Q1)

a) Illustrate the principles of network application architecture.

→ The applications architecture is designed by application developer and dictates how the application developed is structural over the various end systems. There are two architectural paradigms used in modern network applications.

① Client - Server architecture:-

This architecture has an always on host called server which services request from many other hosts called clients. The server has a fixed, well known address called IP address. A client can always contact the server by sending a packet to the server's IP address.

② P2P architecture:-

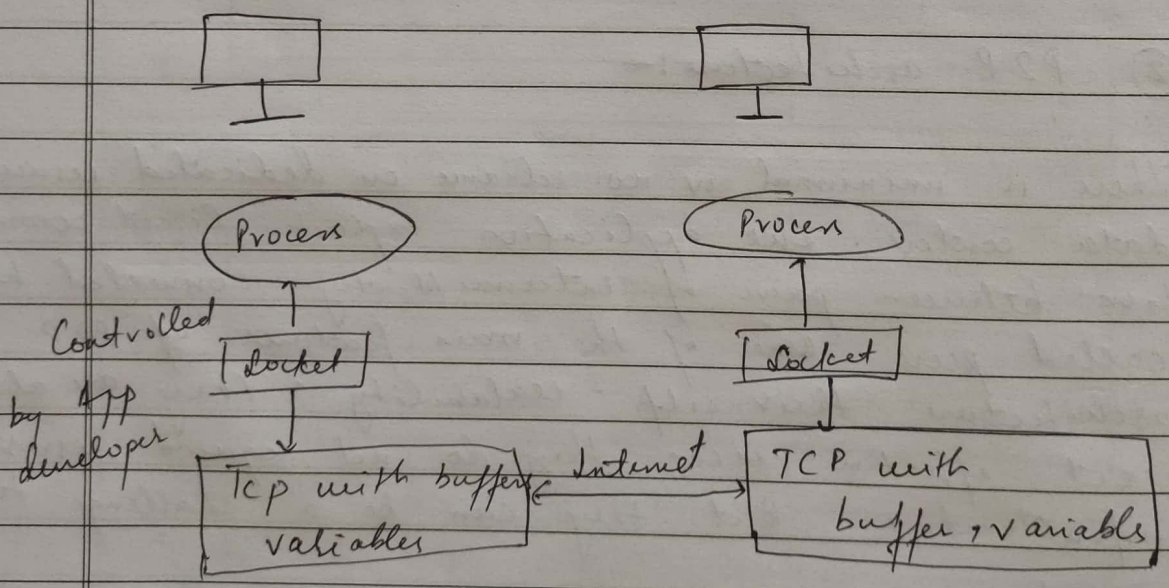
There is minimal or no reliance on dedicated servers in data centers. The application exploits direct communication between pairs of intermittently connected hosts called peers. One of the main features of P2P architecture their self - scalability. They are also cost effective since they do not required server infrastructure. But they can be a challenge to secure

example → Bit torrent, Xunlei, (Internet telephony)

Q1> b) Define Socket. Describe the interface between the process and computer network.

A socket is a software interface through which a process sends messages into and recieves messages from the network.

Let a process be analogous to house and socket to its door when a process wants to send a message to another process on another host, it shoves the message out through the socket. This sending process assumes not that there is a transportation infrastructure on the other side of its door that will transport the message to the door of destination process.



Application Processes, Sockets, Transport Protocol.

As shown in the figure, a socket is the interface between application layer and transport layer. It is also called as the Application Program Interface (API). between the applications network, since socket is the programming interface with which network applications are built

Q2) a.) Explain the transport services provided by the internet.

The internet makes two transport protocols available to applications: UDP and TCP. Each of these protocols offers a different set of services to the invoking applications.

TCP services:

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

1) Connection-Oriented Service:

TCP has the client and server exchange transport layer control information with each other before the application layer message begin to flow. This is called hand shaking

procedure alerts the client and server, allowing them to prepare for an onslaught of packets. After the handshaking phase, TCP connection is said to exist between sockets of the two process. The connection is a full-duplex connection that the two process can send messages to each other over the connection at the same time when the application finishes sending messages.

(2) Reliable data transfer

The communicating process can rely on TCP to deliver all the data sent without error and in the proper order. When one side of the application passes a stream of bytes into the recieving socket, it can count on TCP to deliver the same stream of bytes to the recieving socket, with no missing or duplicate bytes.

