the Options field is of variable length, so the header is, too.

Next comes a 6-bit field that :s not used. The fact that this field has survived intact for over a quarter of a century is testimony to how well thought out TCP is. Lesser protocols would have needed it to fix bugs in the original design.

Now come six 1-bit flags. URG is set to 1 if the Urgent pointer is in use. The Urgent pointer is used to indicate a byte offset from the current sequence number at which urgent data are to be found. This facility is in lieu of interrupt messages. As we mentioned above, this facility is a bare-bones way of allowing the sender to signal the receiver without getting TCP itself involved in the reason for the interrupt.

The ACK bit is set to 1 to indicate that the Acknowledgement number is valid. If ACK is 0, the segment does not contain an acknowledgement so the Acknowledgement number field is ignored.

The PSH bit indicates PUSHed data. The receiver is hereby kindly requested to deliver the data to the application upon arrival and not buffer it until a full buffer has been received (which it might otherwise do for efficiency).

The RST bit is used to reset a connection that has become confused due to a host crash or some other reason. It is also used to reject an invalid segment or refuse an attempt to open a connection. In general, if you get a segment with the RST bit on, you have a problem on your hands.

The SYN bit is used to establish connections. The connection request has SYN = 1 and ACK = 0 to indicate that the piggyback acknowledgement field is not in use. The connection reply does bear an acknowledgement, so it has SYN = 1 and ACK = 1. In essence the SYN bit is used to denote CONNECTION REQUEST and CONNECTION ACCEPTED, with the ACK bit used to distinguish between those twc possibilities.

The FIN bit is used to release a connection. It specifies that the sender has no more data to transmit. However, after closing a connection, the closing process may continue to receive data indefinitely. Both SYN and FIN segments have sequence numbers and are thus guaranteed to be processed in the correct order.

Flow control in TCP is handled using a variable-sized sliding window. The Window size field tells how many bytes may be sent starting at the byte acknowledged. A Window size field of 0 is legal and says that the bytes up to and including Acknowledgement number - 1 have been received, but that the receiver is currently badly in need of a rest and would like no more data for the moment, thank you. The receiver can later grant permission to send by transmitting a segment with the same Acknowledgement number and a nonzero Window size field.

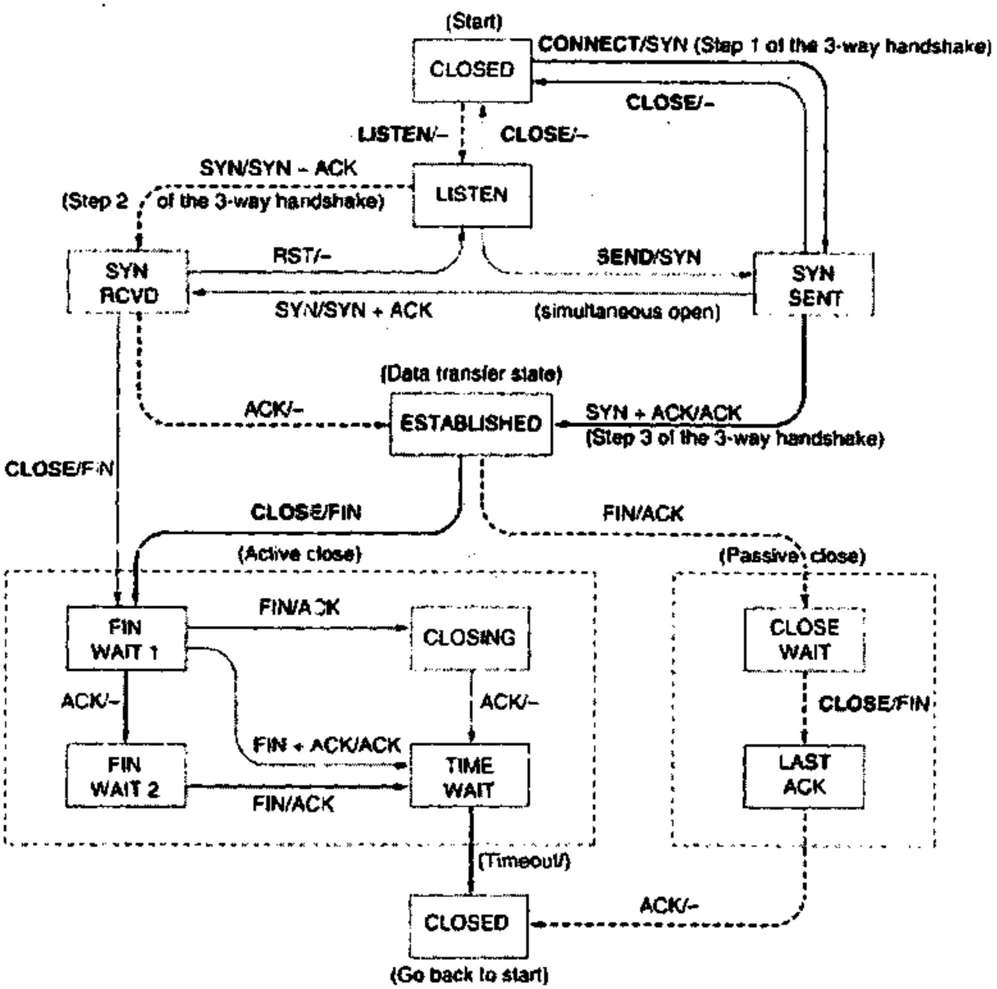
A Checksum is also provided for extra reliability. It checksums the header, the data, and the conceptual pseudoheader. When performing this computation, the TCP

