Application layer

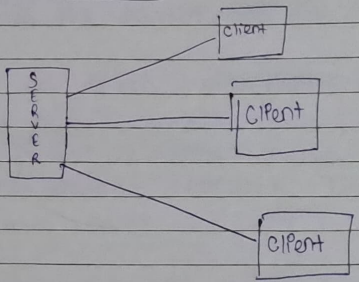
Principals of Network Application

Er web application there 2 programs

→ The browser program (for client)
→ The web server program (for server)

Network application Architecture

→ Client-server.

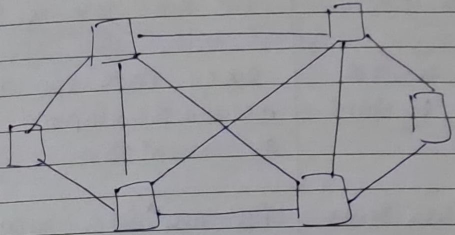


- The Server ~~requests~~ services request from clients

Ex- Web, FTP, e-mail

- In client-server application, a single-server host is incapable of keeping up with all the requests from clients so for this reason a data centre (housing large no. of hosts) is used to create a powerful virtual server.

→ Peer to Peer.



- There is no reliance on dedicated servers instead the application exploits direct communication between pairs of intermittently connected hosts, called peers.

Challenges:-

1) ISP friendly.
2) Security
3) Incentives.

Difference betn ClientServer vs Peer to Peer

ClientServer	Peer to Peer
1) In this network clients and servers are differentiated	In this network, clients and servers are not differentiated.
2) focus on info. sharing	focus on connectivity.

3) Centralized server is used to store data	In Peer to Peer Network, each peer has its own data.
4) Costlier than P2P	P2P are less costlier
5) responds to services requested by client	every node can do both request & respond for the services.
6) CS are more stable than P2P	P2P are less stable if no. of peer increase

Process Communication

- Programs running on different end system are actually processes.
- These processes on 2 different end systems communicate with eachother by exchanging messages across comp. network.

2 process:

-> client process
-> server process

The Interface between the Process & Comp. Network:

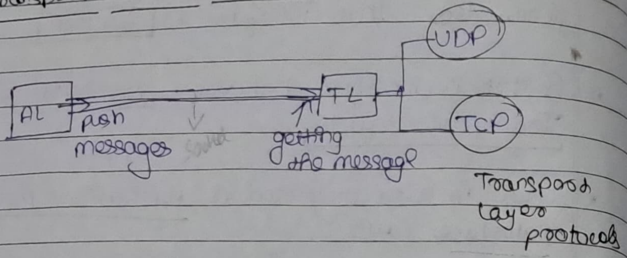
- A process sends messages into, and receives messages from, that Interface is $\boxed{\text{socket}}$.
- Socket is interface between Application layer and Transport layer.
- Socket is also referred as Application Programming Interface (API).
- Application developer has total control over application layer side of the socket and has little control of the transport layer side of the socket i.e:-
  1. choice of transport protocol
  2. Ability to fix few transport layer parameters such as maximum buffer & maximum segment sizes.

Addressing Processes:

- To identify the receiving process (by TL) we need
  -> ~~[illegible]~~ IP address of the host [Host is identified by this]
  -> Port number
- To know the address of the host to which message is destined, the sending process must identify the receiving process running the host.

→ Web server : 80
Mail-server : 25

Transport services available to Applications:



TCP services vs UDP services.

- TCP services.

→ Connection-oriented service
→ Handshaking procedure ~~alt~~ alerts both of them to be prepare for onslaught of packets.
→ After handshaking a TCP connection is said to be exist bet^{n} 2 process.
→ TCP connection delivers data without any errors.

NOTE: SSL: secure socket layer
Same as TCP but provides security services, like encryption, data integrity, and end point authentication.

UDP services.

- Lightweight transport protocol.
- connectionless so no handshaking
- unreliable data transfer service
  ↳ It does not guarantee that the message will ever reach to the receiving process.

Application layer Protocol.

It defines:
→ types of messages exchanged.
→ syntax of various messages
→ semantics of the field.
→ Rules for determing when & how process sends messages and responds to messages.

HTTP.