UDP Services:

UDP is a no frills, lightweight transport protocol, providing minimal services. UDP is connectionless, so there is no handshaking before the two processes start to communicate. UDP provides an unreliable data transfer service that is, when a processes sends a message into a UDP socket, UDP can provide no gavantee that the message will ever reach the recieving process.

Q2) What is web page? Explain HTTP with Non-persistent
b) and persistent connection.

A web page, also known as a document, consists of object. An object is simply a file such as an HTML file, a JPEG image, a video is addressable by a single URL. Most web pages consists of a base HTML file and several referenced objects. Each URL has two components: the hostname of the server that houses the object and the object paths name.

HTTP with non persistent connection:

Let's suppose a web page consists of a base HTML file and 10 JPEG images, and that all 11 of these objects reside on the same server. Further suppose the URL for the base HTML file is.

What happens:-

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

2) The HTTP client sends an HTTP request message to the server via its socket, The request message includes the path name: / some department / home.index

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

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

⑤ The HTTP client receives the response message. The TCP connection terminates. The message indicates that the encapsulated object is an HTML file.

⑥ The first four steps are then repeated for each of the referenced objects (JPEGs).

HTTP with persistent connection:

- The server leaves the TCP connection open after sending a response message subsequent requests and responses between the same client and server can be sent over the same connection.

- Multiple web pages residing on the same server can be sent from the server to the same client over a single persistent TCP connection. These requests can be made back-to-back, without waiting for replies to pending requests. (pipelining). Typically, the HTTP server closes a

connection when it is not used for a certain time. The default mode of HTTP uses persistent connection with pipelining

Q3) a) Give HTTP message format with diagram and explanation.

→ There are two types of HTTP messages: request and response messages

HTTP Request Message

Below is a typical HTTP request messages.

GET / Somedir / page.html HTTP / 1.1
Host: www.someschool.edu
connection: close
User agent: Mozilla/5.0
Accept language: fr.

① The message is written is ordinary ASCII text. The message consists of five lines, each followed by a carriage return and line feed.

② The first line of a HTTP request message is called the request line, the subsequent lines are called header lines. The request line has three fields. The method field, the URL field and the HTTP version field.

③ The method field can take on several different values, including GET, POST, HEAD, PUT and DELETE.

④ The header line HOST specified the host on which the object resides.

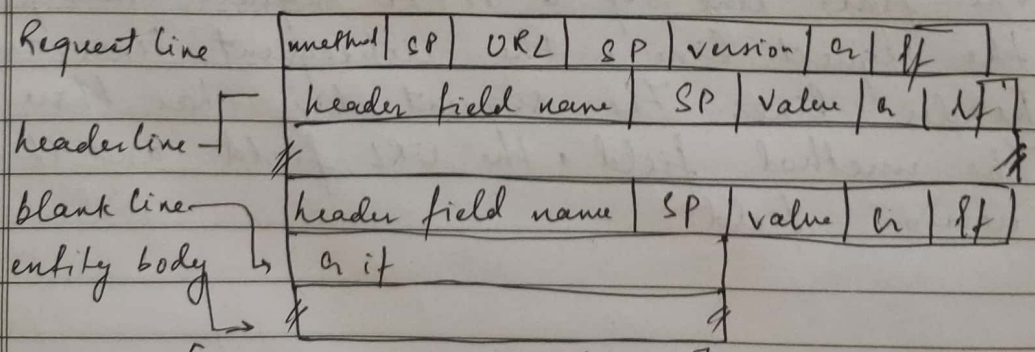
⑤ The connection close header line tells the server that it does not wish to bother with persistent connection.

⑥ The user agent header line specifies the browser type that is making request to the server. Here, the user agent is a firefox browser.

⑦ The head method is responde with HTTP message but leaves out requested object. It is often used for debugging.

⑧ The PUT method is used in conjunction with web publishing tools.

⑨ The Delete method allows a user, or an application, to delete an object as a web browser.



[HTTP Request Message].

Q3)

b) Explain in brief about POP3, IMAP, EMAIL, SMTP.

