Unit -

Transport layer

Page No.
Date : / /

1) What are the major task of transport layer? Explain (1)

Services provided by transport layer? (1+1)

→ The transport layer is located between the application layer and the network layer. It provides a process-to-process communication between two application layers, one at local host and other at remote host.

Transport layer provides logical communication between application processes running on different hosts. Sender breaks application messages into segments, and passes to the network layer. Receiver reassembles segments into messages, passes to application layer. This layer ensures that data must be received in the same sequence in which it was seen. It provides end-to-end delivery of data between hosts.

Some of the main services are:-

- Process to process communication.
- Addressing
- Flow control & buffering
- Multiplexing & demultiplexing
- Congestion control.

Page No.
Date : / /

* Transport Protocols: UDP, TCP

2) Features of UDP protocol. (4+1)
In which case is UDP preferred as a transport layer protocol? Discuss with practical examples. (1)

→

UDP (User Datagram Protocol) is a connectionless, unreliable transport ~~tcp~~ level service protocol. In this protocol, there is no flow control, it doesnot provide packet sequencing.

It is used by applications that do not need a reliable transport service. The main advantage of UDP is speed and is usually used for real time traffics like video streaming or video chatting etc.

Features:

① No connection needed

UDP sends data without ~~sendir~~ setting up a connection first, making it faster but less reliable.

② Unreliable

It doesn't guarentee that the data will reach its destination, so some data might get lost.

Page No.
Date : / /

3) No flow control

UDP sends data as fast as possible without checking if the receiver can handle it, which can cause ~~congo~~ overcrowding.

4) low overhead

UDP has a small header, making it lightweight and efficient for simple tasks.

5) Broadcast & Multicast

It can send data to multiple devices at once, either to all devices (broadcast) or to a specific group (multicast).

6) Simple & lightweight

Source Port	Destination Port
length	Checksum

Fig:- UDP header format

- Source port

This field is an optional field. When meaningful, it indicates the port of the sending process and asumed to be the port to which a reply should be addressed If the field is not used, a value of zero is inserted It is a Byte long field.

Page No.
Date : / /

- **Destination Port**

This field identifies the destination port and is required. It is a 2 byte long field.

- **length**

It is the size in bytes of the UDP packet including the header and data. The minimum length is 8 bytes, the length of the header zone.

- **Checksum**

The 2 byte long checksum field is used for error checking of the header & data.

Page No.
Date: / /

* TCP (Transmission control protocol)

1) What is TCP connection? (1)
2) Explain TCP segment structure (1)
3) Explain TCP datagram format in detail (1)
4) For the client server application over TCP, why must the server program be executed before the client program. (1)
5) Explain TCP protocol with its header. (1+1)
6) Explain connection establishment & termination in TCP (1+1+1)
7) Explain how a TCP connection can be gracefully terminated. (1)
8) Why TCP is known as reliable protocol & describe how reliability is provided by TCP? (1+1)

The transmission control protocol (TCP) is one of the most important protocols of internet protocols suite. It is most widely used protocol for data transmission in communication network such as internet. The length of TCP header is minimum 20 bytes long and maximum 60 bytes.

Source port (16 Bit)								Destination port (16 Bit)
Sequence number (32 Bit)								
Acknowledgement number (32 Bit)								
TCP header length		URG	ACK	PSH	RST	SYN	FIN	Window size (16 bit)
Checksum (16 bit)								Urgent pointer (16 bit)
Options & Padding (upto 40 Bytes)								

Fig:- TCP header format

1) Source port

This is a 16-bit field that holds the port address of the application that is sending the data segment.

2) Destination port address

This is a 16 bit field that holds the port address of the application that is receiving the data segment.

Page No.
Date : / /

3) Sequence number

This is a 32 bit field that holds the sequence number. It is used to reassemble the message at the receiving end if the segments are received out of order.

4) Acknowledgement number

This is a 32 bit field. When ACK flag is set, this number contains the next sequence number of the data byte expected and works as acknowledgement of the previous data received.

5) Header length (HLEN)

This is a 4-bit field that indicates the length of the TCP header by number of 4-byte words in the header. The value of this field is always between 5 and 15.

6) Control Flags

These are six 1-bit control bits that control connection establishment, connection termination, connection abortion, flow control, mode of transfer etc. Their function is

→ URG : Urgent pointer is valid.
→ ACK : Acknowledgement number is valid.
→ PSH : Request for push

Page No.
Date : / /

→ RST : Reset the connection
→ SYN : Synchronize sequence numbers
→ FIN : Terminate the connection (Finish)

7) Window size

This field tells the window size of the sending TCP in bytes.

8) Checksum

This field holds the checksum for error control.

9) Urgent pointer

It points to the urgent data byte if URG flag is set to 1.

10) Options

This field provides additional options which are not covered by the regular header.

Page No.
Date : / /

4)

In a client-server application using TCP, it's crucial to start the server program before the client program because the server needs to be ready & waiting to accept incoming connections from client.

