20CB502/22CM502/20CS502/20DS502/20IT502

Hall Ticket Number:

III/IV B.Tech (Regular/Supplementary) DEGREE EXAMINATION

December, 2024 **Common to CB, CM, CS, DS & IT**

Fifth Semester **Computer Networks**

Time: Three Hours **Maximum:** 70 Marks

			CO	BL	M
1	a)	List out any two differences between synchronous and asynchronous communication.	CO1	L1	1M
	b)	What is Internet?	CO1	L1	1M
	c)	List the categories of networks.	CO1	L1	1M
	d)	What is a full-duplex mode of communication?	CO1	L1	1M
	e)	Define congestion.	CO2	L1	1M
	f)	Define Flooding.	CO2	L1	1M
	g)	Write the purpose of router.	CO2	L1	1M
	h)	Write the objective of admission control.	CO2	L1	1M
	i)	What is a jitter?	CO3	L1	1M
	j)	What is a QoS?	CO3	L1	1M
	k)	List the advantages of IPv4.	CO3	L1	1M
	l)	What is a piggybacking?	CO4	L1	1M
	m)	Define multiplexing.	CO4	L1	1M
	n)	Suppose of URL is "www.becbapatla.ac.in.". Identify top level and second level domain names.	CO4	L4	1M
		Unit-I			
2	a)	Explain different types of components of a Data Communication System. List and explain different forms in which data may be represented.	CO1	L2	7M
	b)	What is Cyclic Redundancy Check (CRC)? Explain CRC encoder and decoder considering data word 101001111 and the divisor 10111. Generate the transmitted message at the sender and verify the correctness of the received message.	CO1	L4	7M
		(OR)			
3	a)	Explain about ISO/OSI reference model with neat sketch.	CO1	L2	7M
	b)	For n devices in a network, what is the number of code links required for a mesh, ring bus and star topology?	CO1	L3	7M
		Unit-II			
4	a)	Explain Sliding Window Flow Control with neat diagram.	CO2	L2	7M
	b)	Why we need datagram network? List the reasons for using datagram network. Compare with virtual circuit.	CO2	L2	7M
		(OR)			
5	a)	Explain the concept of Link State Routing Protocol with suitable example	CO2	L2	7M
	b)	What is load shedding in congestion control, and how does it work?	CO2	L2	7M
		Unit-III			
6	a)	Explain about different algorithms in networks to improve Quality of Service (QoS).	CO3	L2	7M
	b)	How can you justify different addresses as to be used for different networks in Internet and also explain the IPv4 header?	CO3	L3	7M
		(OR)			
7	a)	Explain ARP with an example.	CO3	L2	7M
	b)	DHCP can solve the problem of a shortage of addresses in an organization with different strategies. Explain.	CO3	L3	7M
		Unit-IV			
8	a)	Explain the purpose of Remote Procedure Call (RPC) mechanism.	CO4	L2	7M
	b)	Explain the congestion control in TCP.	CO4	L2	7M
		(OR)			
9	a)	Draw and explain TCP header format.	CO4	L2	7M
	b)	Explain the need of resource record and its format in DNS.	CO4	L2	7M

a) List out any two differences between synchronous and asynchronous communication. CO1 L1 1M

Synchronous Communication: Sender and receiver work in sync with a clock, sending data in fixed intervals.

Asynchronous Communication: Data is sent without a clock, using start and stop signals for each packet.

b) What is Internet? CO1 L1 1M

The **Internet** is a global network of interconnected computers that communicate using standardized protocols to share information and services. It allows access to websites, emails, social media, and other online resources.

c) List the categories of networks. CO1 L1 1M

Ans : The categories of networks are:

1. Local Area Network (LAN): Covers a small area like a home, office, or school.
2. Wide Area Network (WAN): Covers large areas, connecting multiple cities or countries.
3. Metropolitan Area Network (MAN): Covers a city or town, larger than LAN but smaller than WAN.
4. Personal Area Network (PAN): Connects personal devices over a short range (e.g., Bluetooth).
5. Virtual Private Network (VPN): Provides secure remote access over the internet.

d) What is a full-duplex mode of communication? CO1 L1 1M

Full-duplex mode is a communication system where data can be sent and received simultaneously between two devices. It works like a telephone call, allowing both parties to talk and listen at the same time.

e) Define congestion. CO2 L1 1M

Congestion in computer networks occurs when too much data is sent through a network, exceeding its capacity, which leads to delays, packet loss, or poor performance.

f) Define Flooding. CO2 L1 1M

Flooding is a network routing technique where a packet is sent to all nodes in the network, ensuring it reaches the destination by every possible path, often leading to high redundancy and network congestion.

g) Write the purpose of router. CO2 L1 1M

The **purpose of a router** is to connect different networks and direct data packets between them. It ensures that data reaches the correct destination by selecting the best path based on network conditions.

h) Write the objective of admission control. CO2 L1 1M

The **objective of admission control** is to regulate network traffic by deciding whether to accept or reject new connection requests. This ensures the network maintains quality of service (QoS) for existing users and prevents congestion.

i) What is a jitter? CO3 L1 1M

Jitter is the variation in the delay of data packets as they travel through a network. It can cause disruptions in real-time applications like video calls and online gaming.

j) What is a QoS? CO3 L1 1M

QoS (Quality of Service) in computer networks refers to the ability to prioritize and manage network traffic to ensure optimal performance for critical applications. It helps guarantee consistent bandwidth, low latency, and minimal packet loss.

k) List the advantages of IPv4. CO3 L1 1M

Advantages of IPv4:

1. Wide compatibility with devices and networks.
2. Simple addressing and easy configuration.
3. Stable and mature protocol.
4. 4.3 billion unique addresses available.
5. Supports NAT for private networks.

l) What is a piggybacking? CO4 L1 1M

Piggybacking is a technique where an acknowledgment (ACK) is sent along with the data in the same packet, instead of sending it separately. This improves efficiency by reducing the number of packets in the network.

m) Define multiplexing. CO4 L1 1M

Multiplexing is a technique used to combine multiple signals or data streams into one signal over a shared medium, allowing efficient use of the available bandwidth. It helps transmit multiple messages simultaneously.

n) Suppose of URL is "www.becbapatla.ac.in.". Identify top level and second level domain names. CO4 L4 1M
In the URL "**www.becbapatla.ac.in**":

- Top-Level Domain (TLD): .in
- Second-Level Domain (SLD): becbapatla.ac

UNIT-I

2a) Explain different types of components of a Data Communication System. List and explain different forms in which data may be represented. CO1 L1 7M

DATA COMMUNICATION Data Communication is a process of exchanging data or information In case of computer networks this exchange is done between two devices over a transmission medium. This process involves a communication system which is made up of hardware and software. The hardware part involves the sender and receiver devices and the intermediate devices through which the data passes. The software part involves certain rules which specify what is to be communicated, how it is to be communicated and when. It is also called as a Protocol. The following sections describes the fundamental characteristics that are important for the effective working of data communication process and is followed by the components that make up a data communications system.

Components of Data Communication A Data Communication system has five components as shown in the diagram below:

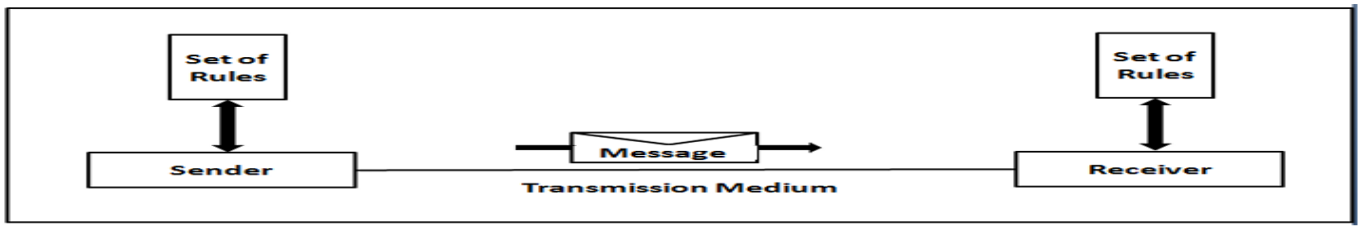


Fig. Components of a Data Communication System

1. Message Message is the information to be communicated by the sender to the receiver.

2. Sender The sender is any device that is capable of sending the data (message).
3. Receiver The receiver is a device that the sender wants to communicate the data (message).
4. Transmission Medium It is the path by which the message travels from sender to receiver. It can be wired or wireless and many subtypes in both.

5. Protocol 4 It is an agreed upon set or rules used by the sender and receiver to communicate data. A protocol is a set of rules that governs data communication. A Protocol is a necessity in data communications without which the communicating entities are like two persons trying to talk to each other in a different language without know the other language.

DATA REPRESENTATION

Data is collection of raw facts which is processed to deduce information. There may be different forms in which data may be represented. Some of the forms of data used in communications are as follows:

1. Text Text includes combination of alphabets in small case as well as upper case. It is stored as a pattern of bits. Prevalent encoding system : ASCII, Unicode
2. Numbers Numbers include combination of digits from 0 to 9. It is stored as a pattern of bits. Prevalent encoding system : ASCII, Unicode
3. Images ―An image is worth a thousand words‖ is a very famous saying. In computers images are digitally stored.

A Pixel is the smallest element of an image. To put it in simple terms, a picture or image is a matrix of pixel elements.

The pixels are represented in the form of bits. Depending upon the type of image (black n white or color) each pixel would require different number of bits to represent the value of a pixel.

The size of an image depends upon the number of pixels (also called resolution) and the bit pattern used to indicate the value of each pixel.