POP3:

→ It is an extremely simple mail access protocol. It is defined in [RFC1939] which is short and quite readable. Its functionality is limited. POP3 begins with agent opens a TCP connection to the mail server. With the TCP connection established, POP3 progress through three phases: authorization, transaction and update:

example :-

telent mail server 110
+OK POP3 server ready
user bob
+OK
pass hungry
+OK user successfully logged on.

During a POP3 session between a user and agent and mail server, the POP3 server maintains some state information. It keeps track of which user messages have been marked deleted. The POP3 server does not carry state information accross POP3 sessions.

IMAP:

POP3 protocol does not provide any means for a user to create remote folders and assign messages to the folders. IMAP is a mail access protocol defined in [RFC 3501] which solves this problem. It has more features than POP3 but is also more complex

- ~~An~~ IMAP protocol provides commands to allow users to create users to search remote folders for messages matching specific criteria

EMAIL:

With EMAIL, the user agent is an ordinary web browser and the user communicates with its remote mailbox via HTTP. When a recipient wants to access a message in his mailbox, the E-mail message is sent from recipients mail server to browser using HTTP protocol. When a ~~sender~~ sender wants to send an email, the message is sent from the browser to mail server over HTTP rather than over SMTP.

SMTP:

SMTP, defined in [RFC 5321] is the heart of Internet electronic mail. Suppose Alice wants to send Bob a simple ASCII message.

SMTP uses persistent connection for each message, the client begins with new MAIL FROM: crys.fr designate end of message with period and issues Quit after all messages are sent.

Q4) Explain in brief: cookies and web caching

a)

Cookies defined in [RFC 6265], allows sites to keep track of users.

Cookie technology has four components:

- A Cookie header line in HTTP response message.
- A cookie header line in HTTP request message
- A back-end database at the website.

When request comes into the Amazon server, the server creates a unique identification number and creates an entry in its back end database that is indexed by the identification number. The amazon then responds to susan's home browser, including HTTP response a set cookie header, which contains the identification number.

Web Caching

A web cache, also called as a proxy server is network entity that satisfies HTTP requests on behalf of an origin web server. The web cache has its own disk storage and keeps copies of recently requested objects in the storage

- A web cache reduces the response time for client request. It can also reduce traffic on an institution's access links to the internet.

Q4) b) Explain the overview of how DNS works?

Suppose that some application running in a user's host needs to translate a hostname to an IP address. The application will involve the client side of DNS, specifying the hostname that needs to be translated.

The process of DNS resolution involves converting a hostname (www.example.com) into a computer friendly IP address. When a user wants to load a webpage, a translation must occur between what a user types into their web browser (example.com) and the machine-friendly address necessary to locate the example.com webpage.

- The four DNS server involved in loading a web page.
- DNS recursor
- Root nameserver → The root server
- TLD nameserver → The top level domain server (TLD)
- Authoritative nameserver → The final nameserver.

Q5) Describe UDP segment structure and checksum.

→ The UDP header is 8 bytes fixed and simple header, while for TCP it may vary from 20 bytes to 60 bytes first 8 bytes contain all necessary header information and remaining part consists of data. UDP fields are each 16 bits long, port number 0 is reserved

8 bytes.

UDP Header	UDP data

Source port 16 bits	destination Port 16 bits
length 16 bits	Check sum 16 bits
Application data / message	

- Source port: It is 2 byte long field used to identify port number of source.
- Destination port: It is 2 byte long field used to identify the port of destined packet
- length: It is the length of UDP including header and data.

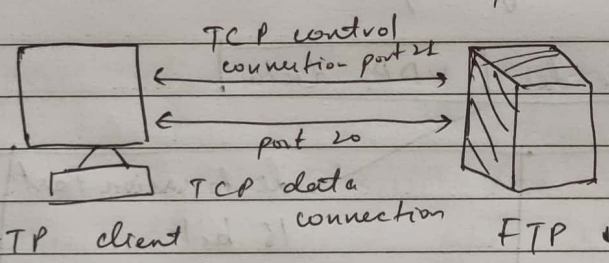
Check sum:

→ It is a 2 bytes long field. It is the 16 bit one's component of the one's complement sum of UDP header, pseudo header of information from the IP header and the data, Its calculation is not mandatory in UDP.

Q6) Explain FTP control and data Connection.

a)

In a typical FTP session, user is sitting infront of one host and wants to transfer files to or from a remote host. In order for the user to access the remote account, the user must provide a user identification and password.