Checksum field is set to zero and the data field is padded out with an additional zero byte if its length is an odd number. The checksum algorithm is simply to add up all the 16-bit words in one's complement and then to take the one's complement of the sum. As a consequence, when the receiver performs the calculation on the entire segment, including the Checksum field, the result should be 0.

TCP Connection Management Modeling

State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incming call
SYN RCVD	A connection request has arrived; wait for ACK
SYN SENT	The application has started to open a connection
ESTABLISHED	The normal data transfer state
FIN WAIT 1	The application has said it is finished
FIN WAIT 2	The other side has said it is finished
TIMED WAIT	Wait for all packets to die off
CLOSING	Both sides have tried to close simultaneously
CLOSE WAIT	The other side has initiated a release
LAS ACK	Wait for all packets to die off

TCP connection management finite state machine. The heavy solid line is the normal path for a client. The heavy dashed line is the normal path for a server. The light lines are unusual events. Each transition is labeled by the event causing it and the action resulting from it, separated by a slash.



Q4. (c). (i) Explain the protocol of transport layer designed for multimedia application

Ans. We are interested in a class of distributed multimedia applications called Cluster-to-Cluster (C-to-C) applications. In a C-to-C application, a collection of computing and communication devices communicates with a remote collection of computing and communication devices. Endpoints in the same cluster share a common gateway node known as the Aggregation Point (AP). While few application flows share the same end-to-end path, all flows share a common intermediary path between clusters. Figure 1 illustrates this model. C-to-C applications share several key characteristics.

Independent, but semantically related flows of data. An application may prioritize streams in a particular way, or divide complex media objects into multiple streams with specific temporal or spatial relationships. Transport-level heterogeneity. UDP- or RTP-based protocols, for example, might be used for streaming media while TCP is used to reliably transport control data.

Changing network conditions along the shared data path. While networks within a cluster can be provisioned to comfortably support an application's requirements, the forwarding path between clusters is shared with other Internet flows and typically cannot be provisioned end-to-end. Hence, it is the primary source of network latency and packet loss.

One example of a C-to-C application is office of the Future, conceived by Fuchs et al.[5] In this immersive application, a set of video cameras and microphones are used to capture an office environment in order to construct a remote virtual environment using various computing and display devices. Complex spatial relationships exist between media streams, and the relative priority of these streams changes constantly as the user moves their region of interest.

A fundamental problem in the C-to-C application context is that of flow coordination. Application streams share a common intermediary path between clusters, and yet employ transport protocols that operate in isolation from one another. As a result, flows may compete with one another when network resources become limited instead of cooperating to use available bandwidth in application-controlled ways.