Example: if an image is purely black and white (two color) each pixel can be represented by a value either 0 or 1, so an image made up of 10 x 10 pixel elements would require only 100 bits in memory to be stored. On the other hand an image that includes gray may require 2 bits to represent every pixel value (00 - black, 01 – dark gray, 10 – light gray, 11 –white). So the same 10 x 10 pixel image would now require 200 bits of memory to be stored. Commonly used Image formats : jpg, png, bmp, etc

4. Audio Data can also be in the form of sound which can be recorded and broadcasted. Example: What we hear on the radio is a source of data or information. Audio data is continuous, not discrete.
5. Video Video refers to broadcasting of data in form of picture or movie

2b) What is Cyclic Redundancy Check (CRC)? Explain CRC encoder and decoder considering data word 101001111 and the divisor 10111. Generate the transmitted message at the sender and verify the correctness of thereceivedmessage. CO1 L2 7M

Cyclic Redundancy Check (CRC) is an error-detection method used in digital networks and storage devices to detect accidental changes to raw data. It works by appending a short, fixed-length binary sequence, known as the CRC code, to the data being transmitted. This sequence is calculated based on a polynomial division of the data.

The components of CRC include the data word, the divisor (generator polynomial), and the codeword. The data word is the binary message to be transmitted, such as `101001111`. The divisor is a predefined binary polynomial, such as `10111`. The codeword is the transmitted message, which is the data word concatenated with the CRC code.

To encode the message, first append `(n - 1)` zeros to the data word, where `n` is the length of the divisor. For the divisor `10111` (length = 5), append 4 zeros to the data word `101001111`, resulting in `1010011110000`. Perform binary division of the appended data word by the divisor using modulo-2 division to calculate the remainder. Modulo-2 division uses XOR instead of subtraction, and leading zeros are dropped during the process. For the given example, the final remainder is `1101`. Append this remainder to the original data word to create the codeword. The transmitted message becomes `1010011111101`.

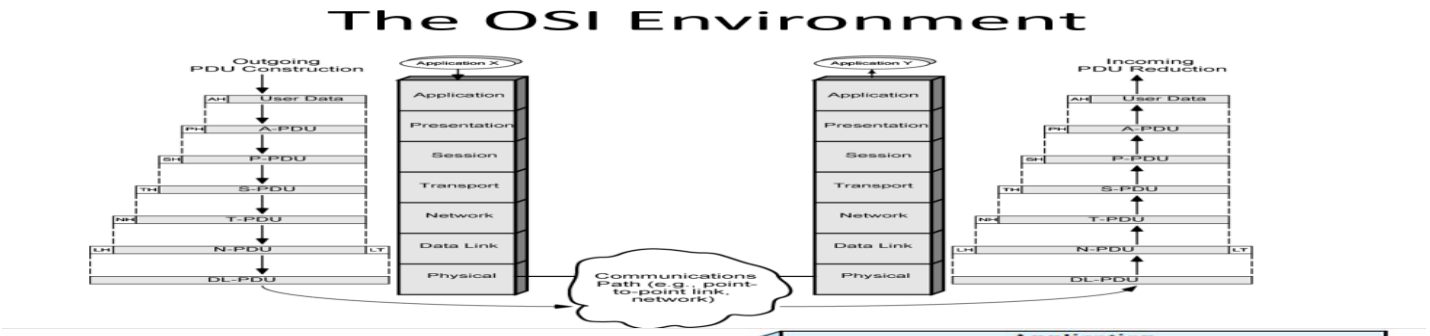
At the receiver, the same process is repeated. The received message is divided by the divisor using modulo-2 division. If the remainder is `0`, the received message is correct. If the remainder is non-zero, an error occurred during transmission. In this example, when the transmitted message `1010011111101` is divided by `10111`, the

remainder is `0`, confirming that the message is error-free.

The transmitted message generated using CRC is `1010011111101`. The receiver verifies the correctness of the received message by performing the same modulo-2 division. Since the remainder is `0`, the message is error-free.

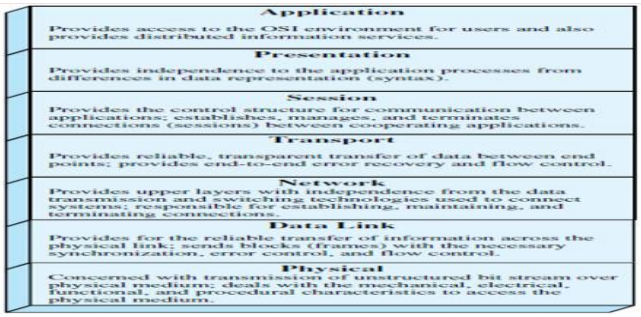
(OR)

3a) Explain about ISO/OSI reference model with neat sketch. **CO1 L2 10M**



OSI Layers

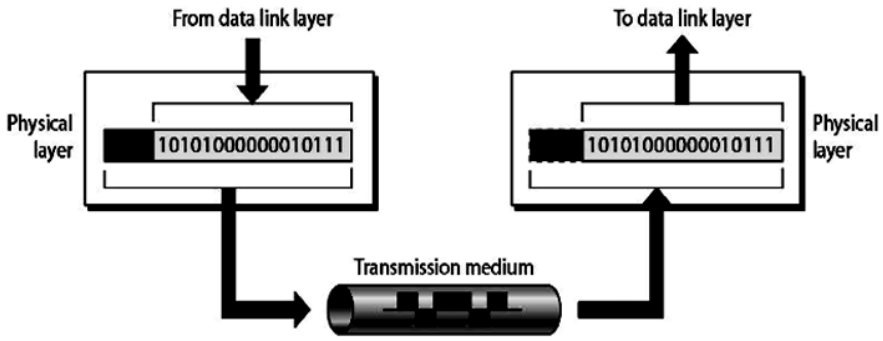
- Below figure illustrates the OSI model and provides a brief definition of the functions performed at each layer.
- The intent of the OSI model is that protocols be developed to perform the functions of each layer.



Description of Layers in the OSI Model

Physical Layer The Physical Layer provides a standardized interface to physical transmission media , including : a. Mechanical specification of electrical connectors and cables , for example maximum cable length b. Electrical specification of transmission line c. Bit-by-bit or symbol-by-symbol delivery

On the sender side, the physical layer receives the data from Data Link Layer and encodes it into signals to be transmitted onto the medium. On the receiver side, the physical layer receives the signals from the transmission medium decodes it back into data and sends it to the Data Link Layer as shown in the figure below:



Data Link Layer I. The Data Link layer adds reliability to the physical layer by providing error detection and correction mechanisms.

II. On the sender side, the Data Link layer receives the data from Network Layer and divides the stream of bits into fixed size manageable units called as Frames and sends it to the physical layer. On the receiver side, the data link layer receives the stream of bits from the physical layer and regroups them into frames and sends them to the Network layer. This process is called Framing. It is shown in the figure below:

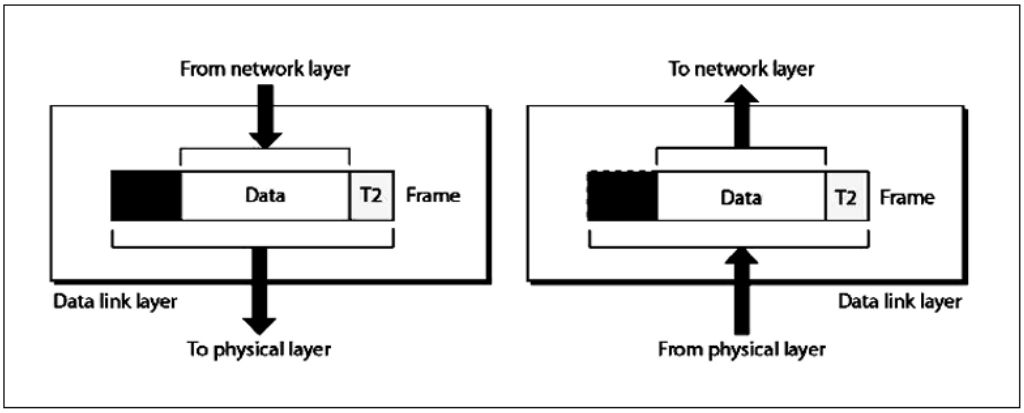


Fig: Data Link Layer: The process of Framing

Network Layer I. The network layer makes sure that the data is delivered to the receiver despite multiple intermediate devices.

The network layer at the sending side accepts data from the transport layer, divides it into packets, adds addressing information in the header and passes it to the data link layer. At the receiving end the network layer receives the frames sent by data link layer, converts them back into packets, verifies the physical address (verifies if the receiver address matches with its own address) and the send the packets to the transport layer.

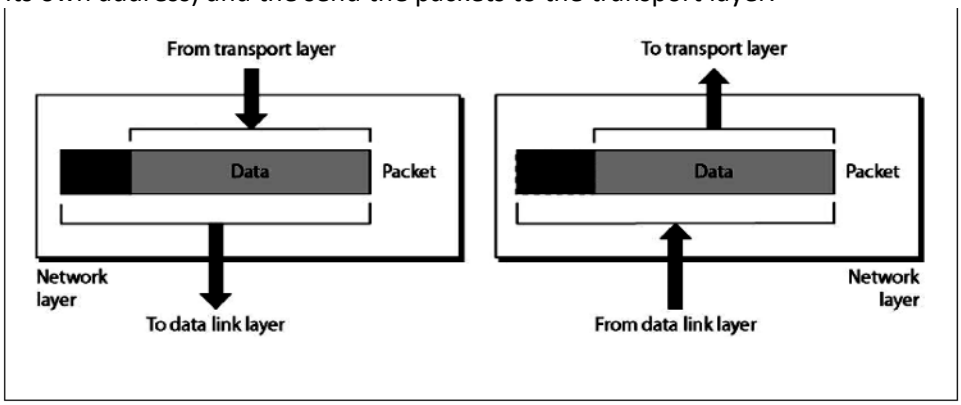


Fig: Network Layer

Transport Layer I. A logical address at network layer facilitates the transmission of data from source to destination device. But the source and the destination both may be having multiple processes communicating with each other. Hence it is important to deliver the data not only from the sender to the receiver but from the correct process on the sender to the correct process on the receiver. The transport layer takes care of process to process delivery of data and makes sure that it is intact and in order.

II. At the sending side, the transport layer receives data from the session layer, divides it into units called segments and sends it to the network layer. At the receiving side, the transport layer receives packets from the network layer, converts and arranges into proper sequence of segments and sends it to the session layer.

