B.E-COMPUTER SCIENCE AND ENGINEERING

13BECSE04 TCP / IP DESIGN AND IMPLEMENTATION 3H-3C

Instruction Hours/week: L:3 T:0 P:0 **Marks:** Internal**:40** External**:60** Total**:100**

End Semester Exam:3 Hours

COURSE OBJECTIVES:

- To understand the IP addressing schemes.
- To learn the fundamentals of network design and implementation
- To understand the design and implementation of TCP/IP networks
- To learn the network management issues
- To understand the design and implement network applications.

COURSE OUTCOME:

Upon completion of this course, the students will be able to:

- Design and implement TCP/IP networks.
- Explain network management issues.
- Develop data structures for basic protocol functions of TCP/IP.
- Apply the members in the respective structures.
- Design and implement data structures for maintaining multiple local and global timers.

UNIT- I INTRODUCTION (9)

Internetworking concepts and architectural model- classful Internet address – CIDR-Subnetting and Supernetting –ARP- RARP- IP – IP Routing –ICMP – Ipv6

UNIT- II TCP (9)

Services – header – connection establishment and termination- interactive data flow- bulk data flow-timeout and retransmission – persist timer - keepalive timer- futures and performance

UNIT- III IP IMPLEMENTATION (9)

IP global software organization – routing table- routing algorithms-fragmentation and reassembly-error processing (ICMP) –Multicast Processing (IGMP)

UNIT- IV TCP IMPLEMENTATION I (9)

Data structure and input processing – transmission control blocks- segment format- comparison-finite state machine implementation-Output processing- mutual exclusion-computing the TCP data length.

UNIT- V TCP IMPLEMENTATION II (9)

Timers-events and messages- timer process- deleting and inserting timer event- flow control and adaptive retransmission-congestion avoidance and control – urgent data processing and push function.

Total Hours:45

TEXT BOOKS:

1. Douglas E.Comer, "Internetworking with TCP/IP Principles Protocols and Architecture ",(4th edition), Pearson Education Asia, 2006.
2. W.Richard Stevens, "TCP/IP Illustrated", Vol 1. Pearson Education, 2003.

REFERENCES:

1. Forouzan, " TCP/IP Protocol Suite" Second Edition, Tata MC Graw Hill, 2003.
2. W.Richard Stevens, "TCP/IP illustrated", Vol 2. Pearson Education, 2003

WEBSITES:

1. https://nptel.ac.in/courses/106105081/
2. https://nptel.ac.in/content/storage2/nptel_data3/html/mhrd/ict/text/106105183/lec45.pdf
3. https://link.springer.com/chapter/10.1007/978-3-642-14533-9_5

KARPAGAM ACADEMY OF HIGHER EDUCATION

Department of Computer Science and Engineering

Faculty of Engineering

Lecture Plan

Subject Name: TCP / IP DESIGN AND IMPLEMENTATION **Subject Code: 13BECSE04**

S.No	Topic Name	No.of Periods	Supporting Materials	Teaching Aids
	UNIT- I INTRODUCTION			
1	Internetworking concepts and architectural model	1	T[1]-1	BB
2	classful Internet address	1	T[1]-8	BB
3	CIDR	1	R[1]-5	PPT
4	Subnetting and Supernetting	2	R[1]-6	PPT
5	ARP	1	T[1]-16	PPT
6	RARP	1	R[1]-45	PPT
7	IP	1	R[1]-60	BB
8	IP Routing	1	T[1]-35	PPT
9	ICMP	1	R[1]-90	PPT
10	Ipv6	1	R[1]-92	PPT
Total		**11**		
	UNIT- II TCP			
11	Services	1	R[1]-156	PPT
12	header	1	Web	PPT
13	connection establishment and termination	1	R[1] 201	BB
14	interactive data flow	1	R[2]101	PPT
15	bulk data flow	1	R[1]214	PPT
16	timeout and retransmission	2	R[2]135	PPT
17	persist timer	1	R[1]218	PPT
18	keepalive timer	1	T[2]-5	BB
19	futures and performance	1	T[2]-25	PPT

Total		**10**		

	UNIT- III IP IMPLEMENTATION			
14	IP global software organization	1	Web	PPT
15	routing table	1	Web	PPT
16	routing algorithms	1	Web	PPT
17	fragmentation and reassembly	2	T[1]-245	BB
18	error processing (ICMP)	2	T[1]-193	PPT
19	Multicast Processing (IGMP)	1	T[2]-205	BB
Total		**8**		
	UNIT- IV TCP IMPLEMENTATION I			
24	Data structure and input processing	1	R[1]-140	PPT
25	transmission control blocks	2	R[1]-160	PPT
26	segment format	1	T[1]-140	PPT
27	Comparison	1	R[1]-162	BB
28	finite state machine implementation	2	R[1]-159	PPT
29	Output processing	1	R[1]-125	BB
30	mutual exclusion	1	R[1]-163	PPT
31	computing the TCP data length	1	R[1]-133	PPT
Total		**10**		
	UNIT- V TCP IMPLEMENTATION II			
34	Timers	1	R[1]-248	PPT
35	events and messages	1	R[1]-465	BB
36	timer process	1	R[1]-465	BB
37	deleting and inserting timer event	1	R[1]-255	PPT
38	flow control and adaptive retransmission	2	R[1]-248	PPT
39	congestion avoidance and control	2	T[1]-1087	PPT
40	urgent data processing and push function.	1	T[1]-1087	PPT
	Discussion on Previous University Question Papers			
Total		**9**		
	Total Hours	**48**		

TEXT BOOKS:

1. Douglas E.Comer, “Internetworking with TCP/IP Principles Protocols and Architecture “, (4th edition), Pearson Education Asia, 2006.
2. W.Richard Stevens, “TCP/IP Illustrated”, Vol 1. Pearson Education, 2003.

REFERENCES:

1. Forouzan, “ TCP/IP Protocol Suite” Second Edition, Tata MC Graw Hill, 2003.
2. W.Richard Stevens, “TCP/IP illustrated”, Vol 2. Pearson Education, 2003

WEBSITES:

1. https://nptel.ac.in/courses/106105081/
2. https://nptel.ac.in/content/storage2/nptel_data3/html/mhrd/ict/text/106105183/lec45.pdf
3. https://link.springer.com/chapter/10.1007/978-3-642-14533-9_5

LECTURE NOTES

UNIT 1

IP Routing

We now take up the question of finding the host that datagrams go to based on the IP address. Different parts of the address are handled in different ways; it is your job to set up the files that indicate how to treat each part.

2.4.1. IP Networks

When you write a letter to someone, you usually put a complete address on the envelope specifying the country, state, and Zip Code. After you put it in the mailbox, the post office will deliver it to its destination: it will be sent to the country indicated, where the national service will dispatch it to the proper state and region. The advantage of this hierarchical scheme is obvious: wherever you post the letter, the local postmaster knows roughly which direction to forward the letter, but the postmaster doesn't care which way the letter will travel once it reaches its country of destination.

IP networks are structured similarly. The whole Internet consists of a number of proper networks, called *autonomous systems*. Each system performs routing between its member hosts internally so that the task of delivering a datagram is reduced to finding a path to the destination host's network. As soon as the datagram is handed to *any* host on that particular network, further processing is done exclusively by the network itself.

2.4.2. Subnetworks

This structure is reflected by splitting IP addresses into a host and network part, as explained previously. By default, the destination network is derived from the network part of the IP address. Thus, hosts with identical IP *network* numbers should be found within the same network.[1]

It makes sense to offer a similar scheme *inside* the network, too, since it may consist of a collection of hundreds of smaller networks, with the smallest units being physical networks like Ethernets. Therefore, IP allows you to subdivide an IP network into several *subnets*.

A subnet takes responsibility for delivering datagrams to a certain range of IP addresses. It is an extension of the concept of splitting bit fields, as in the A, B, and C classes. However, the network part is now extended to include some bits from the host part. The number of bits that are interpreted as the subnet number is given by the so-called *subnet mask*, or *netmask*. This is a 32-bit number too, which specifies the bit mask for the network part of the IP address.

The campus network of Groucho Marx University is an example of such a network. It has a class B network number of 149.76.0.0, and its netmask is therefore 255.255.0.0.

Internally, GMU's campus network consists of several smaller networks, such various departments' LANs. So the range of IP addresses is broken up into 254 subnets, 149.76.1.0 through 149.76.254.0. For example, the department of Theoretical Physics has been assigned 149.76.12.0. The campus backbone is a network in its

own right, and is given 149.76.1.0. These subnets share the same IP network number, while the third octet is used to distinguish between them. They will thus use a subnet mask of 255.255.255.0.

Figure 2-1 shows how 149.76.12.4, the address of quark, is interpreted differently when the address is taken as an ordinary class B network and when used with subnetting.

Figure 2-1. Subnetting a class B network

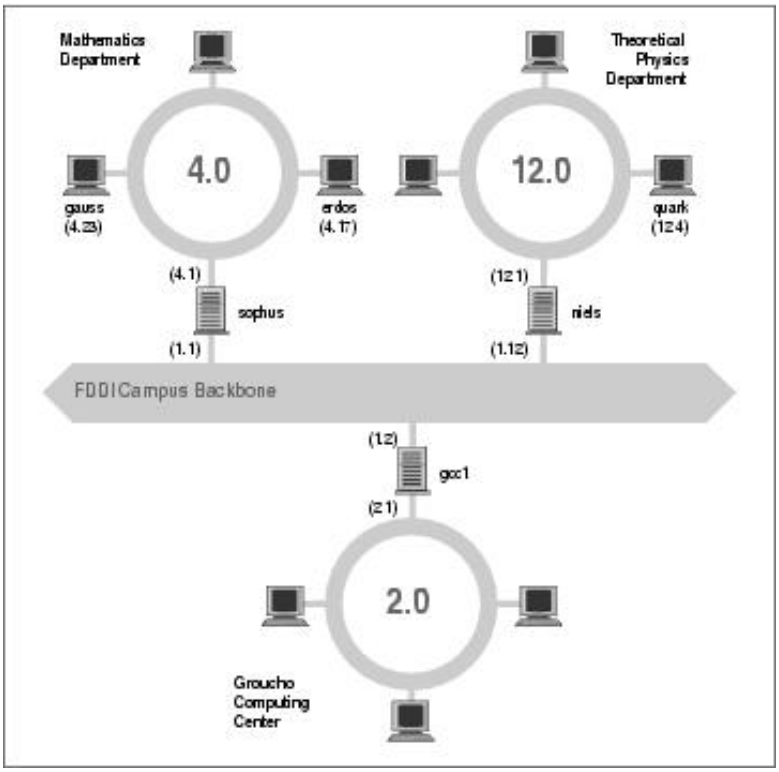
It is worth noting that *subnetting* (the technique of generating subnets) is only an *internal division* of the network. Subnets are generated by the network owner (or the administrators). Frequently, subnets are created to reflect existing boundaries, be they physical (between two Ethernets), administrative (between two departments), or geographical (between two locations), and authority over each subnet is delegated to some contact person. However, this structure affects only the network's internal behavior, and is completely invisible to the outside world.

2.4.3. Gateways

Subnetting is not only a benefit to the organization; it is frequently a natural consequence of hardware boundaries. The viewpoint of a host on a given physical network, such as an Ethernet, is a very limited one: it can only talk to the host of the network it is on. All other hosts can be accessed only through special-purpose machines called *gateways*. A gateway is a host that is connected to two or more physical networks simultaneously and is configured to switch packets between them.

Figure 2-2 shows part of the network topology at Groucho Marx University (GMU). Hosts that are on two subnets at the same time are shown with both addresses.

Figure 2-2. A part of the net topology at Groucho Marx University



Different physical networks have to belong to different IP networks for IP to be able to recognize if a host is on a local network. For example, the network number 149.76.4.0 is reserved for hosts on the mathematics LAN. When sending a datagram to quark, the network software on erdos immediately sees from the IP address 149.76.12.4 that the destination host is on a different physical network, and therefore can be reached only through a gateway (sophus by default).

sophus itself is connected to two distinct subnets: the Mathematics department and the campus backbone. It accesses each through a different interface, `eth0` and `fddi0`, respectively. Now, what IP address do we assign it? Should we give it one on subnet 149.76.1.0, or on 149.76.4.0?