→ Hypertext Transfer Protocol.
→ Implemented in 2 programs: client program
server program
→ Web page contains objects which is addressable by URL.
→ URL — hostname of server
— object's pathname
→ Communication is done by exchanging HTTP messages.

Web browser : client side of HTTP
Web servers : server side

→ HTTP defines how Web client request Web pages from webservers and how web server transfer web pages to client.

→ HTTP uses TCP as its underlying protocol

→ HTTP is a stateless protocol i.e. it does not store info about clients.

Non persistent connection

→ The server closes the TCP connection after it sends the object.
→ It involves a 3 way handshake
→ TCP we need for establishment for each request.
→ After 3 way handshake, the server sends the requested HTML file.

Persistent connection

→ Subsequent requests & responses between the same client & server can be sent over the same connection. (single persistent TCP connection)

HTTP message format

▨ HTTP Request format

~~GET~~ The 1st line of an HTTP request message is called the request line.

Subsequent lines are called the header lines.

Request Line: Method field, URL field, HTTP version field.

- → GET (request an object)
- → POST (fills out a form)
- → HEAD (debugging)
- → PUT (web publishing tools)
- → DELETE (delete an object on a web server)

Format: Host name:
Connection:
User Agent:
Accept-language:

HTTP Response Message.

Status line
Connection:
Date
Server:
Last-Modified:
Content-Length:
Content-Type:

Response message: initial status line, six header line & then the entire body.

→ At connection "close" means that the client is going to close the TCP connection after sending the message.

Status code & phrases

→ At Date header line indicates the time & date when the HTTP response was created & sent by the server.

→ At Server indicates that it was generated by a web server.

→ Last-Modified. indicates when the object was created or last modified

→ Content-Length indicates the no. of bytes in the object being sent.

→ Content-Type indicates that the object in the entire body is HTML text.

Status code & phrases

200 OK: Request succeeded and the information is returned in the response.

301 Moved Permanently: Request object has been moved permanently.

400 Bad Request: Request could not be understood by the server.

404 Not found: The requested document does not exist on this server.

505 HTTP version Not Supported: The requested HTTP protocol version is not supported by the server

User-Server Interaction: Cookies

Cookie technology has 4 components:-

1) A cookie header line in the HTTP response message.
2) A cookie header line in the HTTP request message.
3) A cookie file kept on the user's end system & managed by the user's browser.
4) A back end database at the web site.

- User contacts website for 1st time, the server creates a unique identification number and creates entry in backend database i.e. indexed by identification number.

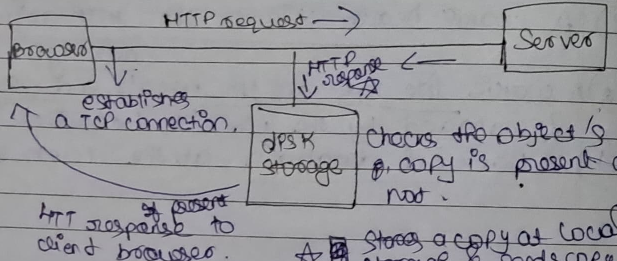
- Server responds to browsers, including in the HTTP response a Set cookie: header containing ident. no.

- Browsers appends a line to the special cookie file. The line includes the hostname of the server & the identification no.

Web Caching

A web Cache is also called proxy server which satisfies requests on the behalf of an origin web server.

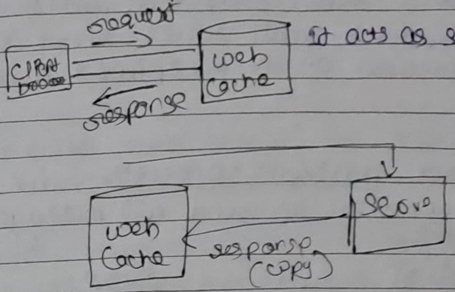
Web Cache has its own disk storage & keeps copies of recently requested objects.



Checks the object's copy is present or not.

★ Stores a copy at local storage & sends copy to client.

★ Cache is both server and a client.



It acts as server.

acts as client

Benefits

- Reduces response time
- Reduces traffic.