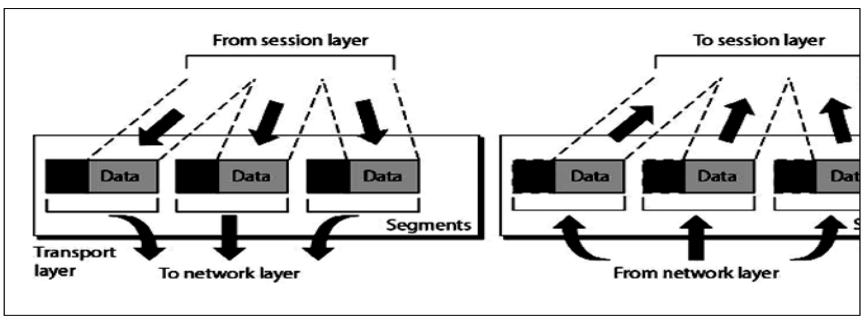
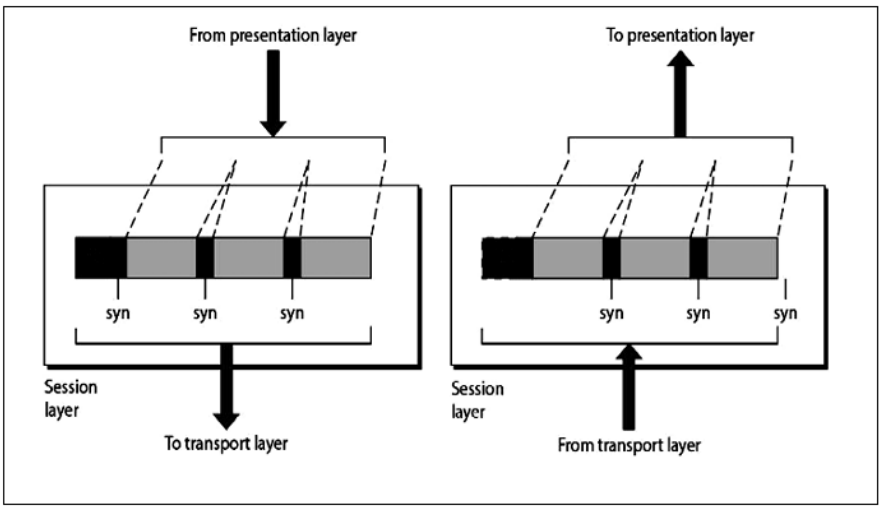


Fig: Transport Layer

Session Layer I.

The session layer establishes a session between the communicating devices called dialog and synchronizes their interaction. It is the responsibility of the session layer to establish and synchronize the dialogs. It is also called the network dialog controller.

II. The session layer at the sending side accepts data from the presentation layer adds checkpoints to it called syn bits and passes the data to the transport layer. At the receiving end the session layer receives data from the transport layer removes the checkpoints inserted previously and passes the data to the presentation layer.



Presentation Layer I. The communicating devices may be having different platforms. The presentation layer performs translation, encryption and compression of data.

II. 42 The presentation layer at sending side receives the data from the application layer adds header which contains information related to encryption and compression and sends it to the session layer. At the receiving side, the presentation layer receives data from the session layer decompresses and decrypts the data as required and translates it back as per the encoding scheme used at the receiver.

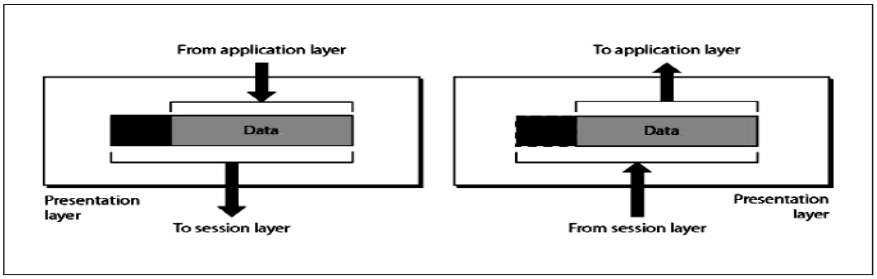


Fig : Presentation Layer

Application Layer I. The application layer enables the user to communicate its data to the receiver by providing certain services. For ex. Email is sent using X.400 service.

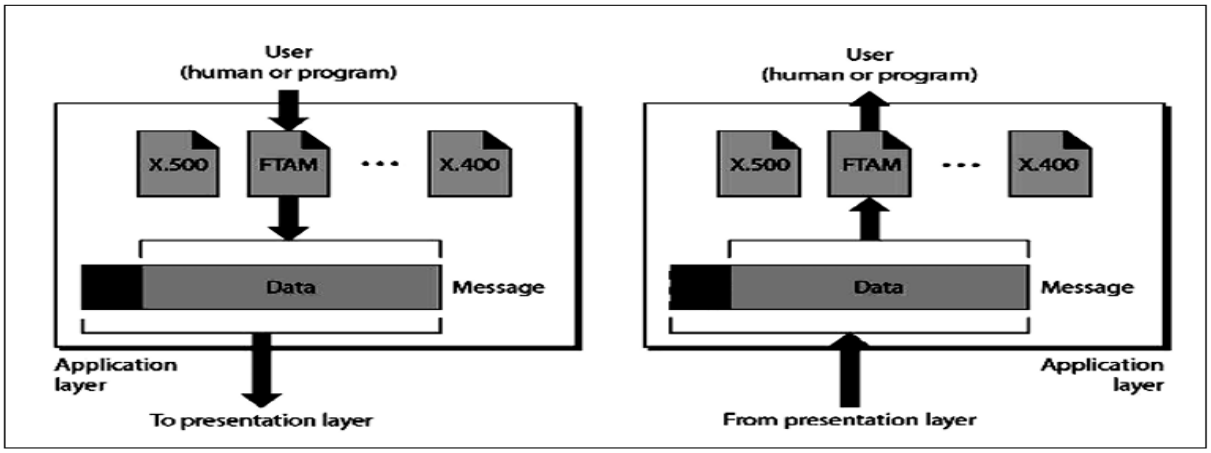


Fig : Application Layer

3b)For n devices in a network, what is the number of code links required for a mesh, ring bus and star topology? CO1 L3 7M

For a network of nn devices, the number of communication links required in different topologies is calculated as follows:

1. **MeshTopology**:
   In a mesh topology, every device is directly connected to every other device. The total number of links is given by the formula for combinations:

   Number of Links=n×(n−1)2\text{Number of Links} = \frac{n \times (n - 1)}{2}

2. **RingTopology**:
   In a ring topology, each device is connected to exactly two other devices, forming a closed loop. The number of links is:

   Number of Links=n\text{Number of Links} = n

3. **BusTopology**:
   In a bus topology, all devices share a single communication line (the bus). The number of links is:

   Number of Links=1\text{Number of Links} = 1

4. **StarTopology**:
   In a star topology, all devices are connected to a central hub. The number of links is:Number of Links=n\text{Number of Links} = n

Summary for nn Devices:

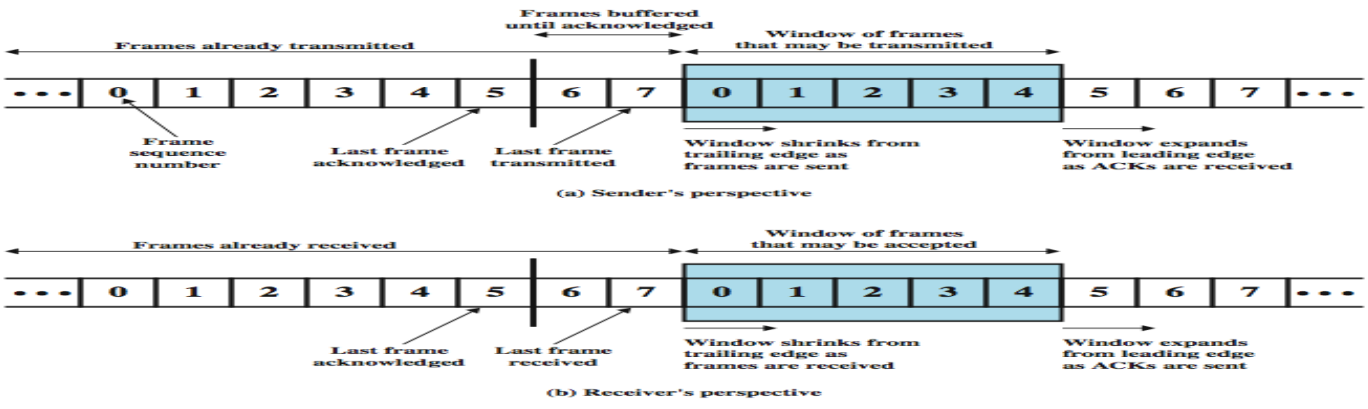
- Mesh Topology: n×(n−1)2\frac{n \times (n - 1)}{2}
- Ring Topology: nn
- Bus Topology: 11
- Star Topology: nn

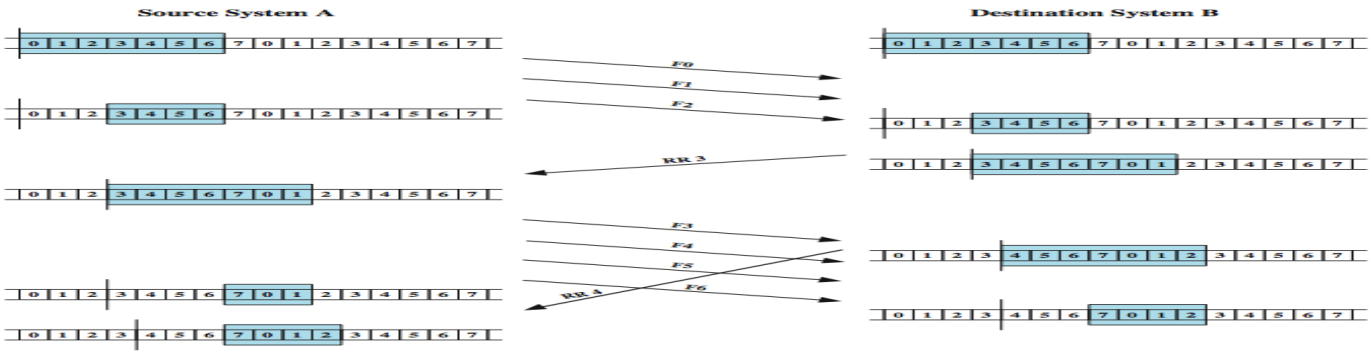
UNIT-II

4a) Explain Sliding Window Flow Control with neat diagram. CO2 L2 7M

Sliding-Window Flow Control

- The essence of the problem described so far is that only one frame at a time can be in transit.
- Efficiency can be greatly improved by allowing multiple frames to be in transit at the same time.
- Consider two stations, A and B, connected via a **full-duplex** link.
- Station B allocates buffer space for *W* frames. Thus, B can accept *W* frames, and A is allowed to send *W* frames without waiting for any acknowledgments.