The answer is: "both." sophus has been assigned the address 149.76.1.1 for use on the 149.76.1.0 network and address 149.76.4.1 for use on the 149.76.4.0 network. A gateway must be assigned one IP address for each network it belongs to. These addresses—along with the corresponding netmask—are tied to the interface through which the subnet is accessed. Thus, the interface and address mapping for sophus would look like this:

Interface	Address	Netmask
`eth0`	149.76.4.1	255.255.255.0
`fddi0`	149.76.1.1	255.255.255.0
`lo`	127.0.0.1	255.0.0.0

The last entry describes the loopback interface `lo`, which we talked about earlier.

Generally, you can ignore the subtle difference between attaching an address to a host or its interface. For hosts that are on one network only, like erdos, you would generally refer to the host as having this-and-that IP address, although strictly speaking, it's the Ethernet interface that has this IP address. The distinction is really important only when you refer to a gateway.

2.4.4. The Routing Table

We now focus our attention on how IP chooses a gateway to use to deliver a datagram to a remote network.

We have seen that erdos, when given a datagram for quark, checks the destination address and finds that it is not on the local network. erdos therefore sends the datagram to the default gateway sophus, which is now faced with the same task. sophus recognizes that quark is not on any of the networks it is connected to directly, so it has to find yet another gateway to forward it through. The correct choice would be niels, the gateway to the Physics department. sophus thus needs information to associate a destination network with a suitable gateway.

IP uses a table for this task that associates networks with the gateways by which they may be reached. A catch-all entry (the *default route*) must generally be supplied too; this is the gateway associated with network 0.0.0.0. All destination addresses match this route, since none of the 32 bits are required to match, and therefore packets to an unknown network are sent through the default route. On sophus, the table might look like this:

Network	Netmask	Gateway	Interface
149.76.1.0	255.255.255.0	-	`fddi0`
149.76.2.0	255.255.255.0	149.76.1.2	`fddi0`
149.76.3.0	255.255.255.0	149.76.1.3	`fddi0`
149.76.4.0	255.255.255.0	-	`eth0`
149.76.5.0	255.255.255.0	149.76.1.5	`fddi0`
...	...	...	...
0.0.0.0	0.0.0.0	149.76.1.2	`fddi0`

If you need to use a route to a network that sophus is directly connected to, you don't need a gateway; the gateway column here contains a hyphen.

The process for identifying whether a particular destination address matches a route is a mathematical operation. The process is quite simple, but it requires an understanding of binary arithmetic and logic: A route matches a destination if the network address logically ANDed with the netmask precisely equals the destination address logically ANDed with the netmask.

Translation: a route matches if the number of bits of the network address specified by the netmask (starting from the left-most bit, the high order bit of byte one of the address) match that same number of bits in the destination address.

When the IP implementation is searching for the best route to a destination, it may find a number of routing entries that match the target address. For example, we know that the default route matches every destination, but datagrams destined for locally attached networks will match their local route, too. How does IP know which route to use? It is here that the netmask plays an important role. While both routes match the destination, one of the routes has a larger netmask than the other. We previously mentioned that the netmask was used to break up our address space into smaller networks. The larger a netmask is, the more specifically a target address is matched; when routing datagrams, we should always choose the route that has the largest netmask. The default route has a netmask of zero bits, and in the configuration presented above, the locally attached networks have a

24-bit netmask. If a datagram matches a local attached network, it will be routed to the appropriate device in preference to following the default route because the local network route matches with a greater number of bits. The only datagrams that will be routed via the default route are those that don't match any other route.

You can build routing tables by a variety of means. For small LANs, it is usually most efficient to construct them by hand and feed them to IP using the **route** command at boot time (see Chapter 5). For larger networks, they are built and adjusted at runtime by *routing daemons*; these daemons run on central hosts of the network and exchange routing information to compute "optimal" routes between the member networks.

Depending on the size of the network, you'll need to use different routing protocols. For routing inside autonomous systems (such as the Groucho Marx campus), the *internal routing protocols* are used. The most prominent one of these is the *Routing Information Protocol* (RIP), which is implemented by the BSD **routed** daemon. For routing between autonomous systems, *external routing protocols* like *External Gateway Protocol* (EGP) or *Border Gateway Protocol* (BGP) have to be used; these protocols, including RIP, have been implemented in the University of Cornell's **gated** daemon.

2.4.5. Metric Values

We depend on dynamic routing to choose the best route to a destination host or network based on the number of *hops*. Hops are the gateways a datagram has to pass before reaching a host or network. The shorter a route is, the better RIP rates it. Very long routes with 16 or more hops are regarded as unusable and are discarded.

RIP manages routing information internal to your local network, but you have to run **gated** on all hosts. At boot time, **gated** checks for all active network interfaces. If there is more than one active interface (not counting the loopback interface), it assumes the host is switching packets between several networks and will actively exchange and broadcast routing information. Otherwise, it will only passively receive RIP updates and update the local routing table.

When broadcasting information from the local routing table, **gated** computes the length of the route from the so- called *metric value* associated with the routing table entry. This metric value is set by the system administrator when configuring the route, and should reflect the actual route cost.[2] Therefore, the metric of a route to a subnet that the host is directly connected to should always be zero, while a route going through two gateways should have a metric of two. You don't have to bother with metrics if you don't use **RIP** or **gated**.

Notes

[1] Autonomous systems are slightly more general. They may comprise more than one IP network.

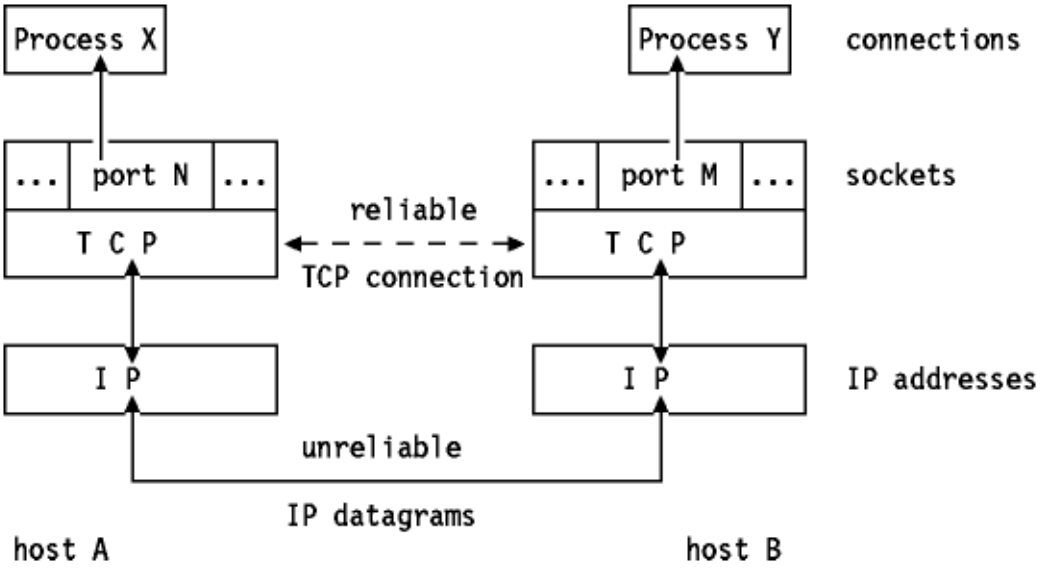
[2] The cost of a route can be thought of, in a simple case, as the number of hops required to reach the destination. Proper calculation of route costs can be a fine art in complex network designs.

UNIT 2

This mini-tutorial provides a general overview of TCP, and is not specific to the SSF implementation.
Prepared by Shweta Sinha 11/98, edited by Andy Ogielski.

TCP SUMMARY

TCP provides a connection oriented, reliable, byte stream service. The term connection-oriented means the two applications using TCP must establish a TCP connection with each other before they can exchange data. It is a full duplex protocol, meaning that each TCP connection supports a pair of byte streams, one flowing in each direction. TCP includes a flow-control mechanism for each of these byte streams that allows the receiver to limit how much data the sender can transmit. TCP also implements a congestion-control mechanism.



Two processes communicating via TCP sockets. Each side of a TCP connection has a socket which can be identified by the pair < *IP_address, port_number* >. Two processes communicating over TCP form a logical connection that is uniquely identifiable by the two sockets involved, that is by the combination < *local_IP_address, local_port, remote_IP_address, remote_port*>.

TCP provides the following facilities to:

Stream Data Transfer

From the application's viewpoint, TCP transfers a contiguous stream of bytes. TCP does this by grouping the bytes in TCP segments, which are passed to IP for transmission to the destination. TCP itself decides how to segment the data and it may forward the data at its own convenience.

Reliability

TCP assigns a sequence number to each byte transmitted, and expects a positive acknowledgment (ACK) from the receiving TCP. If the ACK is not received within a timeout interval, the data is retransmitted. The receiving TCP uses the sequence numbers to rearrange the segments when they arrive out of order, and to eliminate duplicate segments.

Flow Control

The receiving TCP, when sending an ACK back to the sender, also indicates to the sender the number of bytes it can receive beyond the last received TCP segment, without causing overrun and overflow in its internal buffers. This is sent in the ACK in the form of the highest sequence number it can receive without problems.

Multiplexing

To allow for many processes within a single host to use TCP communication facilities simultaneously, the TCP provides a set of addresses or ports within each host. Concatenated with the network and host addresses from the internet communication layer, this forms a socket. A pair of sockets uniquely identifies each connection.

Logical Connections

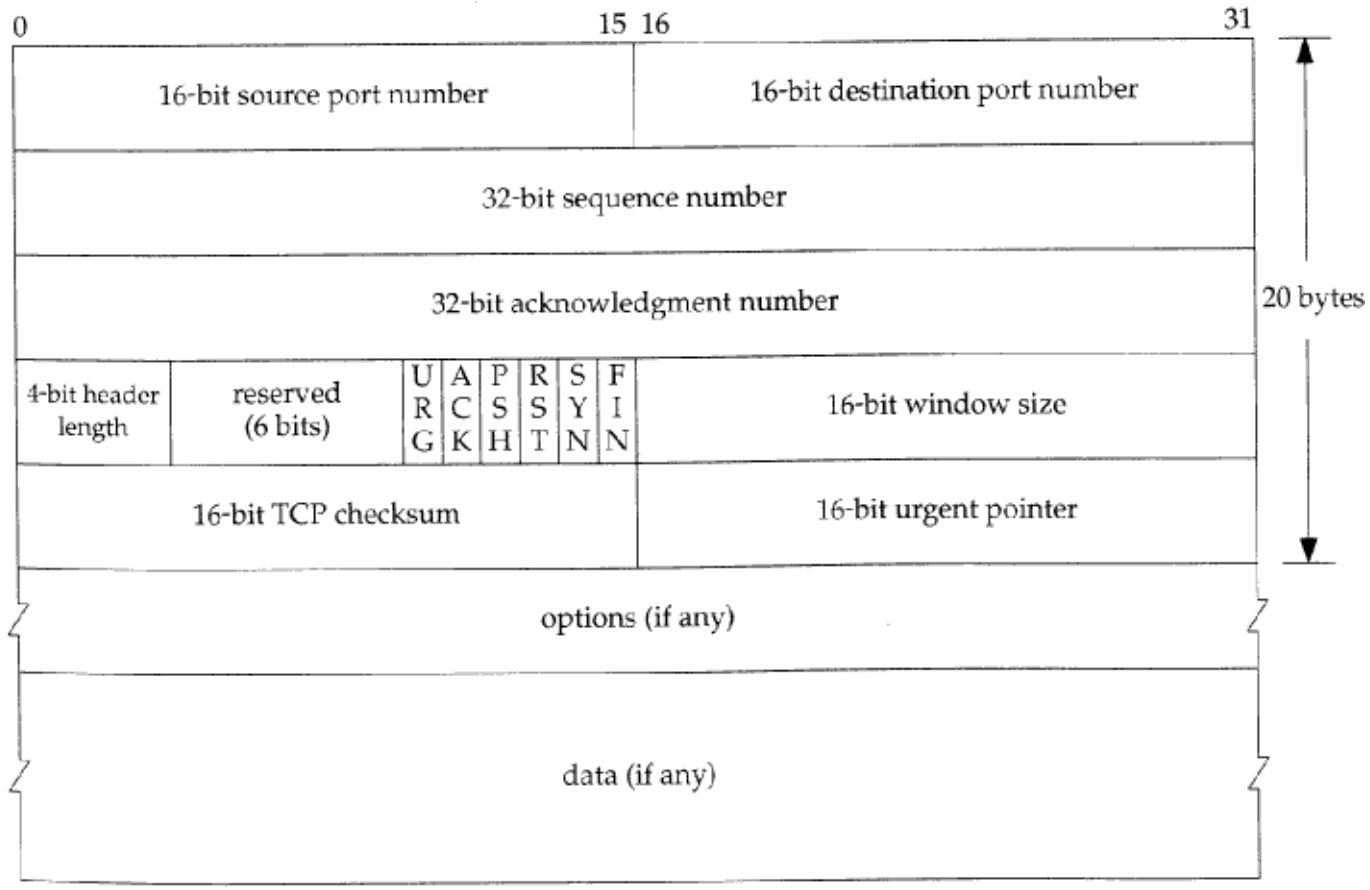
The reliability and flow control mechanisms described above require that TCP initializes and maintains certain status information for each data stream. The combination of this status, including sockets, sequence numbers and window sizes, is called a logical connection. Each connection is uniquely identified by the pair of sockets used by the sending and receiving processes.