COORDINATION PROTOCOL : Our solution to the problem of flow coordination in C-to-C applications is called the Coordination Protocol, or CP. Cooperates between the network layer (IP) and transport layer (TCP, UDP, etc.), making it transparent to IP routers on the C-to-C forwarding path and preserving the semantics of end-to-end transport-level protocols. Application endpoints insert a CP header into each data packet before sending the packet to an endpoint on the remote cluster. The packet receives special handling at its local AP, and then is forwarded toward the remote cluster. After traversing the cluster-to-cluster data path using standard IP forwarding, the remote AP applies special handling to the packet before forwarding it to the destination endpoint. Endpoints use information in CP packet headers to make send rate adjustments that react application coordination strategies. As a packet originates from an application endpoint, its CP header contains a cluster ID telling the AP to associate it with a particular C-to-C application. A flow ID likewise identifies the packet with a particular application flow. The AP keeps a table of bandwidth usage statistics on flows in the same C-to- C application, also tracking the number of flows and aggregate bandwidth usage by the C-to-C application as a whole. Network probing works by using the CP header in each C- to-C data packet to piggyback probe information on the shared data path between APs. Each AP modifies the CP header from packets originating at its local cluster to add timestamp and sequence number information. As additional probe information from the remote AP is returned along the reverse cluster-to- cluster data path, the AP is able to obtain measurements of round trip time and loss rates across all aggregate C-to-C traffic. Using round trip time, loss rate, and packet size information, a bandwidth availability estimate can be made at each AP using a throughput modeling equation. Our work has made use of the TFRC [2] equation, giving a throughput estimate that is both gradually responsive to network congestion and TCP- compatible. In addition, we have developed techniques to extend single-flow modeling equations to multiple flowshares, thus allowing m C-to-C application flows to receive the equivalent of m TCP-compatible flowshares.

Q. 4. (c). (ii) What is the procedure for compressing data using run-length encoding ?

Ans. Run-length encoding (RLE) is a very simple form of data compression in which runs of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run. This is most useful on data that contains many such runs: for example, relatively simple graphic images such as icons, line drawings, and animations. It is not useful with files that don't have many runs as it could potentially double the file size.

For example, consider a screen containing plain black text on a solid white background. There will be many long runs of white pixels in the blank space, and many short runs of black pixels within the text. Let us take a hypothetical single scan line, with B representing a black pixel and W representing white:

WWWWWWWWWWWWBWWWWWWWWWWWWBBBWWWWWWWWWWWWWWW
WWWWWWWWWBWWWWWWWWWWWWWW

If we apply the run-length encoding (RLE) data compression algorithm to the above hypothetical scan line, we get the following:

12W1B12W3B24W1B14W

Interpret this as twelve W's, one B, twelve W's, three B's, etc.

The run-length code represents the original 67 characters in only 18. Of course, the actual format used for the storage of images is generally binary rather than ASCII characters like this, but the principle remains the same. Even binary data files can be compressed with this method; file format specifications often dictate repeated bytes in files as padding space. However, newer compression methods such as DEFLATE often use LZ77-based algorithms, a generalization of run-length encoding that can take advantage of runs of strings of characters (such as BWWBWWBWWBWW).

Applications

Run-length encoding performs lossless data compression and is well suited to palette-based iconic images. It does not work well at all on continuous-tone images such as photographs, although JPEG uses it quite effectively on the coefficients that remain after transforming and quantizing image blocks. Common formats for run-length encoded data include True vision TGA, PackBits, PCX and ILBM.Run-length encoding is used in fax machines (combined with other techniques into Modified Huffman coding). It is relatively efficient because most faxed documents are mostly white space, with occasional interruptions of black. Data that have long sequential runs of bytes (such as lower-quality sound samples) can be RLE compressed after applying a predictive filter such as delta encoding.

Q. 5. Attempt any two parts.

Q. 5. (a) Explain Simple Network Management Protocol. List its various components and briefly discuss each of them.

Ans. Simple Network Management Protocol (SNMP) is a UDP-based network protocol. It is used mostly in network management systems to monitor network-attached devices for conditions that warrant administrative attention. SNMP is a component of the Internet Protocol Suite as defined by the Internet Engineering Task Force (IETF). It consists of a set of standards for network management, including an application layer protocol, a database schema, and a set of data objects.

SNMP exposes management data in the form of variables on the managed systems, which describe the system configuration. These variables can then be queried (and sometimes set) by managing applications.

In typical SNMP use, one or more administrative computers have the task of monitoring or managing a group of hosts or devices on a computer network. Each managed system (also called Slave) executes, at all times, a software component called an agent (see below) which reports information via SNMP to the managing systems (also called Masters). The variables accessible via SNMP are organized in hierarchies. These hierarchies, and other metadata (such as type and description of the variable), are described by Management Information Bases (MIBs).