- An example is shown in above Figure.
- The example assumes a 3-bit sequence number field and a maximum window size of seven frames.
- Initially, A and B have windows indicating that A may transmit seven frames, beginning with frame 0 (F0).
- After transmitting three frames (F0, F1, F2) without acknowledgment, A has shrunk its window to four frames and maintains a copy of the three transmitted frames.
- The window indicates that A may transmit four frames, beginning with frame number 3.
- B then transmits an **RR (receive ready)** 3, which means "I have received all frames up through frame number 2 and am ready to receive frame number 3; in fact, I am prepared to receive seven frames, beginning with frame number 3."
- With this acknowledgment, A is back up to permission to transmit seven frames, still beginning with frame 3; also A may discard the buffered frames that have now been acknowledged.
- A proceeds to transmit frames 3, 4, 5, and 6. B returns RR 4, which acknowledges F3, and allows transmission of F4 through the next instance of F2.
- By the time this RR reaches A, it has already transmitted F4, F5, and F6, and therefore A may only open its window to permit sending four frames beginning with F7.
- Most data link control protocols also allow a station to cut off the flow of frames from the other side by sending a **Receive Not Ready (RNR)** message, which acknowledges former frames but forbids transfer of future frames. At some subsequent point, the station must send a normal acknowledgment to reopen the window.
- If two stations exchange data, each needs to maintain two windows, one for transmit and one for receive, and each side needs to send the data and acknowledgments to the other. To provide efficient support for this requirement, a feature known as **piggybacking** is typically provided.
- Each **data frame** includes a field that holds the sequence number of that frame plus a field that holds the sequence number used for acknowledgment.

4b) Why we need datagram network? List the reasons for using datagram network. Compare with virtual circuit. **CO2 L2 7M**

- Datagrams are connection-less and the subnet corresponding to datagram is called datagram subnet.
- Router maintains the table which specify the destination and output link that is used to send datagram to that destination.

Congestion Control in Virtual-Circuit Subnets

- Various mechanisms that are used for controlling congestion in virtual circuit subnet are,
  - **New virtual circuit connection are not established into the subnet once congestion is triggered**. This strategy is known as controlling the admission of the connection called **admission control**. In this congestion already occurred (closed loop)
  - **Allowing virtual circuit connection even after congestion** but using different routes that are not congested.
  - **Reserving the resource in advance**. This strategy results in less utilization of the bandwidth.

Congestion control in Datagram subnets

- There are three ways for reducing the traffic. They are

a) The warning bit
b) Choke packets
c) Hop-by hop choke packet

The warning bit

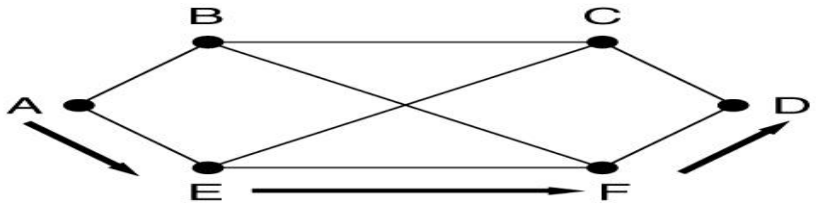
- The old DECNET architecture signaled the warning state by setting a special bit in the packet's header. So does frame relay.
- When the packet arrived at its destination, the transport entity copied the bit into the next acknowledgement sent back to the source.
- The source then cut back on traffic.

Choke packets

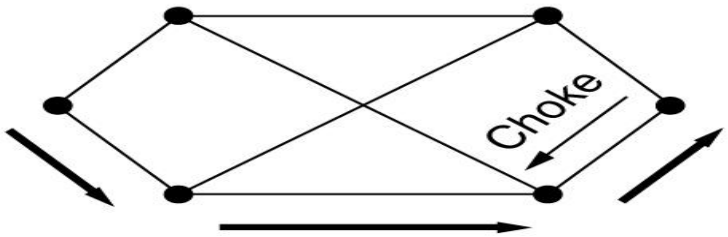
- The choke packets will have the effect of stopping (or) slowing down the rate of transmission from source and hence limit the total number of packets in the network.
- Whenever you moves above the threshold, the output line enter a "warning state". Each newly arriving packet is checked to see if the output line is in warning state. If so, the router sends a choke packet back to the source host giving its destination found in the packet.
- When the source hosts get the choke packet, it is required to reduce the traffic sent to the specified destination. Typically the first choke packet causes the data rate to be reduced to 50% of its previous rate, the next one causes a reduction to 25%, and so on.
- Increases are done in smaller increments to prevent congestion from reoccurring quickly.

Hop-by-Hop choke packets

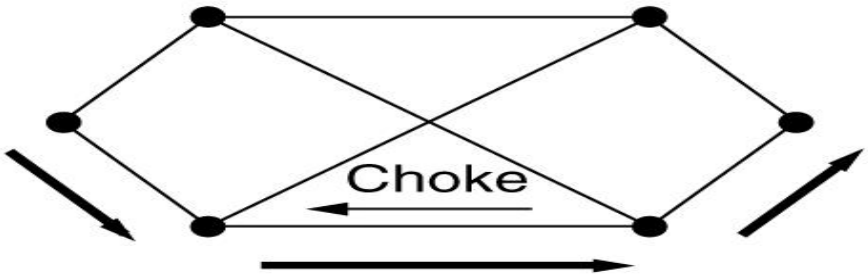
- At high speeds or over long distances, sending a choke packet to the source hosts does not work well because the reaction is so slow.
- Example: Congestion control using choke packets can be done by two ways.
- In first type, the choke packets affects only source as shown in figure(s).
- A subnet with 6 nodes A, B, C, D, E, F in below figure, here source node A and destination node is D.



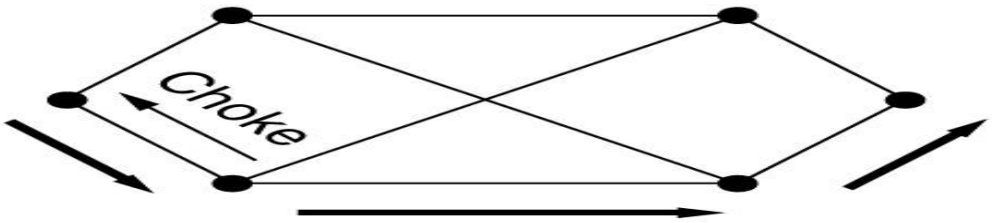
- When link utilization increased above its threshold, destination D starts sending choke packets towards source node A.



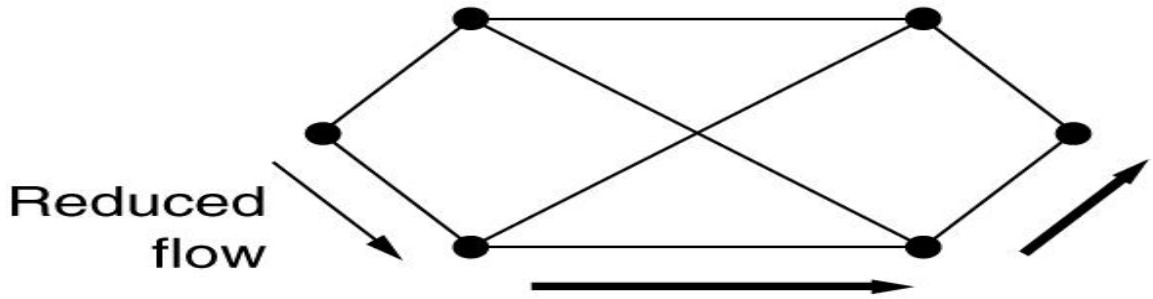
- The choke packet travels through source node A.



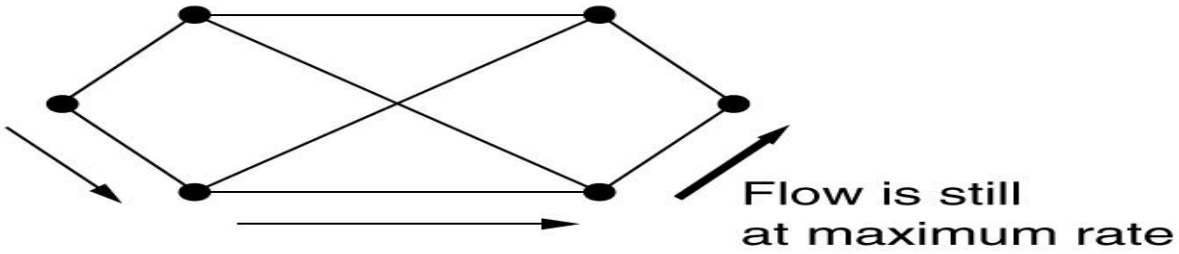
- After receiving first choke packet source node A reduces its flow towards destination



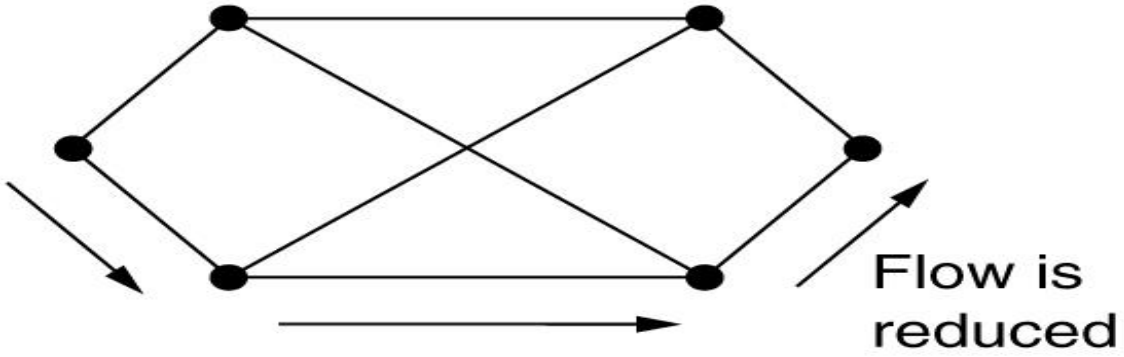
- The reduced packet flow follows the same reverse path.