Full Duplex

TCP provides for concurrent data streams in both directions.

TCP HEADER

TCP data is encapsulated in an IP datagram. The figure shows the format of the TCP header. Its normal size is 20 bytes unless options are present. Each of the fields is discussed below:



The **SrcPort** and **DstPort** fields identify the source and destination ports,respectively. These two fields plus the source and destination IP addresses, combine to uniquely identify each TCP connection.

The **sequence number** identifies the byte in the stream of data from the sending TCP to the receiving TCP that the first byte of data in this segment represents.

The **Acknowledgement number** field contains the next sequence number that the sender of the acknowledgement expects to receive. This is therefore the sequence number plus 1 of the last successfu**l** y received byte of data. This field is valid only if the ACK flag is on. Once a connection is established the Ack flag is always on.

The **Acknowledgement**, **SequenceNum**, and **AdvertisedWindow** fields are a**l** involved in TCP's sliding window algorithm.The Acknowledgement and AdvertisedW indow fields carry information about the flow of dat going in the other direction. In TCP's sliding window algorithm the reciever advertises a window size to the sender. This is done using the AdvertisedWindow field. The sender is then limited to having no more than a value of AdvertisedWindow bytes of un acknowledged data at any given time. The receiver sets a suitable value for the AdvertisedWindow based on the amount of memory a**l** ocated to the connection for the purpose of buffering data.

The **header length** gives the length of the header in 32-bit words. This is required because the length of the options field is variable.

The 6-bit **Flags field** is used to relay control information between TCP peers. The possible flags include SYN, FIN, RESET, PUSH, URG, and ACK.

- The SYN and Fin flags are used when establishing and terminating a TCP connection, respectively.
- The ACK flag is set any time the Acknowledgement field is valid, implying that the receiver should pay attention to it.
- The URG flag signifies that this segment contains urgent data. When this flag is set, the UrgPtr field indicates where the non-urgent data contained in this segment begins.
- The PUSH flag signifies that the sender invoked the push operation, whic h indicates to the receiving side of TCP that it should notify the receiving process of this fact.
- Fina**l** y, the RESET flag signifies that the receiver has become confused and so wants to abort the connection.

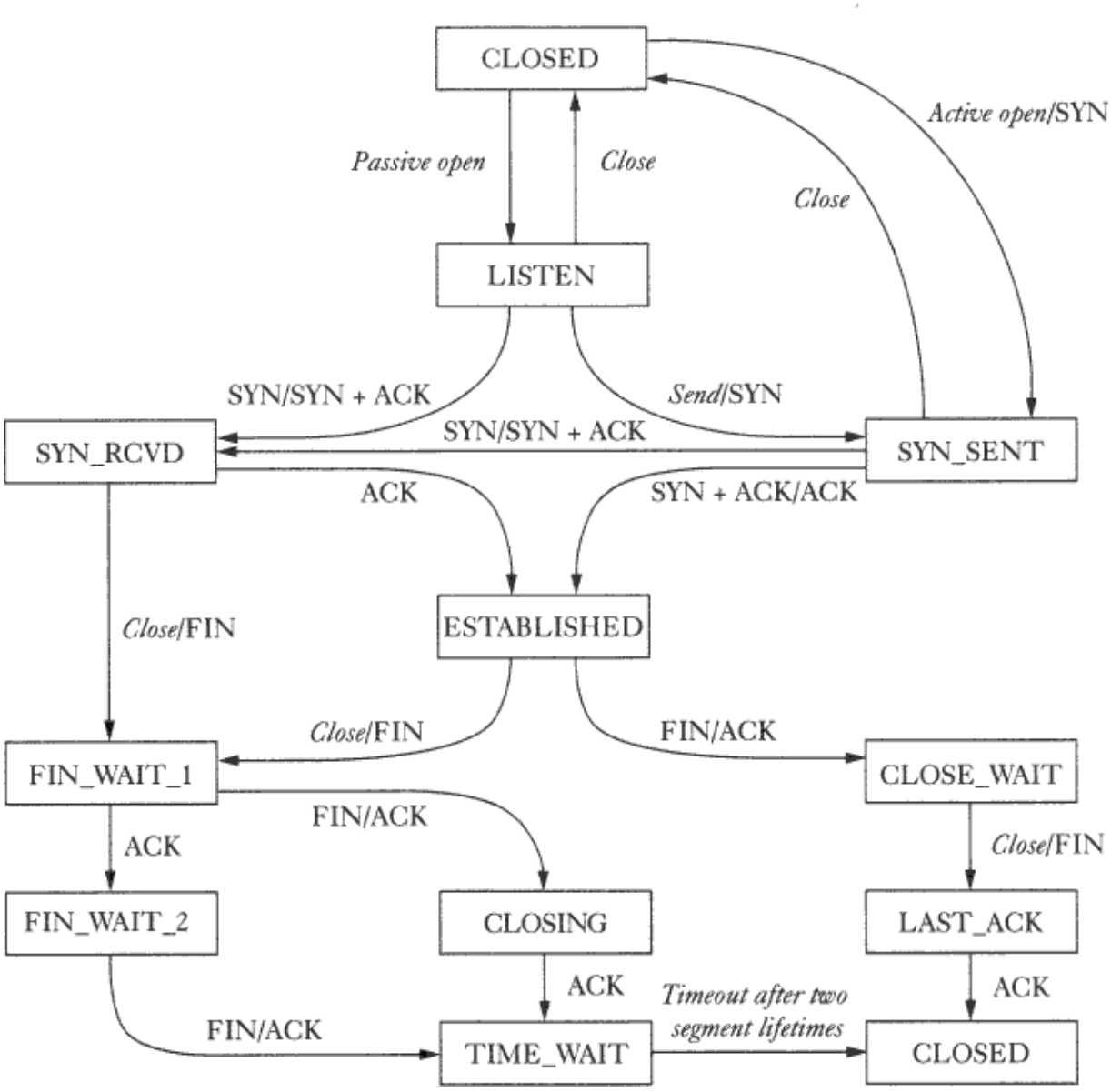
The **Checksum** covers the TCP segment: the TCP header and the TCP data. This is a mandatory field that must be calculated by the sender, and then verified by the receiver.

The **Option field** is the maximum segment size option, ca**l** ed the MSS. Each end of the connection norma**l** y specifies this option on the first segment exchanged. It specifies the maximum sized segment the sender wants to recieve.

The **data** portion of the TCP segment is optional.

UNIT 3

TCP STATE TRANSITION DIAGRAM



The two transitions leading to the ESTABLISHED state correspond to the opening of a connection, and the two transitions leading from the ESTABLISHED state are for the termination of a connection. The ESTABLISHED state is where data transfer can occur between the two ends in both the directions.

If a connection is in the LISTEN state and a SYN segment arrives, the connection makes a transition to the SYN_RCVD state and takes the action of replying with an ACK+SYN segment. The client does an active open which causes its end of the connection to send a SYN segment to the server and to move to the SYN_SENT state. The arrival of the SYN+ACK segment causes the client to mo ve to the ESTABLISHED state and to send an ack back to the server. When this ACK arrives the server fina**l** y moves to the ESTABLISHED state. In other words, we have just traced the THREE-WAY HANDSHAKE.

In the process of terminating a connection, the important thing to kee p in mind is that the application process on both sides of the connection must independently close its half of the connection. Thus, on any one side there are three combinations of transition that get a connection from the ESTABLISHED state to the CLOSED state:

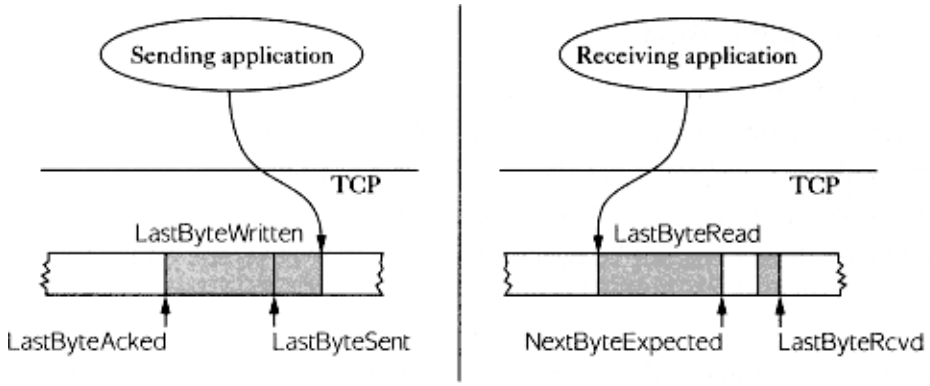
- This side closes first:
  ESTABLISHED -> FIN_WAIT_1-> FIN_WAIT_2 -> TIME_WAIT -> CLOSED.
- The other side closes first:

  ESTABLISHED -> CLOSE_WAIT -> LAST_ACK -> CLOSED.
- Both sides close at the same time:

  ESTABLISHED -> FIN_WAIT_1-> CLOSING ->TIME_WAIT -> CLOSED.

The main thing to recognize about connection teardown is that a connection in the TIME_WAIT state cannot move to the CLOSED state until it has waited for two times the maximum amount of time an IP datagram might live in the Inter net. The reason for this is that while the local side of the connection has sent an ACK in response to the other side's FIN segment, it does not know that the ACK was successful y delivered. As a consequence this other side might re transmit its FIN segment, and this second FIN segment might be delayed in the network. If the connection were allowed to move directly to the CLOSED state, then another pair of application processes might come along and open the same connection, and the delayed FIN segment from the earlier incarnation of the connection would immediately initiate the termination of the later incarnation of that connection.

SLIDING WINDOW



The sliding window serves several purposes:

(1) it guarantees the reliable delivery of data

(2) it ensures that the data is delivered in order,

(3) it enforces flow control between the sender and the receiver.

Reliable and ordered delivery

The sending and receiving sides of TCP interact in the following manner to implement reliable and ordered

delivery: Each byte has a sequence number.

ACKs are cumulative.

- ### Sending side LastByteAcked <=LastByteSent LastByteSent <= LastByteWritten
- bytes between LastByteAcked and LastByteWritten must be buffered.

Receiving side

- LastByteRead < NextByteExpected
- NextByteExpected <= LastByteRcvd + 1
- bytes between NextByteRead and LastByteRcvd must be buffered.