An SNMP-managed network consists of three key components:

Managed device = Slave device

Agent = software which runs on Slave device

Network management system (NMS) = software which runs on Master

A managed device is a network node that implements an SNMP interface that allows unidirectional (read-only) or bidirectional access to node-specific information. Managed devices exchange node-specific information with the NMSs. Sometimes called network elements, the managed devices can be any type of device, including, but not limited to, routers, access servers, switches, bridges, hubs, IP telephones, computer hosts, and printers.

An agent is a network-management software module that resides on a managed device. An agent has local knowledge of management information and translates that information to or from an SNMP specific form.

A network management system (NMS) executes applications that monitor and control managed devices. NMSs provide the bulk of the processing and memory resources required for network management. One or more NMSs may exist on any managed network.

Q. 5. (b). (i) When the web pages are sent out, they are prefixed by MIME headers. Why ?

Ans. Multipurpose Internet Mail Extensions (MIME) is an Internet standard that extends the format of e-mail to support:

- Text in character sets other than ASCII
- Non-text attachments
- Message bodies with multiple parts
- Header information in non-ASCII character sets

MIME's use, however, has grown beyond describing the content of e-mail to describing content type in general, including for the web.

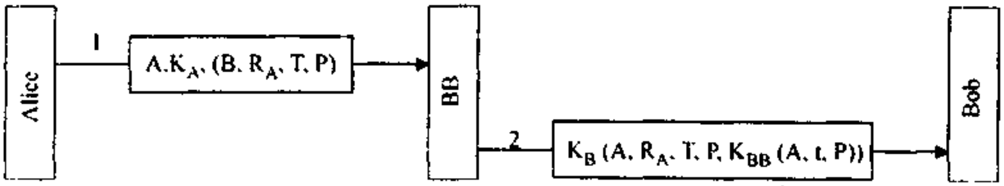
Virtually all human-written Internet e-mail and a fairly large proportion of automated e-mail is transmitted via SMTP in MIME format. Internet e-mail is so closely associated with the SMTP and MIME standards that it is sometimes called SMTP/MIME e-mail.

The content types defined by MIME standards are also of importance outside of e-mail, such as in communication protocols like HTTP for the World Wide Web. HTTP requires that data be transmitted in the context of e-mail-like messages, although the data most often is not actually e-mail.

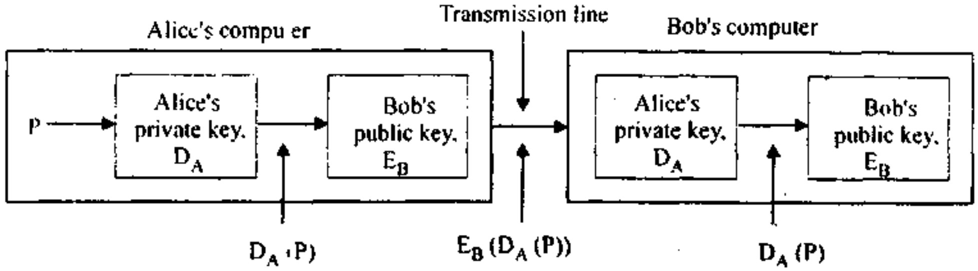
MIME defines mechanisms for sending other kinds of information in e-mail. These include text in languages other than English using character encodings other than ASCII, and 8-bit binary content such as files containing images, sounds, movies, and computer programs. MIME is also a fundamental component of communication protocols such as HTTP, which requires that data be transmitted in the context of e-mail-like messages even though the data might not (and usually doesn't) actually have anything to do with e-mail. Mapping messages into and out of MIME format is typically done automatically by an e-mail client or by mail servers when sending or receiving Internet (SMTP/MIME) e-mail. The goals of the MIME definition included requiring no changes to existent e-mail servers and allowing plain text e-mail to function in both directions with existing clients.

Q. 5. (b). (ii) Explain the working of digital signature.

Ans. One approach to digital signatures is to have a central authority that knows everything and whom everyone trusts, say Big Brother (BB). Each user then chooses a secret key and carries it by hand to BB's office. Thus, only Alice and BB know Alice's secret key, KA, and so on.