When the server program starts first, it gets ready to listen for clients who want to connect. It's like opeaning a shop before customers arrive. The server sets up its ~~ports~~ door where clients can knock to enter and talk. If the server isn't open yet, the doors are closed & clients can't get in, they'll see a message saying they can't connect. So starting the server first ensures it's ready to welcome clients.

On the other hand, clients depend on the server to provide services or data. When a client program starts, it attempts to connect to the server using its address and port number. If the server program isn't running, the client won't find anyone on the other end to respond to its connection request.

Page No.
Date : / /

6)

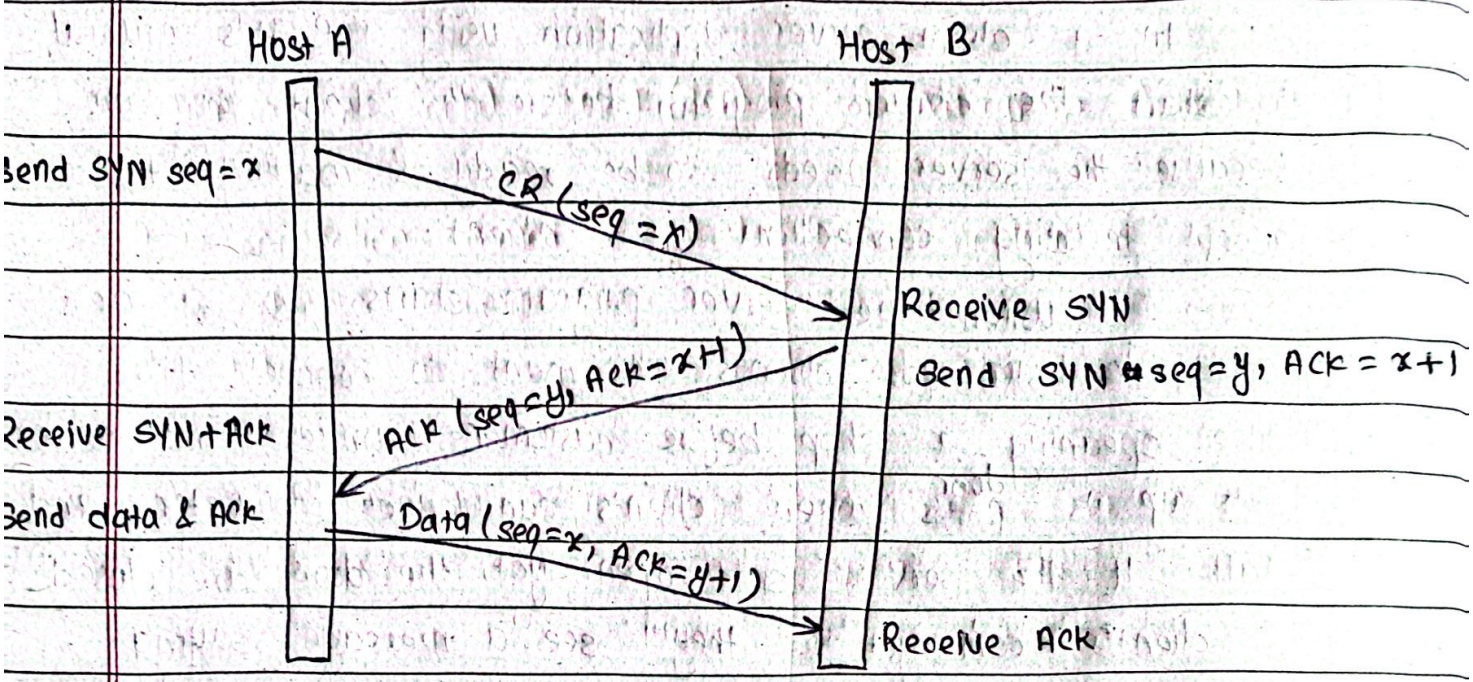


Fig: TCP connection established using 3 way handshaking

To establish a connection between Host A and Host B, first Host A sends a TCP segment for connection request towards Host B. At the same time SYN field get set. Consider initial sequence number (ISN) is 'x'.

Host B sends SYN and ACK to host A when host B ~~has~~ receive a SYN from Host A. The response given by host B has its own ISN. During the response from Host B to Host A SYN field gets set and ACK field is set to value $x+1$.

After delivery of Host B ACK and ISN at Host A, Host A tries to end connection establishment by

responding final acknowledgement. This time ACK field is set to value y+1.

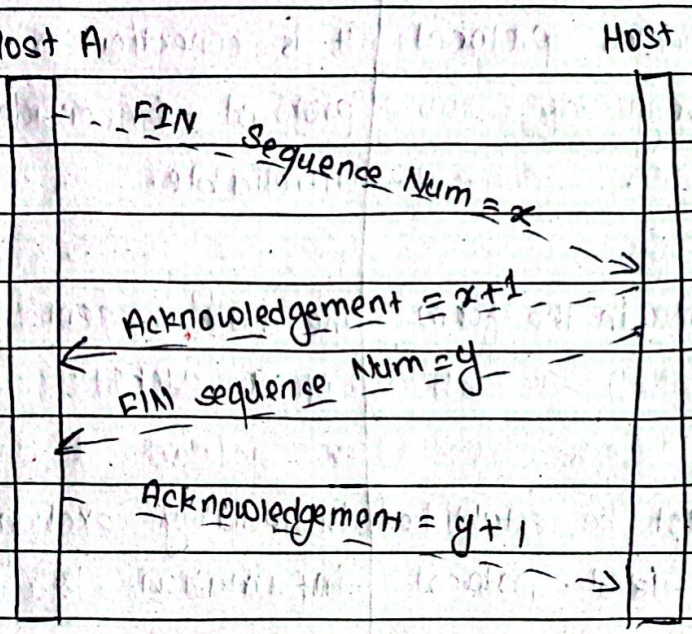


Fig:- Connection termination

In this case, SYN field is replaced with FIN control field to terminate the connection.