Flow Control

Sender buffer size :

MaxSendBuffer Receive buffer

size : MaxRcvBuffer **Receiving**

side

- LastByteRcvd - NextBytteRead <= MaxRcvBuffer
- AdvertisedWindow = MaxRcvBuffer - (LastByteRcvd - NextByteRead)

Sending side

- LastByteSent - LastByteAcked <= AdvertisedWindow
- EffectiveWindow = AdvertisedWindow - (LastByteSent - LastByteAcked)
- LastByteWritten - LastByteAcked <= MaxSendBuffer
- Block sender if (LastByteWritten - LastByteAcked) + y > MaxSendBuffer

Always send ACK in response to an arriving data

segment Persist when AdvertisedWindow = 0

Adaptive Retransmission

TCP guarantees reliable delivery and so it retransmits each segment if an ACK is not received in a certainperiod of time. TCP sets this timeout as a function of the RTT it expects between the two ends of the connection. Unfortunately, given the range of possible RTT's between any pair of hosts in the Internet, as wel as the variation in RTT between the same two hosts over time, choosing an appropriate timeout value is not that easy. To address this problem, TCP uses an adaptive retransmission medchanism. We describe this mechanism and how it has evolved over time.

Original Algorithm

Measure SampleRTT for each segment/ACK pair

Compute weighted average of RTT

EstimatedRTT = a*EstimatedRTT + b*SampleRTT, where a+b = 1

a between 0.8 and 0.9

b between 0.1 and 0.2

Set timeout based on EstimatedRTT

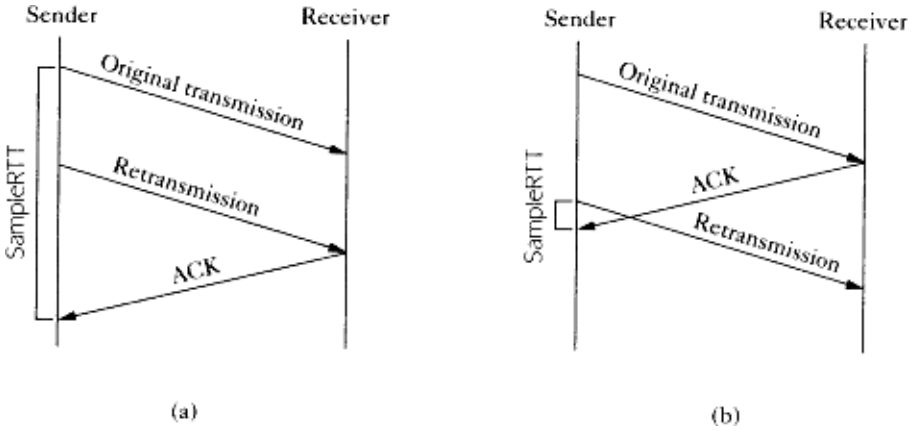
TimeOut = 2 * EstimatedRTT

Karn/Partridge Algorithm

Do not sample RTT when retransmitting

Double timeout after each retransmission

Jacobson/Karels Algorithm



New calculation for average RTT

Difference = SampleRTT -

EstimatedRTT

EstimatedRTT = EstimatedRTT + (d * Difference)

Deviation = Deviation + d (|Difference| - Deviation)), where d is a fraction between 0 and

1 Consider variance when setting timeout value

Timeout = u * EstimatedRTT + q * Deviation, where u = 1 and q = 4

Congestion Control

Slow Start

It operates by observing that the rate at which new packets should be injected into the network is the rate at which the

acknowledgments are returned by the other end.

Slow start adds another window to the sender's TCP: the congestion window, called "cwnd". When a new connection is established with a host on another network, the congestion window is initialized to one segment (i.e., the segment size announced by the other end, or the default, typically 536 or 512). Each time an ACK is received, the congestion window is increased by one segment. The sender can transmit up to the minimum of the congestion window and the advertised window. The congestion window is flow control imposed by the sender, while the advertised window is flow control imposed by the receiver. The former is based on the sender's assessment of perceived network congestion; the latter is related to the amount of available buffer space at the receiver for this connection.

The sender starts by transmitting one segment and waiting for its ACK. When that ACK is received, the congestion window is incremented from one to two, and two segments can be sent. When each of those two segments is acknowledged, the congestion window is increased to four. This provides an exponential growth, although it is not exactly exponential because the receiver may delay its ACKs, typically sending one ACK for every two segments that it receives.

At some point the capacity of the internet can be reached, and an intermediate router will start discarding packets. This tells the sender that its congestion window has gotten too large.

Early implementations performed slow start only if the other end was on a different network. Current implementations always perform slow start.

Congestion Avoidance

Congestion can occur when data arrives on a big pipe (a fast LAN) and gets sent out a small er pipe (a slower WAN). Congestion can also occur when multiple input streams arrive at a router whose output capacity is less than the sum of the inputs. Congestion avoidance is a way to deal with lost packets.

The assumption of the algorithm is that packet loss caused by damage is very small (much less than 1%), therefore the loss of a packet signals congestion somewhere in the network between the source and destination. There are two indications of packet loss: a timeout occurring and the receipt of duplicate ACKs.

Congestion avoidance and slow start are independent algorithms with different objectives. But when congestion occurs TCP must slow down its transmission rate of packets into the network, and then invoke slow start to get things going again. In practice they are implemented together.

Congestion avoidance and slow start require that two variables be maintained for each connection: a congestion window, cwnd, and a slow start threshold size, ssthresh. The combined algorithm operates as follows:

1. Initialization for a given connection sets cwnd to one segment and ssthresh to 65535 bytes.

2. The TCP output routine never sends more than the minimum of cwnd and the receiver's advertised window.

3. When congestion occurs (indicated by a timeout or the reception of duplicate ACKs), one-half of the current window size (the minimum of cwnd and the receiver's advertised window, but at least two segments) is saved in ssthresh. Additionally, if the congestion is indicated by a timeout, cwnd is set to one segment (i.e., slow start).

4. When new data is acknowledged by the other end, increase cwnd, but the way it increases depends on whether TCP is performing slow start or congestion avoidance.

If cwnd is less than or equal to ssthresh, TCP is in slow start; otherwise TCP is performing congestion avoidance. Slow start continues until TCP is halfway to where it was when congestion occurred (since it recorded half of the window size

that caused the problem in step 2), and then congestion avoidance takes over.

Slow start has cwnd begin at one segment, and be incremented by one segment every time an ACK is received. As mentioned earlier, this opens the window exponentially: send one segment, then two, then four, and so on. Congestion avoidance dictates that cwnd be incremented by segsize*segsize/cwnd each time an ACK is received, where segsize is the segment size and cwnd is maintained in bytes. This is a linear growth of cwnd, compared to slow start's exponential growth. The increase in cwnd should be at most one segment each round-trip time (regardless how many ACKs are received in that RTT), whereas slow start increments cwnd by the number of ACKs received in a round-trip time.

Fast Retransmit

TCP may generate an immediate acknowledgment (a duplicate ACK) when an out- of-order segment is received. This duplicate ACK should not be delayed. The purpose of this duplicate ACK is to let the other end know that a segment was received out of order, and to tell it what sequence number is expected.

Since TCP does not know whether a duplicate ACK is caused by a lost segment or just a reordering of segments, it waits for a small number of duplicate ACKs to be received. It is assumed that if there is just a reordering of the segments, there will be only one or two duplicate ACKs before the reordered segment is processed, which will then generate a new ACK. If three or more duplicate ACKs are received in a row, it is a strong indication that a segment has been lost. TCP then performs a retransmission of what appears to be the missing segment, without waiting for a retransmission timer to expire.

Fast Recovery

After fast retransmit sends what appears to be the missing segment, congestion avoidance, but not slow start is performed. This is the fast recovery algorithm. It is an improvement that allows high throughput under moderate congestion, especially for large windows.

The reason for not performing slow start in this case is that the receipt of the duplicate ACKs tells TCP more than just a packet has been lost. Since the receiver can only generate the duplicate ACK when another segment is received, that segment has left the network and is in the receiver's buffer. That is, there is still data flowing between the two ends, and TCP does not want to reduce the flow abruptly by going into slow start.

The fast retransmit and fast recovery algorithms are usually implemented together as follows.

1. When the third duplicate ACK in a row is received, set ssthresh to one-half the current congestion window, cwnd, but no less than two segments. Retransmit the missing segment. Set cwnd to ssthresh plus 3 times the segment size. This inflates the congestion window by the number of segments that have left the network and which the other end has cached .

2. Each time another duplicate ACK arrives, increment cwnd by the segment size. This inflates the congestion window for the additional segment that has left the network. Transmit a packet, if allowed by the new value of cwnd.

3. When the next ACK arrives that acknowledges new data, set cwnd to ssthresh (the value set in step 1). This ACK should be the acknowledgment of the retransmission from step 1, one round-trip time after the retransmission. Additionally, this ACK should acknowledge all the intermediate segments sent between the lost packet and the receipt of the first duplicate ACK. This step is congestion avoidance, since TCP is down to one-half the rate it was at when the packet was lost.

Prev. Topic: TCP Basic Operation: Connection Establishment, Management and Termination

Pages 1 2 3

Next Topic: TCP Connection Preparation: Transmission Control Blocks (TCB) and Passive and Active Socket *OPEN*s

TCP Operational Overview and the TCP Finite State Machine (FSM)

(Page 2 of 3)

The Simplified TCP Finite State Machine

In the case of TCP, the finite state machine can be considered to describe the "life stages" of a connection. Each connection between one TCP device and another begins in a null state where there is no connection, and then proceeds through a series of states until a connection is established. It remains in that state until something occurs to cause the connection to be closed again, at which point it proceeds through another sequence of transitional states and returns to the closed state.

The full description of the states, events and transitions in a TCP connection is lengthy and complicated—not surprising, since that would cover much of the entire TCP standard. For our purposes, that level of detail would be a good cure for insomnia but not much else. However, a ***simplified*** look at the TCP FSM will help give us a nice overall feel for how TCP establishes connections and then functions when a connection has been created.

Table 151 briefly describes each of the TCP states in a TCP connection, and also describes the main events that occur in each state, and what actions and transitions occur as a result. For brevity, three abbreviations are used for three types of message that control transitions between states, which correspond to the TCP header flags that are set to indicate a message is serving that function. These are:

- ***SYN:*** A *synchronize* message, used to initiate and establish a connection. It is so named since one of its functions is to synchronizes sequence numbers between devices.
- ***FIN:*** A *finish* message, which is a TCP segment with the *FIN* bit set, indicating that a

device wants to terminate the connection.

- ***ACK:*** An *acknowledgment*, indicating receipt of a message such as a *SYN* or a *FIN*.

Again, I have not shown every possible transition, just the ones normally followed in the life of a connection. Error conditions also cause transitions but including these would move us well beyond a "simplified" state machine. The FSM is also illustrated in Figure 210, which you may find easier for seeing how state transitions occur.