- The reduced flow reaches to destination node D.



- Flow is reduced



- Another way of reducing congestion is that necessary action is taken at each hop to reduce the traffic towards the congested destination after receiving the choke packet. Using this method, congestion is reduced faster.

5a) Explain the concept of Link State Routing Protocol with suitable example

CO2 L2 7M

Link State Routing

- Link state algorithm is a dynamic routing algorithm that takes into account the complete topology, all delays and bandwidth when choosing routes.
- In this algorithm, each node sends only the state of the directly connected neighbors. But this information is sent to every node in the subnet.

In this algorithm, each router must do the following:

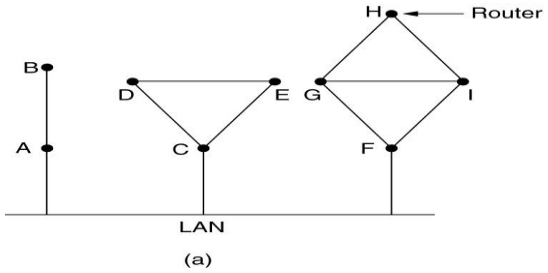
1. Discover its neighbors, learn their network address.
2. Measure the delay or cost to each of its neighbors.
3. Construct a packet telling all it has just learned.
4. Send this packet to all other routers.
5. Compute the shortest path to every other router.

- The link state algorithm uses the "Dijkstra's Algorithm" to find the shortest path.

Learning about the Neighbors

- When a router joins the network, it first learns about it neighbors. To achieve this, it sends a special HELLO packet on each outgoing line. When the packet arrive at the receiver, each receiver reply to it by telling its identity.

Learning about the Neighbors



5b) What is load shedding in congestion control, and how does it work? **CO2 L2 7M**

Load Shedding

- It is applied when none of the technique solve the congestion.
- Load shedding is a discarding policy in which packets can be discarded if the load of packets are not handled by the router.
- It is analogous to distribution of electricity to certain areas which is ON and OFF in some another area, so that the generated power can be distributed properly.

- When a packet arrives, which packet will be discarded is basically depends on type of application and their importance.
- There are two policies
  - Wine Policy →Old is better than new
  - Milk Policy →New is better than old

- While transferring a file, old packets are more valuable and of greater importance than new packets.
- If a new packet is accepted by discarding the old one, it will cause retransmission of packet from an old packet.
- This strategy is called the wine strategy.

Ex: File Transfer Application

- While transferring real-time data such as audio, video, multimedia packet, a new packet is more important than an old packet.
- This strategy is called milk strategy.

Ex: Multimedia Application

- Another useful way of discarding the packets is by assigning priorities to each packet. Packets with low priority will be discarded first and packets with very high priority will never be discarded.
- Ex - For ATM networks priority is marked with using the cell loss priority (CLP) in which value of "1" specifies that the cells of higher priority and the value of "0" specifies that the cell is lower priority and that it can be discarded.

UNIT-III

6a) Explain about different algorithms in networks to improve Quality of Service (QoS). CO3 L2 7M

Quality of Service

- The techniques we looked at in the previous sections are designed to **reduce congestion** and **improve network performance.**
- However, with the growth of multimedia networking, often these ad hoc measures are not enough.
- Serious attempts at guaranteeing quality of service through network and protocol design are needed.
- In the following sections we will continue our study of network performance, but now with a sharper focus on ways to provide a quality of service matched to application needs.
- It should be stated at the start, however, that many of these ideas are in flux and are subject to change.

Techniques for Achieving Good Quality of Service

- No single technique provides efficient, dependable QoS in an optimum way.
- Instead, a variety of techniques have been developed, with practical solutions often combining multiple techniques.
- We will now examine some of the techniques system designers use to achieve QoS.

➢ Overprovisioning
➢ Buffering
➢ Traffic Shaping
  ✓ The Leaky Bucket Algorithm
  ✓ The Token Bucket Algorithm

- Resource Reservation
- Admission Control
- Proportional Routing
- Packet Scheduling
  - ✓ Fair Queuing

Weighted Fair Queuing

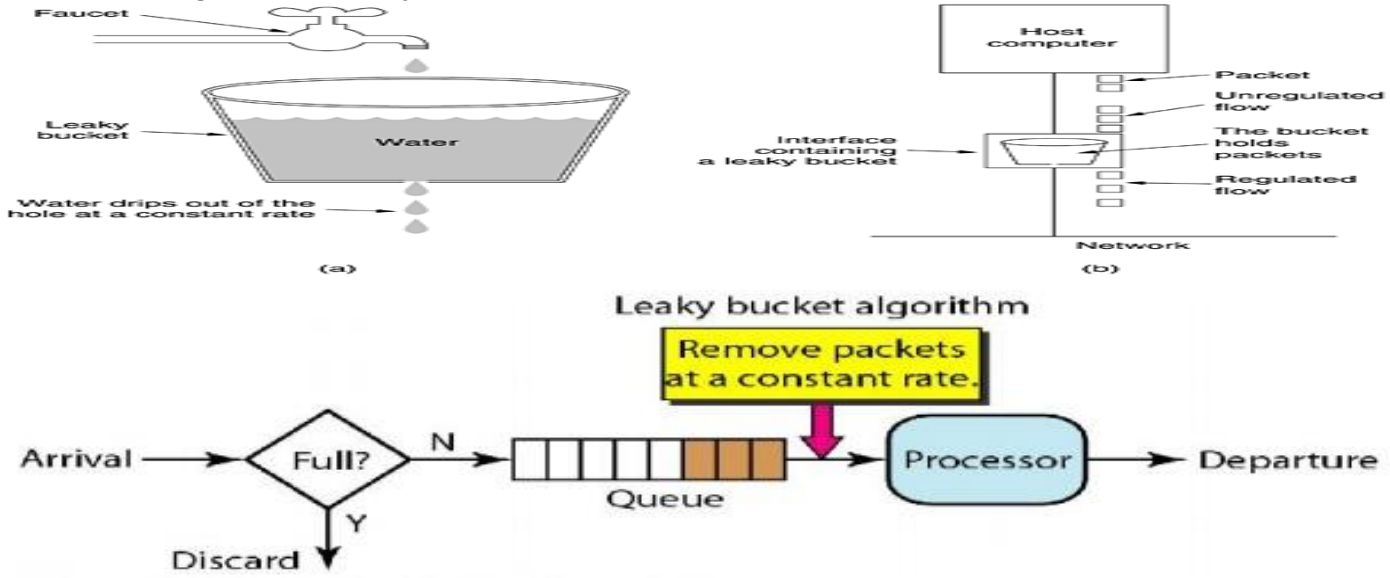
Traffic Shaping

- Traffic shaping is about regulating the average rate (and burstiness) of data transmission.
- Traffic shaping smooths out the traffic on the server side, rather than on the client side.
- When a connection is set up, the user and the subnet (i.e., the customer and the carrier) agree on a certain traffic pattern (i.e., shape) for that circuit. Sometimes this is called a **service level agreement.**
- As long as the customer fulfills her part of the bargain and only sends packets according to the agreed-on contract, the carrier promises to deliver them all in a timely fashion.
- Traffic shaping reduces congestion and thus helps the carrier live up to its promise.
- Such agreements are not so important for file transfers but are of great importance for real-time data, such as audio and video connections, which have stringent quality-of-service requirements.
- Two techniques can shape traffic:
  - Leaky bucket

Token bucket.

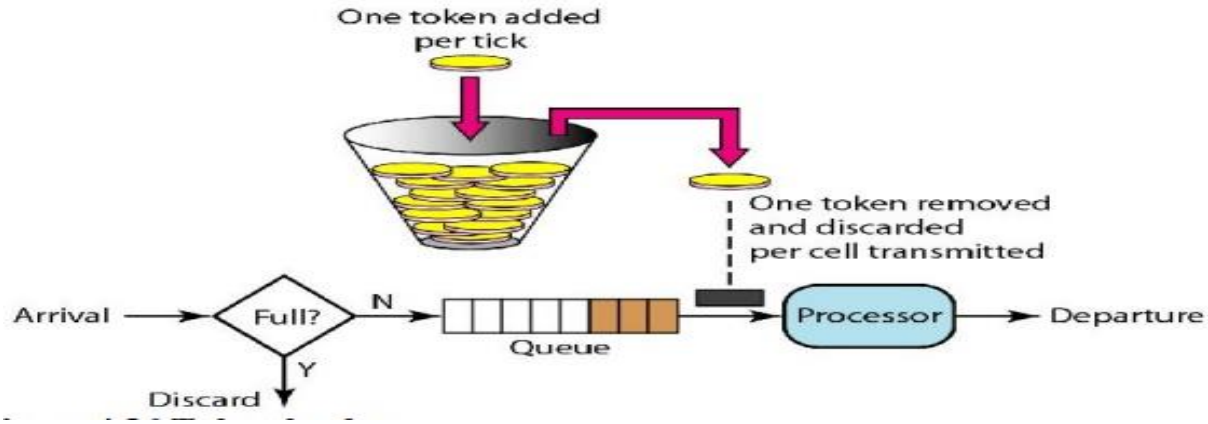
The Leaky Bucket Algorithm

- If a bucket has a small hole at the bottom, the water leaks from the bucket at a constant rate as long as there is water in the bucket.
- The rate at which the water leaks does not depend on the rate at which the water is input to the bucket unless the bucket is empty.
- The input rate can vary, but the output rate remains constant.
- Similarly, in networking, a technique called **leaky bucket** can smooth out bursty traffic.
- Bursty chunks are stored in the bucket and sent out at an average rate.
- Below figure shows a leaky bucket and its effects.