[Control and Data Connection]

FTP uses two parallel TCP connections to transfer a file:

Control connection and data connection. The Control Connection is used for sending control information between the two hosts information such as user identification, password, commands to change remote directory, and commands to "put" and "get" files.

- The data connection is used to actually send a file because FTP uses a seperate control connection. FTP is said to send its control information out of band.

1) When a user starts an FTP session with a remote host, the client side of FTP (user) first initiates a control TCP connection with the server side (remote host) on server port 21.

2) Client side of FTP sends user identification and password over this control connection

3) When server side recieves a command for file transfer over the control connection, the server side initiates the TCP data connection to the client side. FTP sends exactly one file over the data connection and then closes the data connection.

4) Throughout the session, FTP server must maintain state about the user.

Q6) Explain FTP commands and replies.

b)

The commands, from client to server, and replies from server to client are sent across the control connection in 7 bit ASCII format. Thus, like HTTP commands, FTP commands are readable by people.

A carriage return and line feed and each command.

- User username: used to send user identification to server.
- PASS password: used to send password to server.
- LIST: used to ask server to send back a list of all files in current remote directory.
- RETR filename: used to retrieve file from directory.
- STOR filename: used to store (put) a file into current directory.

Some common replies are:-

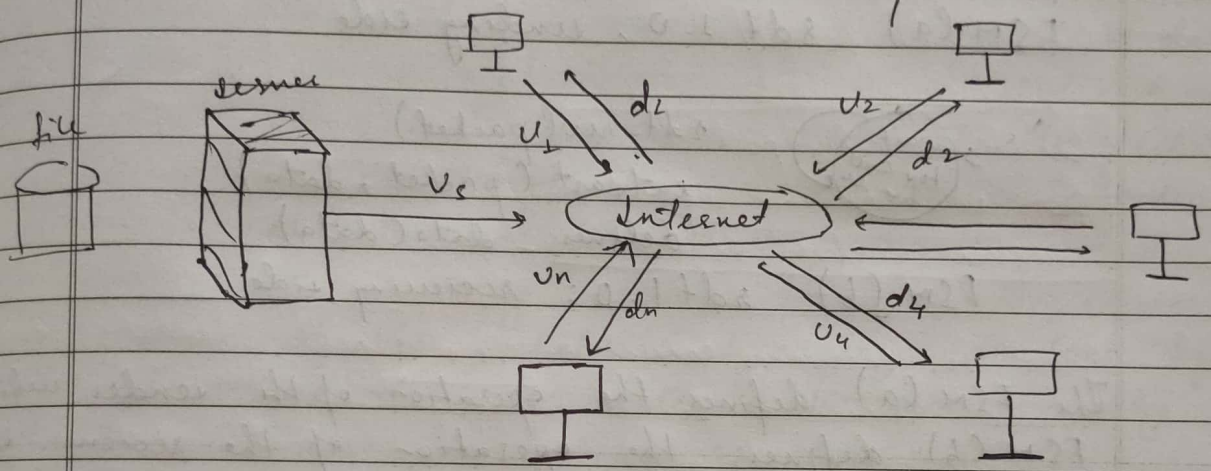
- 331 Username OK, password required.
- 425 cant open data connection
- 452 error writing file
- 404 file not found.

Q7) Explain P2P file distribution?

→ P2P file distribution is an application in which a large file is distributed from a single server to a large server number of hosts called

peers.

Each peer can redistribute any portion of the file it has recieved to any other peers, thereby assisting the server in the distribution process.



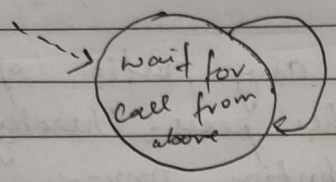
[FILE Distribution].

The server and peers are connected to the internet with access links. The upload rate of server is u_s and of Nth peers access link is u_N. The download rate of Nth peers access link is d_N. The size of file to be distributed is F bits and number of peers that want a copy is N

The distribution time is the time it takes to get a copy of file to all N peers.

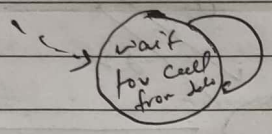
Q8) Explain reliable data transfer over a perfectly reliable rdt 1.0 in brief.