Table 151: TCP Finite State Machine (FSM) States, Events and Transitions

State	State Description	Event and Transition
CLOSED	This is the default state that each connection starts in before the process of establishing it begins. The state is called "fictional" in the standard. The reason is that this state represents the situation where there is no connection	**Passive Open:** A server begins the process of connection setup by doing a passive open on a TCP port. At the same time, it sets up the data structure (transmission control block or TCB) needed to manage the connection. It then transitions to the *LISTEN* state.

	between devices—it either hasn't been created yet, or has just been destroyed. If that makes sense. ☺	**Active Open, Send *SYN*:** A client begins connection setup by sending a *SYN* message, and also sets up a TCB for this connection. It then transitions to the *SYN-SENT* state.
LISTEN	A device (normally a server) is waiting to receive a *synchronize* (*SYN*) message from a client. It has not yet sent its own *SYN* message.	**Receive Client *SYN*, Send *SYN+ACK*:** The server device receives a *SYN* from a client. It sends back a message that contains its own *SYN* and also acknowledges the one it received. The server moves to the *SYN-RECEIVED* state.
SYN-SENT	The device (normally a client) has sent a *synchronize* (*SYN*) message and is waiting for a matching *SYN* from the other device (usually a server).	**Receive *SYN*, Send *ACK*:** If the device that has sent its *SYN* message receives a *SYN* from the other device but not an *ACK* for its own *SYN*, it acknowledges the *SYN* it receives and then transitions to *SYN-RECEIVED* to wait for the acknowledgment to its *SYN*. **Receive *SYN+ACK*, Send *ACK*:** If the device that sent the *SYN* receives both an acknowledgment to its *SYN* and also a *SYN* from the other device, it acknowledges the *SYN* received and then moves straight to the *ESTABLISHED* state.
SYN-RECEIVED	The device has both received a *SYN* (connection request) from its partner and sent its own *SYN*. It is now waiting for an *ACK* to its *SYN* to finish connection setup.	**Receive *ACK*:** When the device receives the *ACK* to the *SYN* it sent, it transitions to the *ESTABLISHED* state.
ESTABLISHED	The "steady state" of an open TCP connection. Data can be exchanged freely once both devices in the connection enter this state. This will continue until the connection is closed for one reason or another.	**Close, Send *FIN*:** A device can close the connection by sending a message with the *FIN* (*finish*) bit sent and transition to the *FIN-WAIT-1* state. **Receive *FIN*:** A device may receive a *FIN* message from its connection partner asking that the connection be closed. It will acknowledge this message and transition to the *CLOSE-WAIT* state.
CLOSE-WAIT	The device has received a close request (*FIN*) from the other device. It must now wait for the application on the local device to acknowledge this request and generate a matching request.	**Close, Send *FIN*:** The application using TCP, having been informed the other process wants to shut down, sends a close request to the TCP layer on the machine upon which it is running. TCP then sends a *FIN* to the remote device that already asked to terminate the connection. This device now transitions to *LAST-ACK*.
LAST-ACK	A device that has already received a close request and acknowledged it, has sent its own *FIN* and is waiting for an *ACK* to this request.	**Receive *ACK* for *FIN*:** The device receives an acknowledgment for its close request. We have now sent our *FIN* and had it acknowledged, and received the other device's *FIN* and acknowledged it, so we go straight to the *CLOSED* state.
FIN-WAIT-1	A device in this state is waiting for an *ACK* for a *FIN* it has sent, or is waiting for a connection termination request from the other device.	**Receive *ACK* for *FIN*:** The device receives an acknowledgment for its close request. It transitions to the *FIN-WAIT-2* state. **Receive *FIN*, Send *ACK*:** The device does not receive an *ACK* for its own *FIN*, but receives a *FIN* from the other device. It acknowledges it, and moves to the *CLOSING* state.
FIN-WAIT-2	A device in this state has received an ACK for its request to terminate the connection and is now waiting for a matching *FIN* from the other device.	**Receive *FIN*, Send *ACK*:** The device receives a *FIN* from the other device. It acknowledges it and moves to the *TIME-WAIT* state.
CLOSING	The device has received a *FIN* from the other device and sent an *ACK* for it, but not yet received an *ACK* for its own *FIN* message.	**Receive *ACK* for *FIN*:** The device receives an acknowledgment for its close request. It transitions to the *TIME-WAIT* state.

TIME-WAIT	The device has now received a *FIN* from the other device and acknowledged it, and sent its own *FIN* and received an *ACK* for it. We are done, except for waiting to ensure the ACK is received and prevent potential overlap with new connections. (See the topic describing connection termination for more details on this state.)	**Timer Expiration:** After a designated wait period, device transitions to the *CLOSED* state.

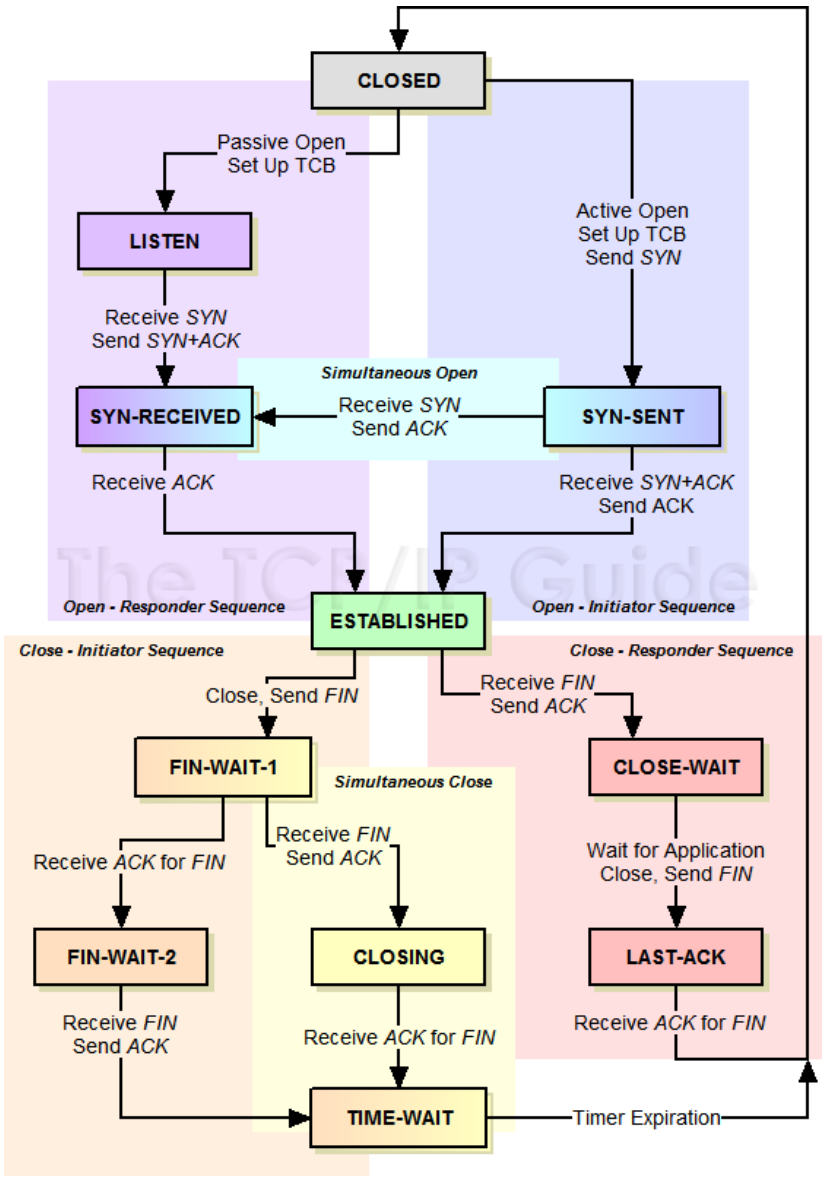


Figure 210: The TCP Finite State Machine (FSM)

This diagram illustrates the simplified TCP FSM. The color codings are not an official part of the definition of the FSM; I have added them to show more clearly the sequences taken by the two devices to open and close a link. For both establishment and termination there is a regular sequence, where the initiating and responding devices go through different states, and a *simultaneous* sequence where each uses the same sequence.

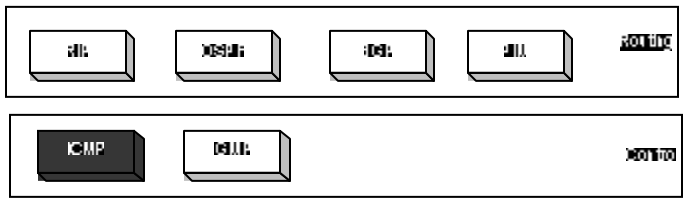
Tap tap… still awake? Okay, I guess even with serious simplification, that FSM isn't all that simple. It may seem a bit intimidating at first, but if you take a few minutes with it, you can get a good handle on how TCP works. The FSM will be of great use in making sense of the connection establishment and termination processes later in this section—and conversely, reading those sections will help you make sense of the FSM. So if your eyes have glazed over completely, just

carry on and try coming back to this topic later.

Unit 4

Relates to Lab 2:

1

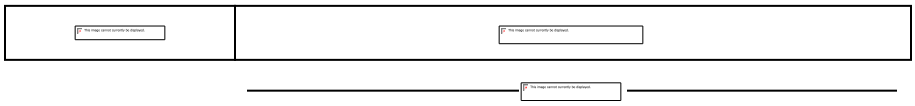


- The IP (Internet Protocol) relies on several other protocols to perform necessary control and routing functions:
  - Control functions (ICMP)
  - Multicast signaling (IGMP)
  - Setting up routing tables (RIP, OSPF, BGP, PIM, …)

Overview

- The **Internet Control Message Protocol (ICMP)** is a helper protocol that supports IP with facility for
  - Error reporting
  - Simple queries

- ICMP messages are encapsulated as IP datagrams:



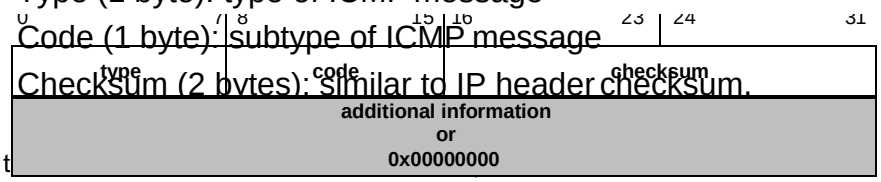
3

ICMP message format

bit #

4 byte header:

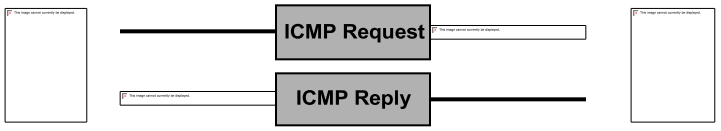
- Type (1 byte): type of ICMP message
- Code (1 byte): subtype of ICMP message
- Checksum (2 bytes): similar to IP header checksum.



If t

4

ICMP Query message



Host

Host or router

ICMP query:

- Request sent by host to a router or host
- Reply sent back to querying host

5

Example of ICMP Queries

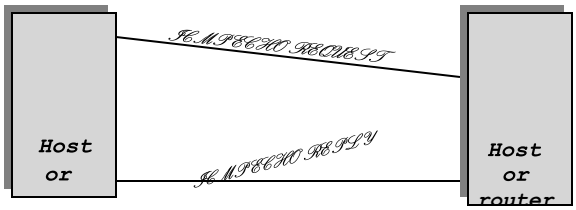
Type/Code:	Description	
8/0	Echo Request	The ping command uses Echo Request/ Echo Reply
0/0	Echo Reply	
13/0	Timestamp Request	
14/0	Timestamp Reply	
10/0	Router Solicitation	

9/0	Router Advertisement

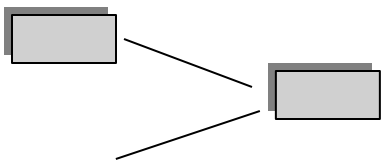
6

Example of a Query: Echo Request and Reply

- Ping's are handled directly by the kernel
- Each Ping is translated into an ICMP Echo Request
- The Ping'ed host responds with an ICMP Echo Reply



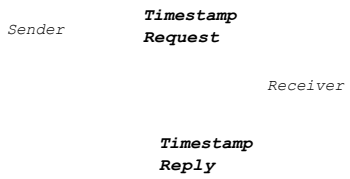
7



Type (= 17 or 18)	Code (=0)	Checksum
identifier		sequence number
32-bit sender timestamp		
32-bit receive timestamp		
32-bit transmit timestamp		

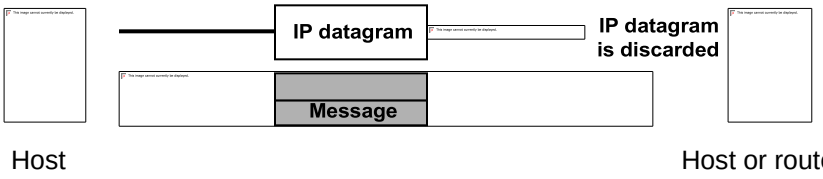
Example of a Query: ICMP Timestamp

- A system (host or router) asks another system for the current time.
- Time is measured in milliseconds after midnight UTC (Universal Coordinated Time) of the current day
- Sender sends a request, receiver responds with reply



8

ICMP Error message

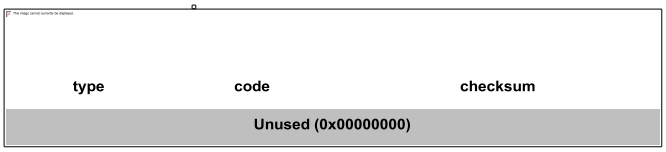


- **ICMP error messages report error conditions**
- **Typically sent when a datagram is discarded**
- **Error message is often passed from ICMP to the application program**