The Token Bucket Algorithm

- The leaky bucket is very restrictive. It does not credit an idle host.
- For example, if a host is not sending for a while, its bucket becomes empty.
- Now if the host has bursty data, the leaky bucket allows only an average rate.
- The time when the host was idle is not taken into account.
- The **token bucket algorithm** allows idle hosts to accumulate credit for the future in the form of tokens.
- For each tick of the clock, the system sends n tokens to the bucket.
- The system removes one token for every cell (or byte) of data sent.
- For example, if n is 100 and the host is idle for 100 ticks, the bucket collects 10,000 tokens.



- The token bucket can easily be implemented with a counter.
  - The token is initialized to zero. Each time a token is added, the counter is incremented by 1.
  - Each time a unit of data is sent, the counter is decremented by 1.
  - When the counter is zero, the host cannot send data.
- **The token bucket allows bursty traffic at a regulated maximum rate.**

Packet Scheduling

- If a router is handling multiple flows, there is a danger that one flow will hog too much of its capacity and starve all the other flows.
- Processing packets in the order of their arrival means that an aggressive sender can capture most of the capacity of the routers its packets traverse, reducing the quality of service for others.
- To thwart such attempts, various packet scheduling algorithms have been devised.
  - ✓ Fair Queuing
  - ✓ Weighted Fair Queuing

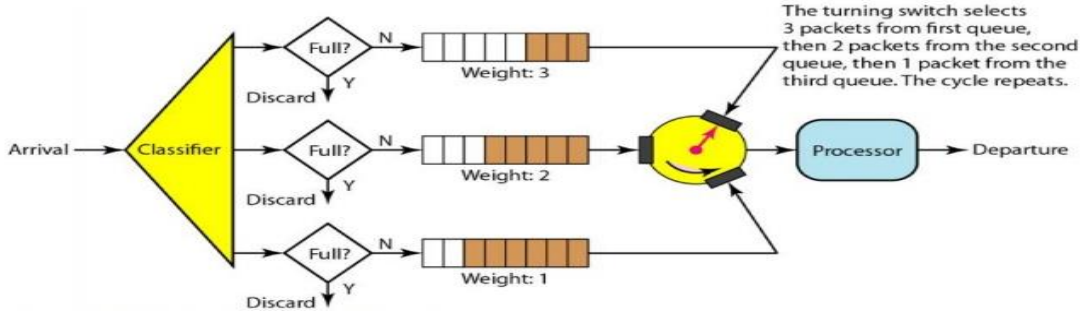
Fair Queuing

- The essence of the algorithm is that routers have separate queues for each output line, one for each flow.
- When a line becomes idle, the router scans the queues round robin, taking the first packet on the next queue.
- In this way, with n hosts competing for a given output line, each host gets to send one out of every n packets.
- Sending more packets will not improve this fraction.
- Although a start, the algorithm has a problem: it gives more bandwidth to hosts that use large packets than to hosts that use small packets.
- Demers et al. (1990) suggested an improvement in which the round robin is done in such a way as to simulate a byte-by-byte round robin, instead of a packet-by-packet round robin.
- In effect, it scans the queues repeatedly, byte-for-byte, until it finds the tick on which each packet will be finished.
- The packets are then sorted in order of their finishing and sent in that order.

The algorithm is illustrated in below figure.

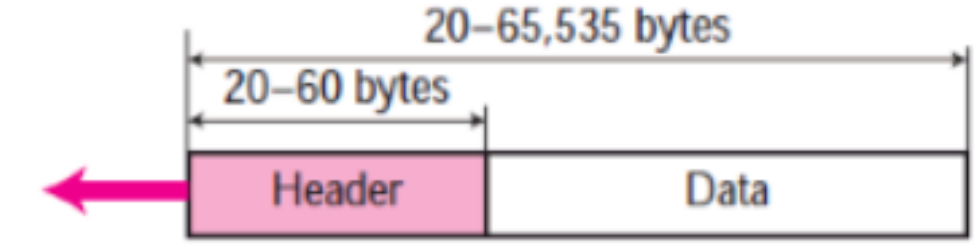
Weighted Fair Queuing

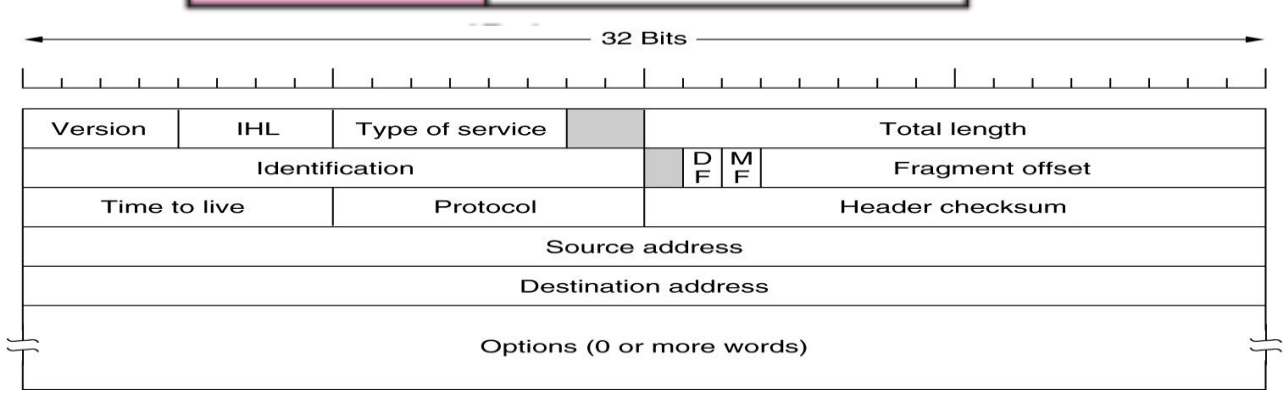
- One problem with fair queuing algorithm is that it gives all hosts the same priority.
- In many situations, it is desirable to give video servers more bandwidth than regular file servers so that they can be given two or more bytes per tick.
- This modified algorithm is called weighted fair queuing and is widely used.
- In this technique, the packets are still assigned to different classes and admitted to different queues.
- The queues, however, are weighted based on the priority of the queues; higher priority means a higher weight.
- The system processes packets in each queue in a round-robin fashion with the number of packets selected from each queue based on the corresponding weight.
- For example, if the weights are 3, 2, and 1, three packets are processed from the first queue, two from the second queue, and one from the third queue.
- If the system does not impose priority on the classes, all weights can be equal.
- In this way, we have fair queuing with priority.
- Below figure shows the technique with three classes.



6b) How can you justify different addresses as to be used for different networks in Internet and also explain the IPv4 header? CO3 L3 7M

The IPv4 (Internet Protocol) header





Version – This 4-bit field defines which version of the protocol the datagram belongs to. Current version of IP is IPV4 and the latest version of IP is IPv6.

- IHL – This 4-bit defines the length of the datagram header in 32-bit words. This field is needed because the length of the header is variable (20 & 60 Bytes).
- When there is no options, the header length is 20 bytes and the value of this field is 5 (5*4=20).
- When option field is at its maximum size, the value of this field is 15 (15*4=60).
- Type of services – This 8-bit field allows the host to tell the subnet what kind of service it wants.
- Various combinations of reliability and speed are possible.
    - For digitized voice, fast delivery beats accurate delivery.
    - For file transfer, error-free transmission is more important than fast transmission.
- This field contains a 3-bit Precedence field, 3 flags, D, T, and R &2 unused bits.
- The 3-bit Precedence field defines the priority of the datagram in case of congestion. Priority range from 0 (normal) to 7 (network control packet).
- Next 3-Flag bits allow the host to specify what it cared most about from the set {Delay, Throughput, Reliability}.

- Total length – This 16-bit field defines the total length of the IPv4 datagram (both header and data) in bytes. the maximum length of this field is 65,535 (2^16-1) bytes of which 20 to 60 bytes are the header and the remaining are data from the upper layer.
- **Identification** – This 16-bit field is needed to allow the destination host to determine which datagram a newly arrived fragment belongs to. All the fragments of a datagram contain the same Identification value.
- Next comes an **unused bit** and then **two 1-bit** fields.
- **DF (Don't Fragment)** – This 1-bit field is set to 1, means the routers must not fragment the datagram because the destination is incapable of putting the pieces back together again.
- **MF (More Fragments)** – This 1-bit field is set to 1, means the datagram is not the last fragment, there are more fragments after this one.
- **Fragment Offset** – This 13-bit field tells where in the current datagram this fragment belongs to.
- **Time to live** – It is 8-bit field. A datagram has a limited lifetime in its travel through an internet. It is supposed to count time in seconds, allowing a maximum lifetime of 255 sec. It must be decremented on each hop. When it hits zero, the datagram is discarded and a warning packet is sent back to the source host.
- **Protocol** – It is 8-bit field, specify to which transport layer protocol (TCP/UDP) the datagram is to be given.
- **Header checksum** – This 16-bit field verifies the header only. Such a checksum is useful for detecting errors generated by bad memory words inside a router.
    - **Note** - The Header checksum must be recomputed at each hop because at least one field always changes (Time to live field).
- **Source address** – This 32-bit field defines the IPv4 address of the source. This field remain unchanged during the time the IPv4 datagram travels from the source host to destination host.
- **Destination address** – This 32-bit field define the IPv4 address of the destination.
- options are not required for every datagram. They are used for **network testing and debugging**.

- IP provides several optional features, allowing a packets sender to set requirements on the path it takes through the network (source routing), trace the route a packet takes (record route), and label packets with security features.