Public-key cryptography can make an important contribution in this area. Let us assume that the public-key encryption and decryption algorithms have the property that *E(D(P)) = P* in addition, of course, to the usual property that *D(E(P)) = P.* (RSA has this property, so the assumption is not unreasonable.) Assuming that this is the case, Alice can send a signed plaintext message, *P*, to Bob by transmitting *EB(DA(P))*. Note carefully that Alice knows her own (private) key, *DA*, as well as Bob's public key, EB, so constructing this message is something Alice can do. When Bob receives the message, he transforms it using his private key, as usual, yielding *DA(P)*, as shown in Fig. He stores this text in a safe place and then applies *EA* to get the original plaintext.



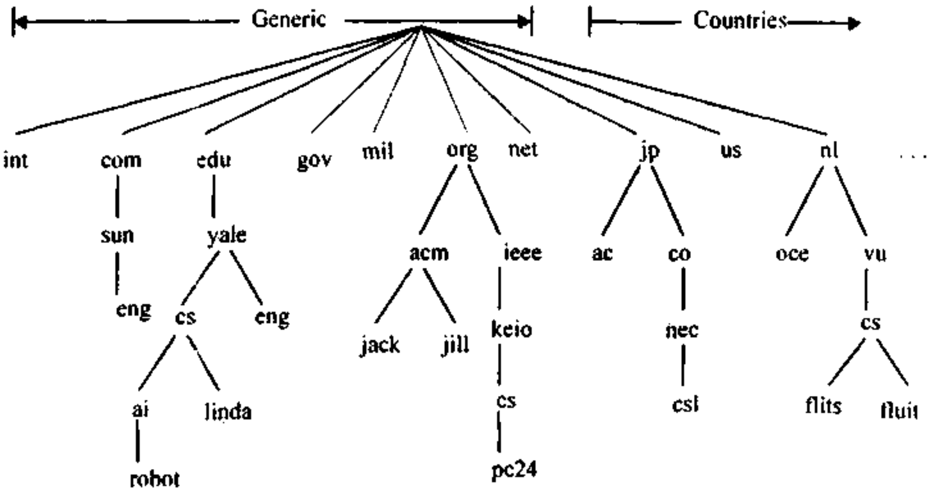
Q. 5. (c). Write short notes on any two

Q. 5. (c) (i). DNS

Ans. The essence of DNS is the invention of a hierarchical, domain-based naming scheme and a distributed database system for implementing this naming scheme. It is primarily used for mapping host names and e-mail destinations to IP addresses but can also be used for other purposes. DNS is defined in RFCs 1034 and 1035.

The DNS Name Space

The Internet is divided into over 200 top-level domains, where each domain covers many hosts. Each domain is partitioned into sub domains, and these are further partitioned, and so on. All these domains can be represented by a



tree, as shown in Fig. . The leaves of the tree represent domains that have no sub domains (but do contain machines, of course). A leaf domain may contain a single host, or it may represent a company and contain thousands of hosts.

A portion of the Internet domain name space

The top-level domains come in two flavors: generic and countries. The original generic domains were com (commercial), edu (educational institutions), gov (the U.S. Federal Government), int (certain international organizations), mil (the U.S. armed forces), net (network providers), and org (nonprofit organizations). The country domains include one entry for every country, as defined in ISO 3166. Each domain controls how it allocates the domains under it. For example, Japan has domains ac.jp and co.jp that mirror edu and com. The Netherlands does not make this distinction and puts all organizations directly under nl. Thus, all three of the following are university computer science departments:

1. cs.yale.edu (Yale University, in the United States)
2. cs.vu.nl (Vrije Universities, in The Netherlands)
3. cs.keio.ac.jp (Keio University, in Japan)

To create a new domain, permission is required of the domain in which it will be included. For example, if a VLSI group is started at Yale and wants to be known as vlsi.cs.yale.edu, it has to get permission from whoever manages cs.yale.edu.