9

ICMP Error message

ICMP Message



- ICMP error messages include the complete IP header and the first 8 bytes of the payload (typically: UDP, TCP)

IP header	ICMP header	IP header	8 bytes of payload

10

Frequent ICMP Error message

Type	Code	Description	
3	0–15	Destination unreachable	Notification that an IP datagram could not be forwarded and was dropped. The code field contains an explanation.
5	0–3	Redirect	Informs about an alternative route for the datagram and should result in a routing table update. The code field explains the reason for the route change.
11	0, 1	Time exceeded	Sent when the TTL field has reached zero (Code 0) or when there is a timeout for the reassembly of segments (Code 1)
12	0, 1	Parameter problem	Sent when the IP header is invalid (Code 0) or when an IP header option is missing (Code 1)

11

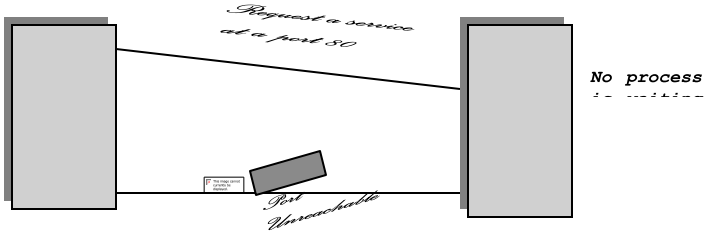
Some subtypes of the “Destination Unreachable”

Code	Description	Reason for Sending
0	Network Unreachable	No routing table entry is available for the destination network.
1	Host Unreachable	Destination host should be directly reachable, but does not respond to ARP Requests.
2	Protocol Unreachable	The protocol in the protocol field of the IP header is not supported at the destination.
3	Port Unreachable	The transport protocol at the destination host cannot pass the datagram to an application.
4	Fragmentation Needed and DF Bit Set	IP datagram must be fragmented, but the DF bit in the IP header is set.

12

Example: ICMP Port Unreachable

- RFC 792: If, in the destination host, the IP module cannot deliver the datagram because the indicated protocol module or process port is not active, the destination host may send a destination unreachable message to the source host.
- Scenario:



13

Supernetting

Supernetting is used in routing tables to compact contiguous Class C networks. Suppose that a company needs to address 1,024 hosts. The company is assigned the four contiguous Class C addresses of 192.168.0.0 through 192.168.3.0, and it sets up its router to the Internet with the address of 192.168.0.1. The routes in the ISP routing table will contain the following:

Network	Subnet Mask	Route
192.168.0.0	255.255.255.0	192.168.0.1
192.168.1.0	255.255.255.0	192.168.0.1
192.168.2.0	255.255.255.0	192.168.0.1
192.168.3.0	255.255.255.0	192.168.0.1

Notice that all of the routes point to the same IP address of 192.168.0.1. These routes therefore seem redundant. The subnet mask tells IP at the router to examine 24 bits of every packet to determine the route that each packet will take. IP then examines 24 bits of the destination address of each packet and finds that the only difference in any of these four routes is in the third octet (specifically the 23rd and 24th bit):

Network	Third Octet
192.168.0.0	0000 0000
192.168.1.0	0000 0001
192.168.2.0	0000 0010
192.168.3.0	0000 0011

Any packet that is bound for any of these contiguous networks has the same first 22 bits;

the only difference is in the 23rd and 24th bits. Since all of the networks are routed to the same IP address, supernetting can tell IP to look at only 22 bits. Using supernetting, the same routing table would include only one route instead of four:

Network	**Subnet Mask**	**Route**
192.168.0.0	255.255.252.0	192.168.0.1

Now if a packet is bound for 192.168.1.12, 192.168.2.115, 192.168.3.5, or 192.168.0.10, the subnet mask of 255.255.252.0 tells IP to look only at the first 22 bits. All of these addresses have the same first 22 bits:

Destination	First 22 Bits	Last 10 Bits
192.168.0.10	1100 0000.1010 1000.0000 00	00.0000 1010
192.168.1.12	1100 0000.1010 1000.0000 00	01.0000 1100
192.168.2.115	1100 0000.1010 1000.0000 00	10.0111 0011
192.168.3.5	1100 0000.1010 1000.0000 00	11.0000 0101

Internet Protocol Version 6 (IPv6)

Overview of IPv6

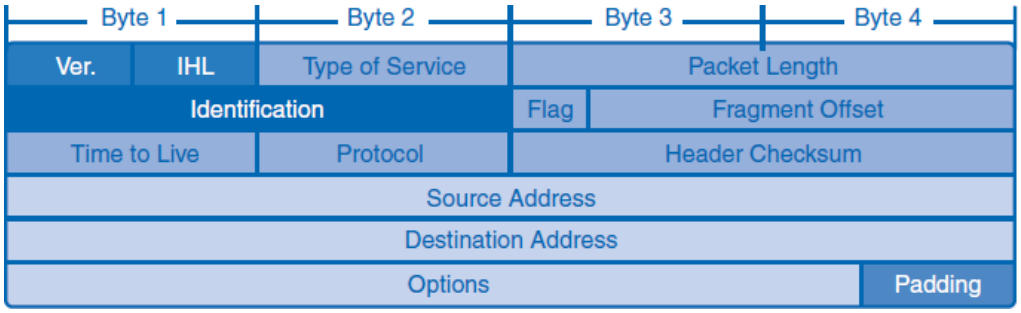
In the early 1990s, the Internet Engineering Task Force (IETF) grew concerned about the exhaustion of the IPv4 network addresses and began to look for a replacement for this protocol. This activity led to the development of what is now known as IPv6. This section presents a brief introduction to IPv6.

Creating expanded addressing capabilities was the initial motivation for developing this new protocol. Other issues were also considered during the development of IPv6, such as these:

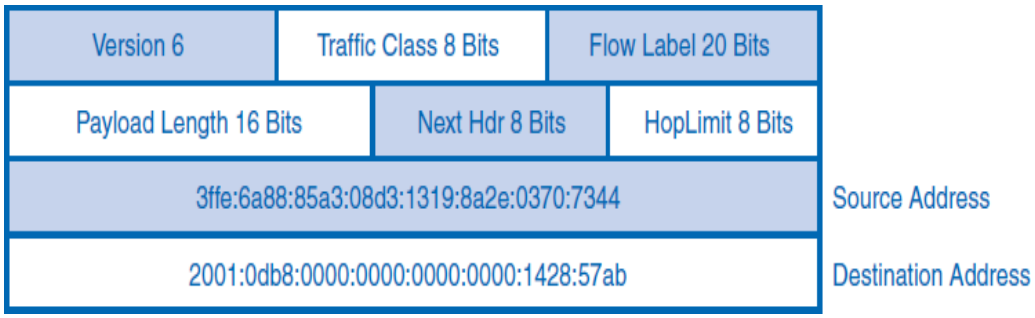
- Improved packet handling
- Increased scalability and longevity
- Quality of service (QoS) mechanisms
- Integrated security

To provide these features, IPv6 offers the following:

- 128-bit hierarchical addressing to expand addressing capabilities
- Header format simplification to improve packet handling
- Improved support for extensions and options for increased scalability/longevity and improved packet handling
- Flow-labeling capabilities as QoS mechanisms
- Authentication and privacy capabilities to integrate security



IPv4 Packet Header Fields

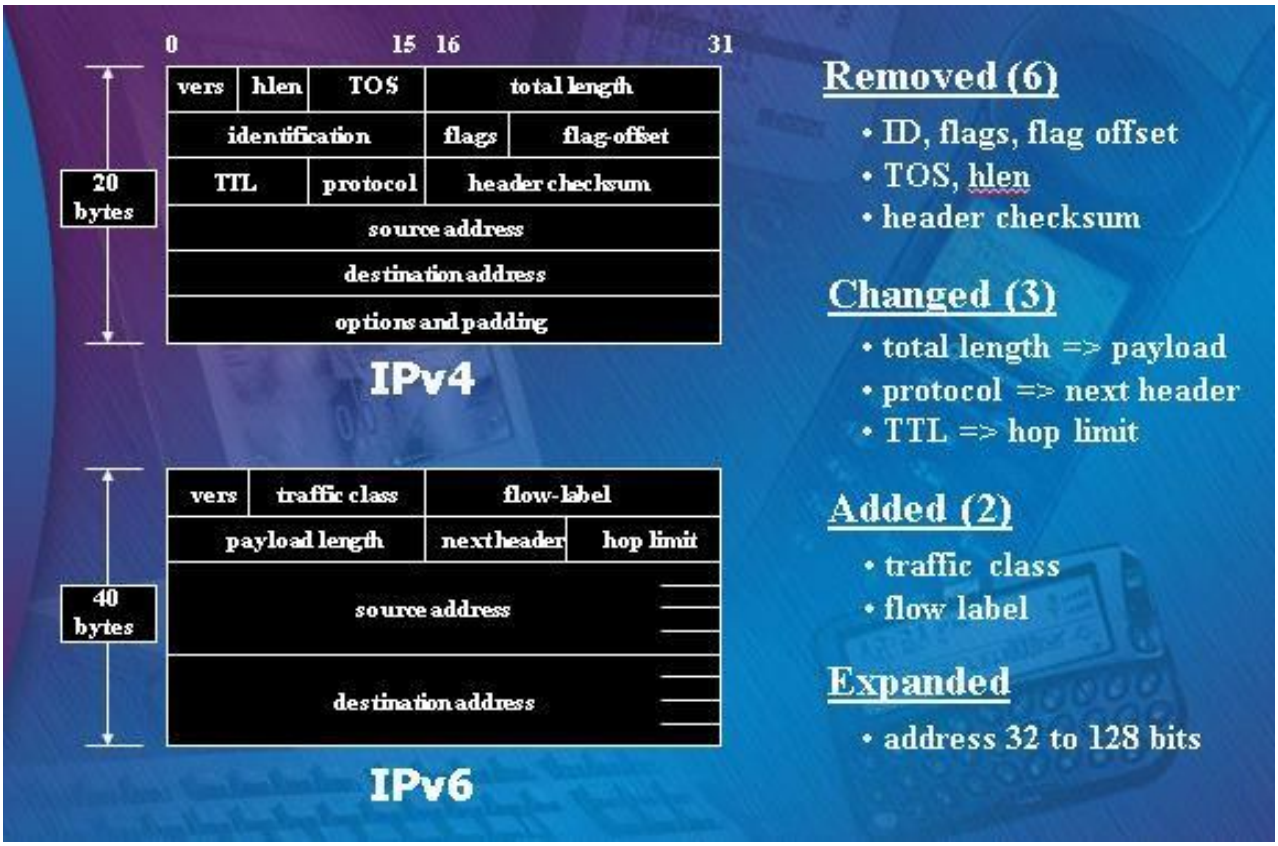


IPv6 Packet Header Fields

Features of IPv6

- Larger Address Space (128-bit IPv6 Address)
- Aggregation-based address hierarchy – Efficient backbone routing
- Efficient and Extensible IP datagram
- Stateless Address Autoconfiguration
- Security (IPsec mandatory)
- Mobility

Header Comparison between IPv4 and IPv6



UNIT 5

Internet Protocol version 6

Internet Protocol version 6 is a new addressing protocol designed to incorporate all the possible requirements of future Internet known to us as Internet version 2. This protocol as its predecessor IPv4, works on the Network Layer (Layer-3). Along with its offering of an enormous amount of logical address space, this protocol has ample features to address the shortcoming of IPv4.

WhyNewIPVersion?

So far, IPv4 has proven itself as a robust routable addressing protocol and has served us for decades on its best-effort-delivery mechanism. It was designed in the early 80s and did not get any major change afterward. At the time of its birth, Internet was limited only to a few universities for their research and to the Department of Defense. IPv4 is 32 bits long and offers around 4,294,967,296 (2^{32}) addresses. This address space was considered more than enough that

time.