(OR)

7a) Explain ARP with an example. CO3 L1 7M

Address Resolution Protocol (ARP) is a communication protocol used in computer networks to map an **IP address** to a device's **MAC address** within a local network (LAN). Since devices communicate using MAC addresses at the data link layer, ARP resolves the issue of finding the corresponding MAC address for a given IP address.

How ARP Works

1. **Request**: A device broadcasts an ARP request on the network asking, "Who has this IP address?"
2. **Reply**: The device with the matching IP address responds with its MAC address.

The ARP entry (IP-MAC mapping) is then cached for future use to reduce repetitive ARP requests.

Example

- **Scenario**: A computer (Host A) with IP address `192.168.1.2` wants to send data to another computer (Host B) with IP address `192.168.1.3` in the same network.
- **Steps**:
    1. Host A checks its ARP cache for the MAC address of `192.168.1.3`. If not found, it broadcasts an ARP request:
    2. `Who has 192.168.1.3? Tell 192.168.1.2`
    3. All devices on the network receive the request, but only Host B responds:
    4. `192.168.1.3 is at 00:1A:2B:3C:4D:5E`
    5. Host A stores this mapping (`192.168.1.3 ->00:1A:2B:3C:4D:5E`) in its ARP cache and uses the MAC address to send the data.

Key Points

- **Broadcasting**: ARP requests are broadcasted to all devices in the LAN.
- **Caching**: ARP entries are cached to improve efficiency.
- **Dynamic Updates**: ARP caches update dynamically but may timeout after a certain period.

Practical Example

Imagine two devices connected to a router in your home network:

- Host A: Laptop (IP: 192.168.0.10, MAC: AA:BB:CC:DD:EE:FF)
- Host B: Printer (IP: 192.168.0.20, MAC: 11:22:33:44:55:66)

To send a document from the laptop to the printer, ARP resolves `192.168.0.20` to its MAC address `11:22:33:44:55:66` so the data can be delivered at the data link layer.

7b)DHCP can solve the problem of a shortage of addresses in an organization with different strategies. Explain. CO3 L3 7M

Dynamic Host Configuration Protocol (DHCP) is a protocol used to automatically assign IP addresses and other network configuration parameters to devices on a network. It can help address the problem of **IP address shortages** in an organization by employing strategies like **dynamic allocation**, **IP address reuse**, and **efficient subnetting**.

Strategies Used by DHCP

1. **Dynamic Allocation of IP Addresses**:
   - DHCP dynamically assigns IP addresses to devices for a limited duration called the **lease time**.
   - Once a device disconnects or its lease expires, the IP address can be reclaimed and reassigned to another device.
   - This ensures efficient utilization of a limited pool of IP addresses.

   Example:

   - If the organization has 50 devices but only 30 IP addresses available, DHCP can assign IPs to only those devices currently active on the network, allowing reuse of addresses.

2. **Private IP Addressing with NAT**:
   - DHCP assigns private IP addresses (e.g., `192.168.x.x`) within an organization.
   - These private IPs are translated to a single public IP using **Network Address Translation (NAT)** when accessing external networks.
   - This minimizes the need for public IPs and reduces address exhaustion.

   Example:

   - Hundreds of devices can share a single public IP address while operating with unique private IP addresses inside the organization.

3. **Subnetting and Address Pool Segmentation**:
   - DHCP can divide the network into subnets, allocating a separate pool of IP addresses to each department or floor.
   - This prevents overlapping and ensures better address management.

   Example:

   - Floor 1: `192.168.1.x`
   - Floor 2: `192.168.2.x`

4. **Reclaiming Unused IP Addresses**:
   - DHCP keeps track of active devices and reclaims IPs that are no longer in use.
   - This prevents IP address hoarding and ensures that no address is wasted.

   Example:

   - If a guest device disconnects after its lease expires, its IP becomes available for another guest.

5. **DHCP Reservations**:
   - For critical devices (like servers or printers), DHCP can reserve specific IP addresses, leaving dynamic allocation for other devices.
   - This ensures efficient allocation for non-critical devices while maintaining stable configurations for important ones.

Advantages of DHCP in Address Management

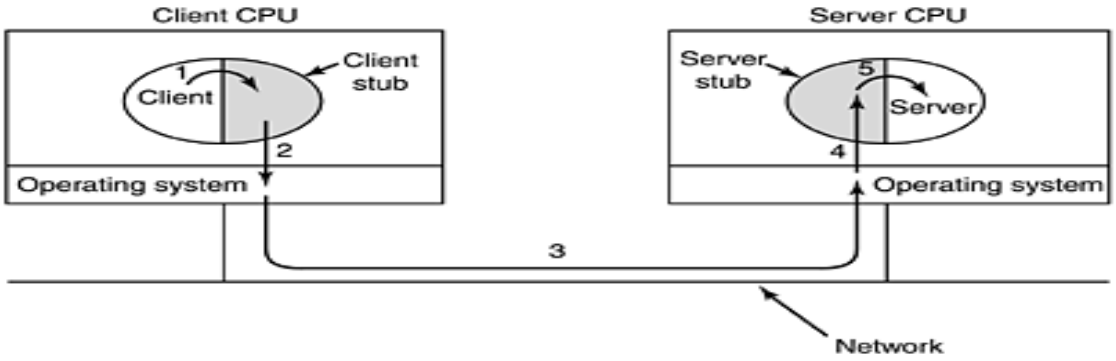
- **Scalability**: Supports large networks by automating IP management.
- **Flexibility**: Frees administrators from manually assigning addresses.
- **Address Reuse**: Optimizes limited address pools by reassigning unused IPs.

(OR)

8a) Explain the purpose of Remote Procedure Call (RPC) mechanism. CO4 L1 7M

Remote Procedure Call

- When a process on machine 1 calls a procedure on machine 2, the calling process on 1 is suspended and execution of the called procedure takes place on 2.
- Information can be transported from the caller to the callee in the parameters and can come back in the procedure result.
- No message passing is visible to the programmer.
- This technique is known as RPC (Remote Procedure Call).
- The calling procedure is known as the client and the called procedure is known as the server.
- The idea behind RPC is to make a remote procedure call look as much as possible like a local one.
- To call a remote procedure, the client program must be bound with a small library procedure, called the client stub, that represents the server procedure in the client's address space.
- Similarly, the server is bound with a procedure called the server stub.
- These procedures hide the fact that the procedure call from the client to the server is not local.
- The actual steps in making an RPC are shown in below figure.



- Step 1 is the client calling the client stub. This call is a local procedure call, with the parameters pushed onto the stack in the normal way.
- Step 2 is the client stub packing the parameters into a message and making a system call to send the message. Packing the parameters is called marshaling.
- Step 3 is the kernel sending the message from the client machine to the server machine.
- Step 4 is the kernel passing the incoming packet to the server stub.
- Finally, step 5 is the server stub calling the server procedure with the unmarshaled parameters. The reply traces the same path in the other direction.
- Disadvantages of RPC:
    - A First problem is the use of pointer parameters.
    - A second problem is that in weakly-typed languages, like C, it is perfectly legal to write a procedure that computes the inner product of two vectors (arrays), without specifying how large either one is.
    - A third problem is that it is not always possible to deduce the types of the parameters, not even from a formal specification or the code itself.
    - A fourth problem relates to the use of global variables.

8b) Explain the congestion control in TCP. CO4 L1 7M

TCP Congestion Control

- When the load offered to any network is more than it can handle, congestion builds up.
- The Internet is no exception.
- Although the network layer also tries to manage congestion, most of the heavy lifting is done by TCP because the real solution to congestion is to slow down the data rate.
- The first step in managing congestion is detecting it.
- In the old days, detecting congestion was difficult.
- A timeout caused by a lost packet could have been caused by either (1) noise on a transmission line or (2) packet discard at a congested router.
- Telling the difference was difficult.
- Before discussing how TCP reacts to congestion, let us first describe what it does to try to prevent congestion from occurring in the first place.

- When a connection is established, a suitable window size has to be chosen. The receiver can specify a window based on its buffer size. If the sender sticks to this window size, problems will not occur due to buffer overflow at the receiving end, but they may still occur due to internal congestion within the network.
- In the below figure, we see this problem illustrated hydraulically.
- In figure(a), we see a thick pipe leading to a small-capacity receiver. As long as the sender does not send more water than the bucket can contain, no water will be lost.
- In figure(b), the limiting factor is not the bucket capacity, but the internal carrying capacity of the network. If too much water comes in too fast, it will back up and some will be lost (in this case by overflowing the funnel).