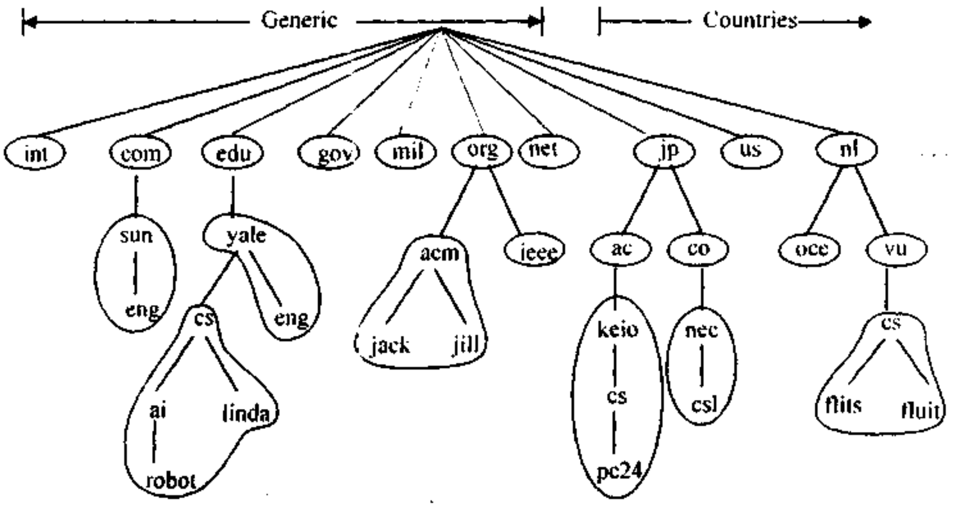
Resource Records : Every domain, whether it is a single host or a top-level domain, can have a set of resource records associated with it. For a single host, the most common resource record is just its IP address, but many other kinds of resource records also exist. When a resolver gives a domain name to DNS, what it gets back are the resource records associated with that name. Thus, the primary function of DNS is to map domain names onto resource records. A resource record is a five-tuple. Although they are encoded in binary for efficiency, in most expositions, resource records are presented as ASCII text, one line per resource record. The format we will use is as follows:

Domain_name Time_to_live Class Type Value

The Domain_name tells the domain to which this record applies.

The Time_to_live field gives an indication of how stable the record is. Information that is highly stable is assigned a large value, such as 86400 (the number of seconds in 1 day). Information that is highly volatile is assigned a small value, such as 60 (1 minute).

Name Servers



Q. 5. (c) (ii). Vertical terminal

Ans. A terminal devices synchronizing method for synchronizing a plurality of terminal devices interconnected through a network, Each of said plurality of respective terminal devices comprising vertical synchronizing signal generating means for generating vertical synchronizing signals, and control means for making synchronization control operations and data communication, based on the vertical synchronizing signals, respectively, wherein the vertical synchronizing signal generating means comprises a vertical synchronizing counter, a horizontal synchronizing counter, and a reset circuit for resetting both the vertical synchronizing counter and the horizontal synchronizing counter, the respective terminal devices extract a synchronizing signal from signals wirelessly inputted from the outside other than the respective terminal devices, when the synchronizing signal is extracted, the reset circuit of the respective terminal devices reset both the vertical synchronizing counter and the horizontal synchronizing counter in synchronization with the synchronizing signal, and the vertical synchronizing signal generating means of the respective terminal devices output the synchronizing signal as a vertical synchronizing signal, when the synchronizing signal is not extracted, the vertical synchronizing signal generating means of the respective terminal devices output a back-up vertical synchronizing signal, and the control means of the respective terminal devices make synchronization control operation and data communication, based on the vertical synchronizing signal or the back-up vertical synchronizing signal.

Q. 5. (c) (iii) USENET

Ans. Usenet is one of the oldest computer network communications systems still in widespread use. It was conceived in 1979 and publicly established in 1980 at the University of North Carolina at Chapel Hill and Duke University, over a decade before the World Wide Web was developed and the general public got access to the Internet. It was originally built on the "poor man's ARPANET," employing UUCP as its transport protocol to offer mail and file transfers, as well as announcements through the newly developed news software. The name USENET emphasized its creators' hope that the USENIX organization would take an active role in its operation

When a user posts an article, it is initially only available on that user's news server. Each news server, however, talks to one or more other servers (its "news feeds") and exchanges articles with them. In this fashion, the article is copied from server to server and (if all goes well) eventually reaches every server in the network. The later peer-to-peer networks operate on a similar principle; but for Usenet it is normally the sender, rather than the receiver, who initiates transfers. Some have noted that this seems an inefficient protocol in the era of abundant high-speed network access. Usenet was designed for a time when networks were much slower, and not always available. Many sites on the original Usenet network would connect only once or twice a day to batch-transfer messages in and out.

Usenet has significant cultural importance in the networked world, having given rise to, or popularized, many widely recognized concepts and terms such as "FAQ" and "spam".

The format and transmission of Usenet articles is similar to that of Internet e-mail messages. The difference between the two is that Usenet articles can be read by any user whose news server carries the group to which the message was posted, as opposed to email messages which have one or more specific recipients.