To close an established connection, Host A need to send a closing signal over TCP. Hence Host A produce a FIN signal and sends it to Host B. Host B generates acknowledgement signal and send towards host A to notify the termination request of destination. When Host B decided to terminate the connection, it generate FIN signal and send it towards Host A which will processed by Host A.

Again Host A gives a response with Ack.

Page No.
Date : / /

Q) Difference Between TCP and UDP (1+1+1)

	Transmission Control Protocol (TCP)	User Datagram Protocol (UDP)
–	It is connection-oriented protocol and reliable delivery of messages.	It is conectionless oriented protocol & considered as unrellable.
–	The data is sent in the form of byte-stream.	The data is sent in the form of a packet.
–	The connection must be established before application-level protocol to exchange the information.	Immediately exchange the information by application level protocol.
–	Guaranteed delivery due to error correction mechanism.	No error correction mechanism so unsecure communication.
–	Used to send important data such as web pages, database information etc.	It has high speed to deliver the data so used for real time audio, video exchange.
–	Only concerns with accuracy so the speed is automatically slow.	only concerned with speed but not accuracy.
–	Well known applications are :- FTP, Telnet, HTTP.	well known applications are DNS, DHCP, SNMP.

Page No.
Date : / /

Q) Why is it necessary to standarize the port number for well known server? (1)

Why port number is used in networking? (1)

What is port number? (1)

→

A port number is a numerical identifier used in computer networking to specify a particular process or service within a device.

It helps distinguish between different services or applications running on the same computer or server.

There are two types of port:-

i) TCP port

ii) UDP ports.

Uses

(i) Identify specific services

Diff. applications & services run on the same device can use the same IP address. Port numbers help distinguish between these services.

(ii) Efficient Data handling

By using port numbers, a computer can handle multiple network connections at once.

Page No.
Date: / /

iii) Facilitate Communication

Port numbers make it easier for devices to communicate over the-internet.

iv) Organization & security

Port numbers help organize network traffic & can also enhance security.

Q) What happens when a web service is hosted at some different port such as 8765 instead of 80? (1)

→

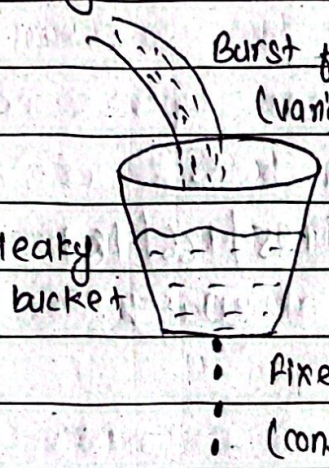
When a web service is hosted on a non-standard port, such as 8765 instead of the standard port 80, users need to specify the port number in the URL to access the web ~~servers~~ service.

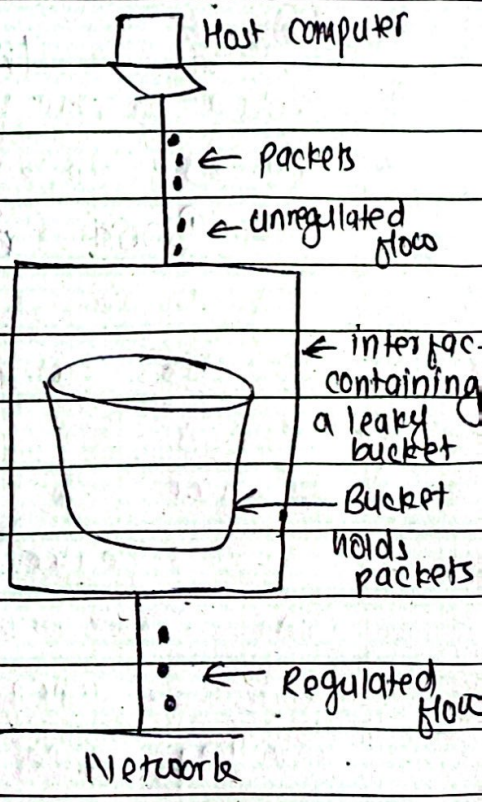
For example, instead of typing 'http://google.com', they must type 'http://google.com:8765'. If the port number is not included, the browser will default to port 80 & fail to connect to the service.

Page No.
Date : / /

* Leaky Bucket Algorithm (1+1)

It is an algorithm used to control congestion in network traffic. It uses a similar technique to a leaky bucket.





Book page 185.