→ The finite state machine (FSM) definitions for rdt 1.0 sender and reciever are shown.



rdt_send(data)
packet = make_pkt(data)
udt_send(packet)

FSM (a) rdt 1.0 ; sending side



rdt_rcv(packet)
extract (packet, data)
deliver_data(data)

FSM (b) rdt 1.0 : recieving side

The FSM (a) defines the operation of the sender while FSM (b) defines the operation of the reciever. It is imparent to note that there are seperate FSMs for sender and recieners. The event calling the transitions is shown above the horizontal line labeling the transition, and the actions taken when the event occurs are shown below the horizontal line.

The sending file of rdt simply accepts data from the upper layer-side via the rdt_send (data) event, a packet containing the data.
On the recieving side the rdt recieves the data from the underlying channel via, the rdt_rcv (packet) even, removes the data via the action attracts and passes the data up to upper layer (via action deliver_data (data)).

Q9) Explain socket programming in TCP and UDP?

Socket programming with UDP.

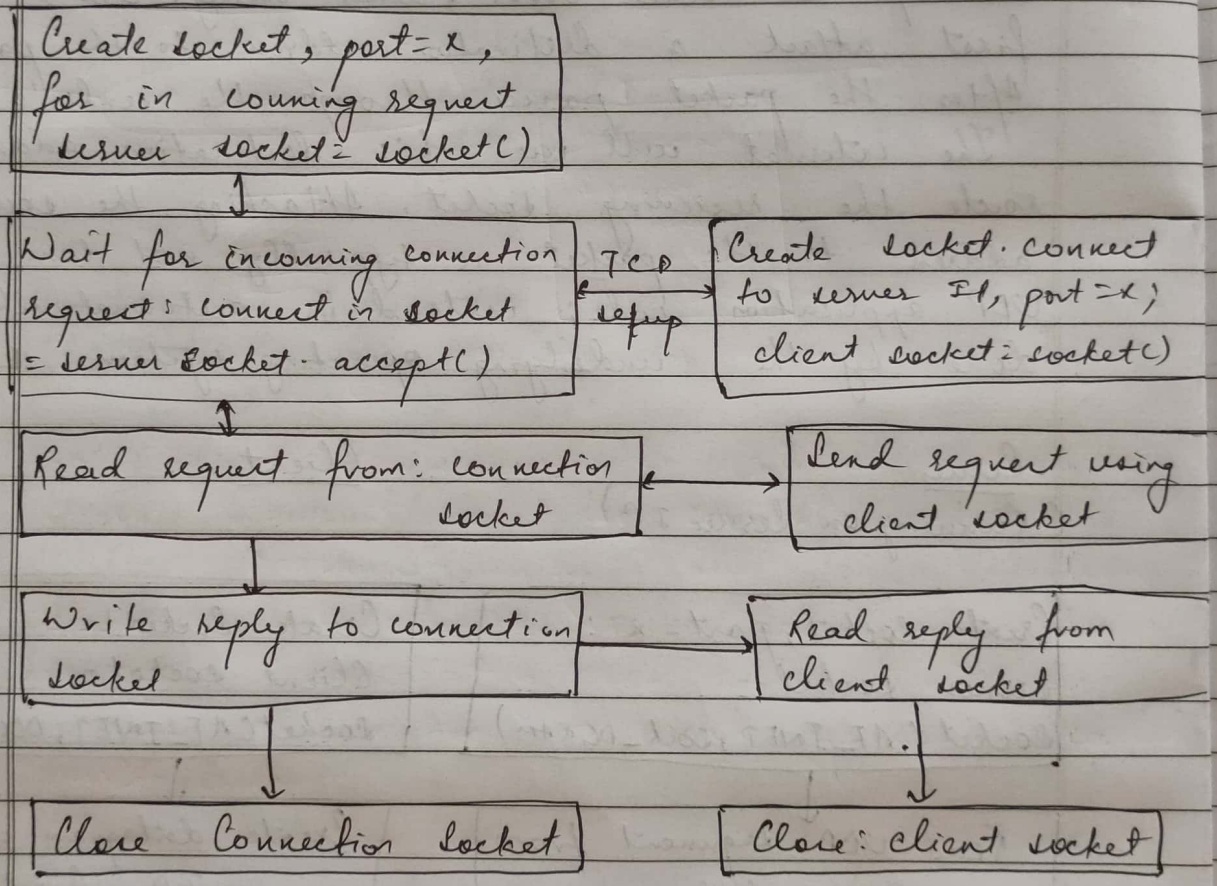
Before the sending process can push a packet of data out of the socket door, when using UDP, it must first attach a destination address to the packet. After the packet passes through the sender's socket. The internet will use this destination address to route the recieving socket. Attaching the source address to the packet is typically not done by the UDP application code; instead its automatically done by the underlying operating system.