Given below are the major points that played a key role in the birth of IPv6:

- Internet has grown exponentially and the address space allowed by IPv4 is saturating. There is a requirement to have a protocol that can satisfy the needs of future Internet addresses that is expected to grow in an unexpected manner.
- IPv4 on its own does not provide any security features. Data has to be encrypted with some other security application before being sent on the Internet.
- Data prioritization in IPv4 is not up-to-date. Though IPv4 has a few bits reserved for Type of Service or Quality of Service, but they do not provide much functionality.
- IPv4 enabled clients can be configured manually or they need some address configuration mechanism. It does not have a mechanism to configure a device to have globally unique IP address.

WhyNotIPv5?

Till date, Internet Protocol has been recognized has IPv4 only. Version 0 to 3 were used while the protocol was itself under development and experimental process. So, we can assume lots of background activities remain active before putting a protocol into production. Similarly, protocol version 5 was used while experimenting with the stream protocol for Internet. It is known to us as Internet Stream Protocol which used Internet Protocol number 5 to encapsulate its datagram. It was never brought into public use, but it was already used.

Here is a table of IP versions and how they are used:

Decimal	Keyword	Version
0-1		Reserved
2-3		Unassigned
4	IP	Internet Protocol
5	ST	ST Datagram mode
6	IPv6	Internet Protocol version 6
7	TP/IX	TP/IX: The Next Internet
8	PIP	The P Internet Protocol
9	TUBA	TUBA
10-14		Unassigned
15		Reserved

BriefHistory

After IPv4's development in the early 80s, the available IPv4 address pool begun to shrink rapidly as the demand of addresses exponentially increased with Internet. Taking pre- cognizance of the situation that might arise, IETF, in 1994, initiated the development of an addressing protocol to replace IPv4. The progress of IPv6 can be tracked by means of the RFC published:

- 1998 – RFC 2460 – Basic Protocol
- 2003 – RFC 2553 – Basic Socket API
- 2003 – RFC 3315 – DHCPv6
- 2004 – RFC 3775 – Mobile IPv6
- 2004 – RFC 3697 – Flow Label Specification
- 2006 – RFC 4291 – Address architecture (revision)
- 2006 – RFC 4294 – Node requirement

On June 06, 2012, some of the Internet giants chose to put their Servers on IPv6. Presently they are using Dual Stack mechanism to implement IPv6 in parallel with IPv4.

The successor of IPv4 is not designed to be backward compatible. Trying to keep the basic functionalities of IP addressing, IPv6 is redesigned entirely. It offers the following features:

Larger Address Space

In contrast to IPv4, IPv6 uses 4 times more bits to address a device on the Internet. This much of extra bits can provide approximately 3.4×1038 different combinations of addresses. This address can accumulate the aggressive requirement of address allotment for almost everything in this world. According to an estimate, 1564 addresses can be allocated to every square meter of this earth.

Simplified Header

IPv6's header has been simplified by moving all unnecessary information and options (which are present in IPv4 header) to the end of the IPv6 header. IPv6 header is only twice as bigger than IPv4 provided the fact that IPv6 address is four times longer.

End-to-end Connectivity

Every system now has unique IP address and can traverse through the Internet without using NAT or other translating components. After IPv6 is fully implemented, every host can directly reach other hosts on the Internet, with some limitations involved like Firewall, organization policies, etc.

Auto-configuration

IPv6 supports both stateful and stateless auto-configuration mode of its host devices. This way, absence of a DHCP server does not put a halt on inter-segment communication.

Faster Forwarding/Routing

Simplified header puts all unnecessary information at the end of the header. The information contained in the first part of the header is adequate for a Router to take routing decisions, thus making routing decision as quickly as looking at the mandatory header.

IPSec

Initially it was decided that IPv6 must have IPSec security, making it more secure than IPv4. This feature has now been made optional.

No Broadcast

Though Ethernet/Token Ring are considered as broadcast network because they support Broadcasting, IPv6 does not have any broadcast support anymore. It uses multicast to communicate with multiple hosts.

Anycast Support

This is another characteristic of IPv6. IPv6 has introduced Anycast mode of packet routing. In this mode, multiple interfaces over the Internet are assigned same Anycast IP address. Routers, while routing, send the packet to the nearest destination.

Mobility

IPv6 was designed keeping mobility in mind. This feature enables hosts (such as mobile phone) to roam around in different geographical area and remain connected with the same IP address. The mobility feature of IPv6 takes advantage of auto IP configuration and Extension headers.

Enhanced Priority Support

IPv4 used 6 bits DSCP (Differential Service Code Point) and 2 bits ECN (Explicit Congestion Notification) to provide Quality of Service but it could only be used if the end-to-end devices support it, that is, the source and destination device and underlying network must support it.

In IPv6, Traffic class and Flow label are used to tell the underlying routers how to efficiently process the packet and route it.

Smooth Transition

Large IP address scheme in IPv6 enables to allocate devices with globally unique IP addresses. This mechanism saves IP addresses and NAT is not required. So devices can send/receive data among each other, for example, VoIP and/or any streaming media can be used much efficiently.

Other fact is, the header is less loaded, so routers can take forwarding decisions and forward them as quickly as they arrive.

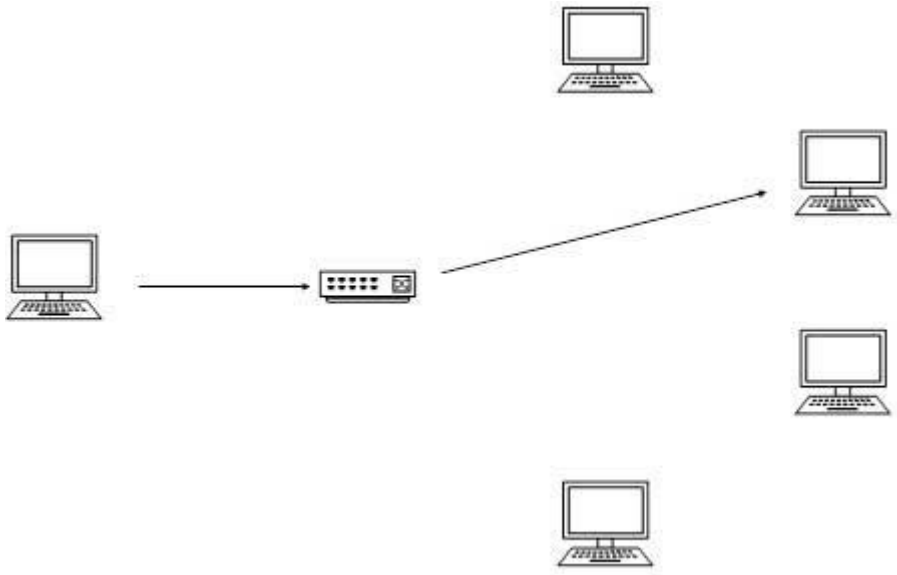
Extensibility

One of the major advantages of IPv6 header is that it is extensible to add more information in the option part. IPv4 provides only 40-bytes for options, whereas options in IPv6 can be as much as the size of IPv6 packet itself.

In computer networking, addressing mode refers to the mechanism of hosting an address on the network. IPv6 offers several types of modes by which a single host can be addressed. More than one host can be addressed at once or the host at the closest distance can be addressed.

Unicast

In unicast mode of addressing, an IPv6 interface (host) is uniquely identified in a network segment. The IPv6 packet contains both source and destination IP addresses. A host interface is equipped with an IP address which is unique in that network segment. When a network switch or a router receives a unicast IP packet, destined to a single host, it sends out one of its outgoing interface which connects to that particular host.



[*Image: Unicast Messaging*]

The IPv6 multicast mode is same as that of IPv4. The packet destined to multiple hosts is sent on a special multicast address. All the hosts interested in that multicast information need to join that multicast group first. All the interfaces that joined the group receive the multicast packet and process it, while other hosts not interested in multicast packets ignore the multicast information.

KARPAGAM ACADEMY OF HIGHER EDUCATION

COIMBATORE-21

Faculty of Engineering

Department of Computer Science and Engineering

UNIVERSITY EXAMINATION

Subject Code : 13BECSE04

Title of the paper : TCP/IP Design and Implementation **Max Marks : 60 Marks**

PART-A (9*2=18 Marks)

Answer the following question

1. What is the use of window advertisement in a TCP SEGMENT?
2. How will you define push function in TCP?
3. What is meant by TCP/IP?
4. Define Finite State Machine.
5. How to control congestion while transmission?
6. Discuss about the various data structures used in TCP implementation.
7. What is mean by Transmission Control Blocks?
8. Explain deleting and inserting timer event in TCP.
9. What is mean by TCP allocation and initialization?

PART-B (3*14=42)

ANSWER all the Questions

10. a) Explain in detail about Transmission Control Blocks. (14)

(OR)

b) Write a note on Data Structure and Mutual Exclusion. (14)

---

11. a) Explain in detail about Events and Messages for TCP. (14)

(OR)

b) Explain in detail about Timer Process. (14)

12. a) Explain in detail about Finite State Machine implementation. (14)

(OR)

b) Explain in detail about Push function used in TCP. (14)

Enable | Enlighten | Enrich
KARPAGAM
ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
(Established Under Section 3 of UGC Act, 1956)

KARPAGAM ACADEMY OF HIGHER EDUCATION

COIMBATORE-21

Faculty of Engineering

Department of Computer Science and Engineering

Subject Code : 13BECSE04

Title of the paper : TCP/IP Design and Implementation Max Marks : 20

ANSWER all the Questions

1. Explain Data Structure and Transmission Control Block of TCP. (10)

2. Discuss in detail about Congestion Avoidance and Control in TCP (10)

Reg.……

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Established Under Section 3 of UGC Act 1956)

COIMBATORE-641 021

(For the candidates admitted from 2008 onwards) – FULL TIME

B.E DEGREE EXAMINATION

COMPUTER SCIENCE AND ENGINEERING

TCP/IP DESIGN AND IMPLEMENTATION

Time: 3 Hours **Maximum: 100**

PART – A (20 x 1 = 20 Marks)

1.Number of bits per symbol used in Baudot code is
a. 7 c. 8
b. 5 d. 9

2.What is the main difference between DDCMP and SDLC?
a. DDCMP does not need special hardware to final the beginning of a message
b. DDCMP has a message header
c. SDLC has a IP address
d. SDLC does not use CRC

3.An example of digital, rather than analog, communication is
a. DDD c. WATS
b. DDS d. DDT

4.Terminals are required for
a. real-time, batch processing & time-sharing
b. real time, time-sharing & distributed message processing
c. real time, distributed processing & manager inquiry
d**. real-time, time sharing & message switching**

5.The receive equilizer reduces delay distortions using a
a. tapped delay lines c. descrambler
b. gearshift d. difference engine

6.Ina synchronous modem, the receive equilizer is known as

a. adaptive equilizer c. statistical equilizer

b. impariment equilizer d. compromise equilizer

7.The channel in the data communication model can be

a. postal mail services c. radio lines

b. telephone lines **d. any of the above**

8.A data terminal serves as an

a. Effector **c. both a and b**

b. sensor d. neither a nor b

9.Which of the following transmission systems provide the highest data rate to in individual device?

a. computer bus c. voice and mode

b. telephone lines d. lease lines

10.A protocol is a set of rules governing a time sequence of events that must take place

a. between peers c. between modems

b. between an interface d. across an interface

11.A remote batch-processing operation in which data is solely input to a central computer would require

a. telegraphp line c. mixed bad channel

b. simplex lines d. all of above

12.A band is always equivalent to

a. a byte c. 100 bits

b. a bit **d. none of above**

13.The loss in signal power as light travels down the fiber is called

a. attenuation c. scattering

b. progragation d. interruption

14.Avalanche photodiode receivers can detect bits of transmitted data by receiving

a. 100 photons c. 2000 photons

b. 200 photons d. 300 photons

15.Communiction circuits that transmit data in both directions but not at the same time are operating in

a. a simplex mode c. a full duplex mode

b. a half duplex mode d. an asynchronous mode

16.An example of a medium speed, switched communications service is

a. series 1000 **c. DDD**

b. data phone 50 d. All of the above

17.In communication satellite, multiple repeaters are known as
a. detector
c. stations
b. modulator
d. transponders

18.While transmitting odd-parity coded symbols, the number of zeros in each symbol is
a. odd
c. a and b both
b. even
d. unknown

19.Data communications monitors available on the software marked include
a. ENVIRON/1
c. BPL
b. TOTAL
d. Telnet

20.An example of an analog communication method is
a. laser beam
b. microwave
c. voice grade telephone line
d. all of the above

PART-B (5x2 = 10 Marks)

21. What is meant by TCP/IP?
22. Name the timers used in TCP congestion control.
23. Explain any one routing algorithm used in Internet.
24. What is mean by mutual exclusion?
25. What is the use of window advertisement in a TCP SEGMENT?