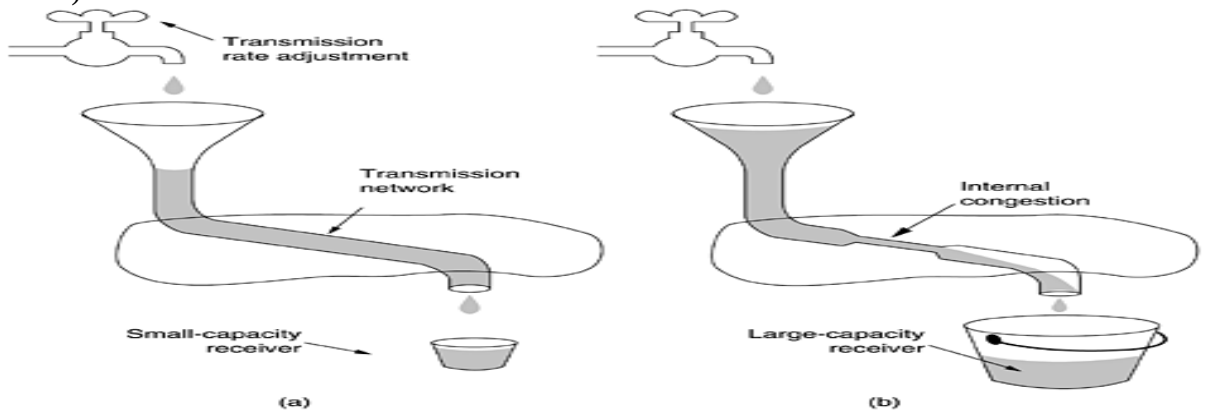


Figure:(a) A fast network feeding a low-capacity receiver

(b) A slow network feeding a high-capacity receiver

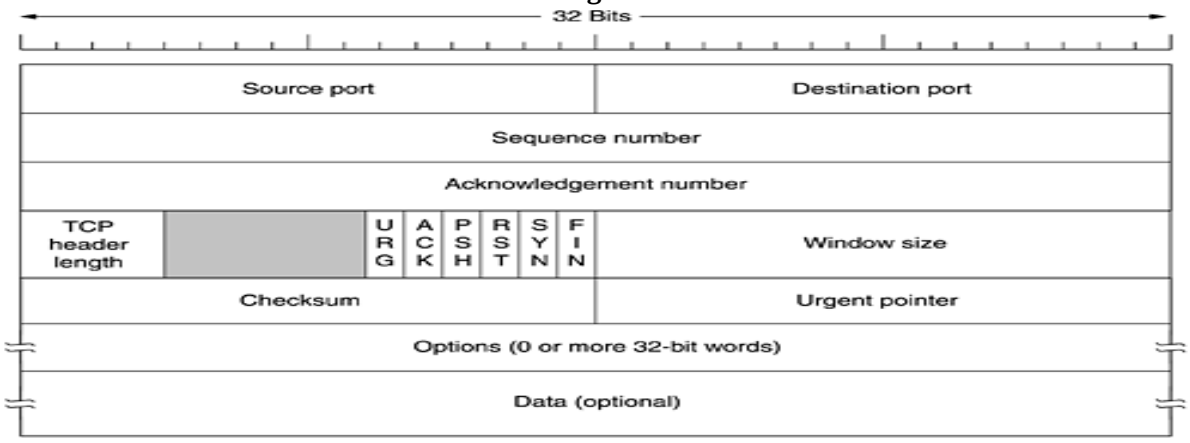
- When a connection is established, the sender initializes the congestion window to the size of the maximum segment in use on the connection.
- It then sends one maximum segment. If this segment is acknowledged before the timer goes off, it adds one segment's worth of bytes to the congestion window to make it two maximum size segments and sends two segments. As each of these segments is acknowledged, the congestion window is increased by one maximum segment size.
- When the congestion window is n segments, if all n are acknowledged on time, the congestion window is increased by the byte count corresponding to n segments. In effect, each burst acknowledged doubles the congestion window.
- The congestion window keeps growing exponentially until either a timeout occurs or the receiver's window is reached.
- The idea is that if bursts of size, say, 1024, 2048, and 4096 bytes work fine but a burst of 8192 bytes gives a timeout, the congestion window should be set to 4096 to avoid congestion.
- As long as the congestion window remains at 4096, no bursts longer than that will be sent, no matter how much window space the receiver grants.
- This algorithm is called **slow start,** but it is not slow at all. It is exponential. All TCP implementations are required to support it.
- Now let us look at the Internet congestion control algorithm.
- It uses a third parameter, the threshold, initially 64 KB, in addition to the receiver and congestion windows.
- When a timeout occurs, the threshold is set to half of the current congestion window, and the congestion window is reset to one maximum segment.
- Slow start is then used to determine what the network can handle, except that exponential growth stops when the threshold is hit.
- From that point on, successful transmissions grow the congestion window linearly (by one maximum segment for each burst) instead of one per segment.
- In effect, this algorithm is guessing that it is probably acceptable to cut the congestion window in half, and then it gradually works its way up from there.
- As an illustration of how the congestion algorithm works, see below figure.
- The maximum segment size here is 1024 bytes. Initially, the congestion window was 64 KB, but a timeout occurred, so the threshold is set to 32 KB and the congestion window to 1 KB for transmission 0 (zero) here.
- The congestion window then grows exponentially until it hits the threshold (32 KB). Starting then, it grows linearly.
- Transmission 13 is unlucky (it should have known) and a timeout occurs.
- The threshold is set to half the current window (by now 40 KB, so half is 20 KB), and slow start is initiated all over again.
- When the acknowledgements from transmission 14 start coming in, the first four each double the congestion window, but after that, growth becomes linear again.
- If no more timeouts occur, the congestion window will continue to grow up to the size of the receiver's window.

- At that point, it will stop growing and remain constant as long as there are no more timeouts and the receiver's window does not change size.
- As an aside, if an ICMP SOURCE QUENCH packet comes in and is passed to TCP, this event is treated the same way as a timeout.

(OR)

9a) Draw and explain TCP header format. **CO4 L2 7M**

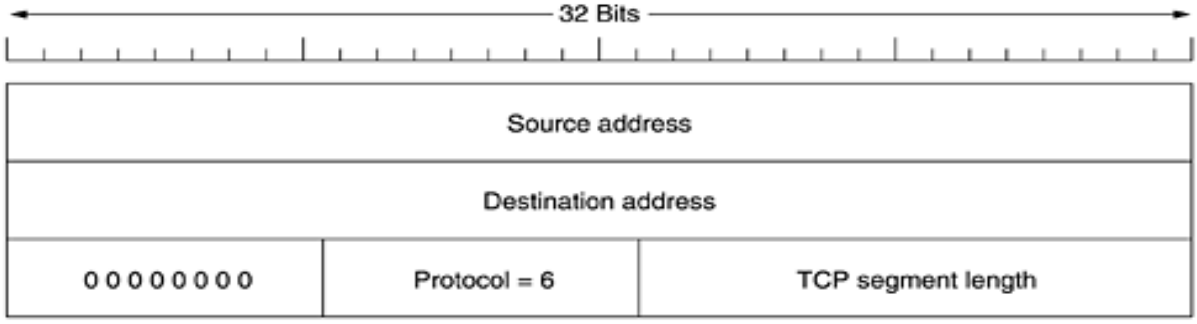
TheTCPSegmentHeader



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. 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 is not used.
- 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.
  - 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.
  - The **FIN** bit is used to release a connection. It specifies that the sender has no more data to transmit.
- 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 shown in below figure.
- 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.



- The pseudoheader contains the 32-bit IP addresses of the source and destination machines, the protocol number for TCP (6), and the byte count for the TCP segment (including the header).
- Including the pseudoheader in the TCP checksum computation helps detect misdelivered packets, but including it also violates the protocol hierarchy since the IP addresses in it belong to the IP layer, not to the TCP layer.
- UDP uses the same pseudoheader for its checksum.
- The **Options** field provides a way to add extra facilities not covered by the regular header.
- The most important option is the one that allows each host to specify the maximum TCP payload it is willing to accept.
- Using large segments is more efficient than using small ones because the 20-byte header can then be amortized over more data, but small hosts may not be able to handle big segments.
- During connection setup, each side can announce its maximum and see its partner's.
- If a host does not use this option, it defaults to a 536-byte payload.
- All Internet hosts are required to accept TCP segments of 536 + 20 = 556 bytes.
- The maximum segment size in the two directions need not be the same.

9b) Explain the need of resource record and its format in DNS. CO4 L2 7M

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.
- Normally, many records exist for each domain and each copy of the database holds information about multiple domains.
- This field is thus the primary search key used to satisfy queries.
- The order of the records in the database is not significant.
- The **Time_to_live** field gives an indication of how stable the record is.
- Information that is highly stable 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).
- The third field of every resource record is the **Class**.
- For Internet information, it is always IN.
- For non-Internet information, other codes can be used, but in practice, these are rarely seen.
- The **Type** field tells what kind of record this is. The most important types are listed in below figure.

Type	Meaning	Value
SOA	Start of Authority	Parameters for this zone
A	IP address of a host	32-Bit integer
MX	Mail exchange	Priority, domain willing to accept e-mail
NS	Name Server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
HINFO	Host description	CPU and OS in ASCII
TXT	Text	Uninterpreted ASCII text

- An **SOA** record provides the name of the primary source of information about the name server's zone (described below), the e-mail address of its administrator, a unique serial number, and various flags and timeouts.

- The most important record type is the **A** (Address) record.
- It holds a 32-bit IP address for some host.
- Every Internet host must have at least one IP address so that other machines can communicate with it.
- Some hosts have two or more network connections, in which case they will have one type A resource record per network connection (and thus per IP address).
- DNS can be configured to cycle through these, returning the first record on the first request, the second record on the second request, and so on.
- The next most important record type is the **MX** record.
- It specifies the name of the host prepared to accept e-mail for the specified domain.
- It is used because not every machine is prepared to accept e-mail.
- If someone wants to send e-mail to, for example, bill@microsoft.com, the sending host needs to find a mail server at microsoft.com that is willing to accept e-mail.
- The MX record can provide this information.
- The **NS** records specify name servers.
- For example, every DNS database normally has an NS record for each of the top-level domains, so, for example, e-mail can be sent to distant parts of the naming tree.
- **CNAME** records allow aliases to be created. For example, a person familiar with Internet naming in general and wanting to send a message to someone whose login name is paul in the computer science department at M.I.T. might guess that paul@cs.mit.edu will work.
- Actually, this address will not work, because the domain for M.I.T.'s computer science department is lcs.mit.edu.
- However, as a service to people who do not know this, M.I.T. could create a CNAME entry to point people and programs in the right direction.
- An entry like this one might do the job:

cs.mit.edu 86400 IN CNAME lcs.mit.edu

- Like CNAME, **PTR** points to another name.
- However, unlike CNAME, which is really just a macro definition, PTR is a regular DNS datatype whose interpretation depends on the context in which it is found.
- In practice, it is nearly always used to associate a name with an IP address to allow lookups of the IP address and return the name of the corresponding machine.
- These are called **reverse lookups.**
- **HINFO** records allow people to find out what kind of machine and operating system a domain corresponds to.
- Finally, **TXT** records allow domains to identify themselves in arbitrary ways.
- Both of these record types are for user convenience. Neither is required, so programs cannot count on getting them (and probably cannot deal with them if they do get them).
- Finally, we have the **Value** field.
- This field can be a number, a domain name, or an ASCII string.
- The semantics depend on the record type.
- A short description of the Value fields for each of the principal record types is given in above table.

S.NO	FACULTY NAME	COLLEGE NAME
1		
2		
3		