Server (Running on server IP)	Client
Create socket, port = x : server socket = Socket (AF_INET, sock_DGRAM)	Create Socket! Client socket = socketCAF_INET, DOCK
Read UDP segment from server socket	Create datagram with server IP and port = x; Send diagram via client socket
Write reply to server socket specifying client address, port number	Read datagram from client socket
	Close Client - socket

Socket programming with TCP:

Server | Client

(Running on Server IP)



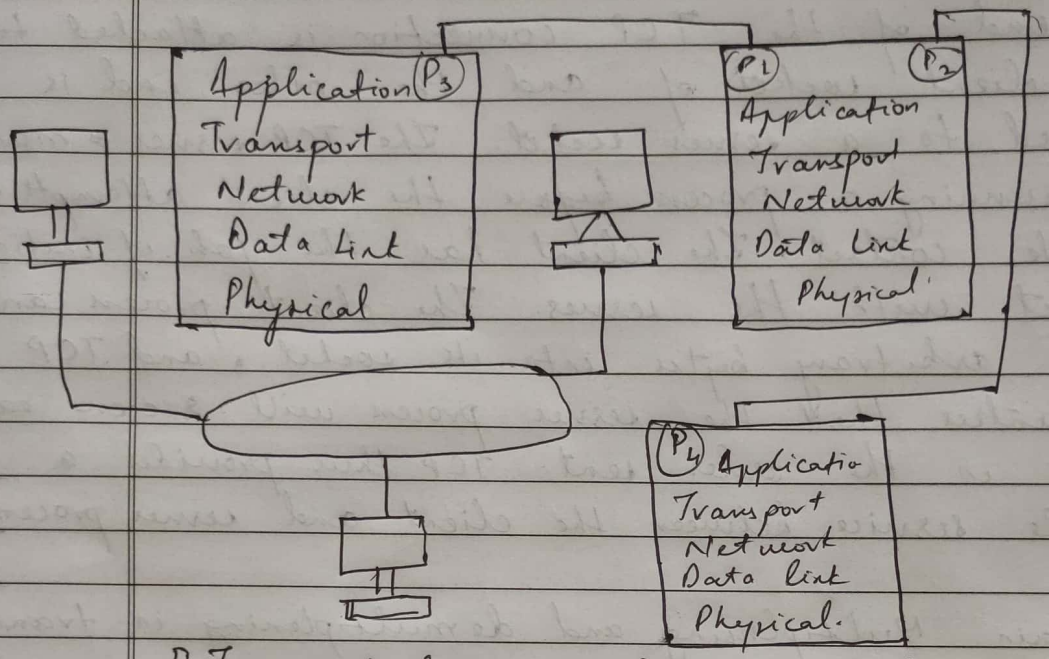
TCP is a connection oriented protocol. Before the client and server can start to send data to each other, they first need to handshake and establish a TCP connection.

One end of the TCP connection is attached to the client socket of and the other end is attached to a server socket. The TCP server must be running a process before the client attempts to initiate contact. The client has the job of initiating contact with the server. The client process can send arbitrary bytes into its socket, and TCP guarantees that the server process will receive each byte in the order sent. TCP thus provides a reliable service between the client and server processes.

Q10) Explain Multiplexing and demultiplexing in transport layer.

A process can leave one or more sockets, doors through which data passes from the network to the process and through which data passes from the process to the network.
The transport layer in the receiving host does not actually deliver data directly to a process, but instead to an intermediary socket.

At any given time there can be more than one socket in the receiving host, each socket has a unique identifier. The format of the identifier depends on whether the socket is a UDP or a TCP socket.



[Transport layer multiplexing & demultiplexing].

The job of delivering the data in a transport layer segment to the correct socket is called demultiplexing. The job of gathering data chunks at the source host from different sockets, encapsulating each data chunks with header information to create segments is called multiplexing.