PART-C (5x14 = 70 Marks)

26. a) i)Explain in detail about Internetworking concepts. (4)

ii) Explain briefly about Network Architectural model (10)

(OR)

b) i)Write a note on Subnetting with example. (4)

ii) Describe ARP & RARP. (10)

27. a) Explain in detail about TCP Services and headers (14)

(OR)

b) Write a note on TCP Connection Establishment &Termination(14)

28. a) Write a note on Routing table & Algorithm (14)

(OR)

b) Explain in detail about ICMP Error Processing (14)

29. a) Explain in detail about Transmission Control Blocks (14)

(OR)

b) Write a note on Data Structure and Mutual Exclusion (14)

30. a) Explain in detail about Events and Messages for TCP (14)

(OR)

b) Explain in detail about Timer Process (14)

Reg.……

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Established Under Section 3 of UGC Act 1956)

COIMBATORE-641 021

(For the candidates admitted from 2008 onwards) – FULL TIME

B.E DEGREE EXAMINATION

COMPUTER SCIENCE AND ENGINEERING

TCP/IP DESIGN AND IMPLEMENTATION

Time: 3 Hours **Maximum: 100**

PART – A (20 x 1 = 20 Marks)

1.The interactive transmission of data within a time sharing system may be best suited to
a. simplex lines
c. full duplex lines
b. half-duplex lines
d. biflex-lines

2.Which of the following statement is incorrect?
a. The difference between synchronous and asynchronous transmission is the clocking derived from the data in synchronous transmission.
b. Half duplex line is a communication line in which data can move in two directions, but not at the same time.
c. Teleprocessing combines telecommunications and DP techniques in online activities
d. Batch processing is the prefered processing mode for telecommunication operation.

3.Which of hte following is considered a broad band communication channel?
a. coaxial cable
c. microwave circuits
b. fiber optics cable
d. all of above

4.Which of the following is not a transmission medium?

a. telephone lines
b. coaxial cables
c. modem
d. microwave systems

5.Which of the following does not allow multiple uses or devices to share one communication line?

a. doubleplexer
b. multiplexer
c. concentrator
d. controller

6.Which of the following signal is not standard RS-232-C signal?

a. VDR
b. RTS
c. CTS
d. DSR

7.Which of the following statement is incorrect?

a. Multiplexers are designed to accept data from several I/O devices and transmit a unified stream of data on one communication line
b. HDLC is a standard synchronous communication protocol.
c. RTS/CTS is the way the DTE indicates that it is ready to transmit data and the way the DCW indicates that it is ready to accept data
d. RTS/CTS is the way the terminal indicates ringing

8.Which of the following is an advantage to using fiber optics data transmission?

a. resistance to data theft
b. fast data transmission rate
c. low noise level
d. all of above

9.Which of the following is required to communicate between two computers?

a. communications software
b. protocol
c. communication hardware
d. all of above including access to transmission medium

10.In OSI network architecture, the dialogue control and token management are responsibility of

a. session layer
b. network layer
c. transport layer
d. data link layer

11.In OSI network architecture, the routing is performed by

a. network layer
b. data link layer
c. transport layer
d. session layer

12.Which of the following performs modulation and demodulation?

a. fiber optics
b. satellite
c. coaxial cable
d. modem

13.The process of converting analog signals into digital signals so they can be processed by a receiving computer is referred to as:

a. modulation
b. demodulation
c. synchronizing
d. digitising

14.How many OSI layers are covered in the X.25 standard?

a. Two
b. Three
c. Seven
d. Six

15.Layer one of the OSI model is

a. physical layer
b. link layer
c. transport layer
d. network layer

16.The x.25 standard specifies a

a. technique for start-stop data
b. technique for dial access
c. DTE/DCE interface
d. data bit rate

17.Which of the following communication modes support two-way traffic but in only one direction at a time?

a. simplex
b. half duplex
c. three-quarters duplex
d. all of the above

18.Which of the following might be used by a company to satisfy its growing communications needs?

a. front end processor
b. multiplexer
c. controller
d. **None Of the above**

19.What is the number of separate protocol layers at the serial interface gateway specified by the X.25 standard?

a. 4
b. 2
c. 6
d. 3

20.The transmission signal coding method of TI carrier is called

a. Bipolar
b. NRZ
c. Manchester
d. Binary

PART-B (5x2 = 10 Marks)

26. Give the advantages of CIDR .
27. Define socket.
28. How is fragmentation and reassembly of datagrams implemented?
29. Explain urgent data processing and push function.
30. How is congestion avoidance and control done in TCP?

PART-C (5x14 = 70 Marks)

26. a) Explain in detail about Internet Protocol & its routing. (14)

(OR)

b) Explain Routing in detail. (14)

27. a) Explain Keepalive timer in detail. (14)

(OR)

b) Explain Interactive data flow in detail. (14)

28. a) Explain in detail about Multicast processing used in IGMP (14)

(OR)

b) Write a note on IP software (14)

29. a) Describe Input processing in TCP. (14)

(OR)

b) Explain in detail about Finite State Machine implementation (14)

30. a) Describe Timer Process and Events used in TCP (14)

(OR)

b) Explain in detail about Push function used in TCP (14)

ONLINE 1 MARK QUESTIONS

The _____ is the physical path over which a message travels	Protocol	Medium	Signal	Amplitude
The information to be communicated in a data communications system is the _____	Medium	Protocol	Message	Transmission
Frequency of failure and network recovery time after a failure are the measures of the _____ of a network	Performance	Reliability	Security	Privacy
Which topology requires a central controller or hub	Mesh	Star	Bus	Ring
Which topology requires a multipoint connection	Mesh	Star	Bus	Ring
Which of the following transmission is used for communication between a computer and a keyboard	Simplex	Half- duplex	Full-duplex	Automatic
In a network with 25 computers , which topology would require the most extensive cabling	Mesh	Star	Bus	Ring
Which transmission is used in television broadcast	Half-duplex	Full-duplex	Simplex	Automatic
A Connection which provides a dedicated link between two devices is called as	Multipoint	Primary	Secondary	Point-to-point

In which transmission the channel capacity is shared by both communicating devices at all times	Half-duplex	Full-duplex	Simplex	Automatic
A cable break in a _____ topology stops all transmission	Mesh	Star	Primary	Bus
Which organization has authority over interstate and international commerce in the communication field	ITU-T	IEEE	FCC	ISO
Assume six devices are arranged in mesh topology. How many cables & ports for each device are needed	15 & 5	14 & 5	15 & 6	10 & 5
An unauthorized user is a network _____ issue	Performance	Reliability	Security	Flexibility
For n devices in a network, What is the number of cable links required for a Bus topology	n+1	n	n(n-1)/2	1
How many cables required in star topology for the network with n devices	1	n+1	n	n/2
Which of the following topology is support Robustness	Bus	Star	Ring	Primary
Which of the following is the fundamental characteristics for effective data communication	Delivery , Accuracy , Timeliness	Delivery , Reliable, Privacy	Reliable, Privacy , Security	Accuracy, Flexible, Protocol
Which of the following statement is True(i). Protocol is a set of rules(ii). Protocol represent an agreement between the communicating devices(iii). Without a protocol , 2 devices may be communicate but not connected	(i) & (iii)	(i) ,(ii) & (iii)	(i) & (ii)	(ii) & (iii)
Which of the following code that uses 32 bit pattern & represent 232 symbols	Unicode	ASCII	Extended ASCII	ISO
The size of the pixel per inch in image representation is called as	Picture Element	Resolution	Intensity	Bit pattern
What is the bit pattern used for a pixel in black and white image	one – bit	Three – bit	Two – bit	Eight – bit
In which of the following topology the unidirectional is weakness	Bus	Ring	Mesh	Star

The type of network that provides long distance transmission of data , voice, image and video information is referred as	Local area network	Metropolitan network	Lan – Lan network	Wide area network
A WAN that is wholly owned and used by a single company is referred as	Private company	Private network	Enterprise network	Intra network

Standards that have not been approved by an organized body but have adapted as standards through widespread use is called as	De jure	De facto	By law	By regulation
The ISO/OSI model consists of _____ layers	Three	Five	Seven	Eight
Which layer has the responsibility for the process - to - process delivery of the entire message	Transport	Network	Application	Physical
Which layer is closest to the transmission medium & transmitting individual bits	Transport	Network	Data link	Physical
Which layer provide mail services to the network users	Transport	Network	Application	Physical
As the data packet moves from the lower to upper layers headers are _____	Added	Subtracted	Rearranged	Modified
As the data packet moves from the upper to lower layers , headers are _____	Added	Removed	Rearranged	Modified
When data are transmitted from device A to device B, which of the B's layer read the header from A's layer 4	Physical	Transport	Application	Data link
The physical layer is concerned with the transmission of _____ over the physical medium	Programs	Dialogs	Protocols	Bits
Which of the ISO/OSI model changes bits into electromagnetic signals	Physical	Data link	Transport	Application
Which layer is used to establish, maintain and synchronize the interaction between communicating systems	Physical	Transport	Session	Application
The layer which handle the syntax and semantics of the information exchanged between the two systems is called as	Transport	Network	Data link	Presentation
Which of the following layer is designed for data translation , encryption, and compression	Presentation	Session	Application	Network

What is the main function of Data link layer in ISO/OSI model	Move packets	Create frame	Allow access to resource	Process to process delivery
The _____ layer lies between the network layer and presentation layer	Physical	Data link	Transport	Application
Which of the following is not the type of line coding	NPZ	NZ	Bipolar	Uni polar
The process of converting binary data , a sequence of bits, to a digital signal is called as	Encryption	Decryption	Line coding	Data coding
The number of values allowed in a particular signal & Number of values used to represent data is respectively called as	Signal level & Data level	Data level & Signal level	Bit level & Pulse level	Pulse level & Bit level
A signal has 2 data levels with a pulse duration of 1ms. What is the pulse rate & Bit rate	100 pulses/s & 1000 bps	1000 pulses/s & 1000 bps	100 pulses/s & 100 bps	1000 pulses/s & 100 bps
A signal has 4 data levels with a pulse duration of 1ms. What is the pulse rate & Bit rate	200 pulses/s & 2000 bps	2000 pulses/s & 2000 bps	1000 pulses/s & 2000 bps	1000 pulses/s & 100 bps
Which of the following statements are True (i).DC Component is undesirable (ii). The transformer does not allow a passage for DC component (iii). DC component is extra energy residing on the line	(i)	(i)&(ii)	(ii) &(iii)	(i),(ii)&(iii)
In a digital transmission , a receiver clock is 0.1 percent faster than the sender clock. How many extra bits per second does the receiver receive if the data rate is 1 Kbps	1	2	100	1000
In a digital transmission , a receiver clock is 0.1 percent faster than the sender clock. How many extra bits per second does the receiver receive if the data rate is 1 Mbps	1	100	1000	10000
The line coding scheme that uses only one voltage level for encoding process is called as	Bipolar	Polar	Uni polar	Tri polar
Which of the following encoding scheme has a transition at the middle of each bit	NRZ-L	RZ	NRZ-I	PZ-R

In _____ encoding , the transition at the middle of the bit is used for both synchronization and bit representation	Uni pilar	Manchester	Differential Manchester	Bi polar
Which encoding type always has a nonzero average amplitude	Uni polar	Polar	Bi polar	Manchester
Which of the following encoding methods does not provide for synchronization	NRZ-I	RZ	NRZ-L	NPZ