Electronic Mail in the Internet

3 major components.
- User agent
- mail servers
- SMTP (Simple Mail Transfer Protocol)

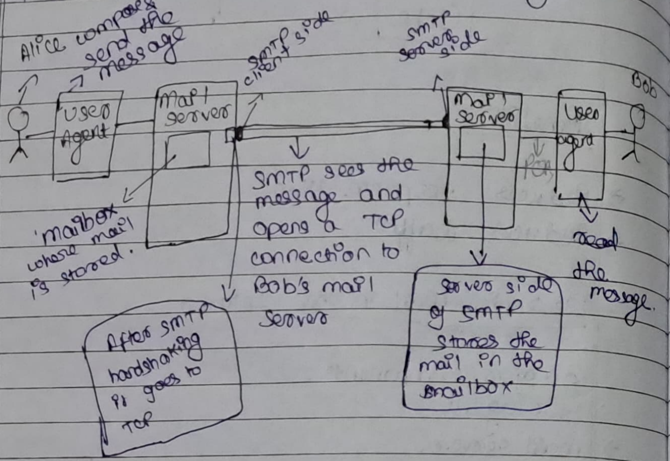
User agents it allows users to read, reply, save & compose messages

SMTP

- Application layer protocol
- Uses TCP to transfer mail from sender's mail server to the recipient's mail server.

SMTP has 2 sides: client side, which executes on the sender's mail server, server side which executes on the recipient's mail server.

→ SMTP uses persistent connections i.e. in one TCP connection it send all messages


Alice composes send the message
SMTP client side
SMTP server side
Bob
User Agent
Mail server
Mail server
User agent
POP
'mailbox' where mail is stored.
SMTP sees the message and opens a TCP connection to Bob's mail server
After SMTP handshaking it goes to TCP
server side of SMTP stores the mail in the mailbox
read the message.

SMTP	HTTP
1) Transfer files from one mail server to other.	Transfer files from web server to a web client
2) It uses only persistent connection	⑥ Persistent HTTP uses only persistent connection
3) Push protocol - the sending mail server pushes the mail to the receiving mail server.	Pull protocol - someone loads info. on web server and HTTP is used to pull the info.
4) It requires each message to be a 7 bit ASCII format.	Does not impose restriction
5) ~~encapsu~~ places all of the message's objects into one message	encapsulates each object in its own HTTP response message.

Popular mail access protocols: POP3
IMAP
HTTP

* POP3 is used to transfer mail from recipient's mail server to recipient user agent.

POP3

It begins when user agent opens TCP connection to mail server on port no. 110.

3 phases:

Authorization
Transaction
update

Authorization: User agent sends a username and a password to authenticate the user.

Transaction: User agent retrieves the messages → the user agent issues commands, and the server responds to each command with a reply.

Update: The mail server deletes the messages that were mask for deletion.

2 possible responses.

1) OK: by server to indicate previous command was fine

2) ERR: by server to indicate that something was wrong with previous command.

IPV4 Header.

Length of IPV4 header [20-60 Bytes]

Min = 20 Bytes [Each row = 4 bytes ($4 \times 5 = 20$)]

Scaling factor = 4

Types of service is used for QOS (Quality of Service)

Total Length = Header Length + Payload Header.

DF = Do not fragment Bit
0 → permission does give to device to fragment the datagram.
1 → It does not give. " "

MF = More fragment Bit

Time to Live.
Max. no. of hopes a datagram can take to reach the destination

Protocol:-
It tells the network layer that which protocol the IP datagram belongs to.

Options:-
Record Route
Source Routing
Padding.

DNS.

-> Distributed database implemented in data servers

-> Application layer protocol that connects ~~the dat~~ host with the database to know IP address of the hostname.

-> Port NO: 53.

-> UDP

-> Translating username ~~a~~ to IP address

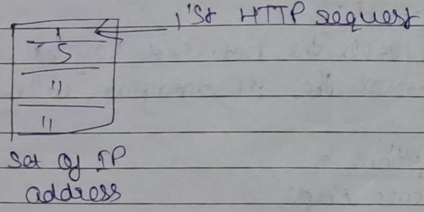
Host-Aliasing.

-> Canonical

-> mnemonic

Mail server aliasing

Load distribution



If one DNS server:-

-> A DNS server would have crashed entire internet

-> Traffic increase

-> Distant centralized database

-> maintainance

Classes of DNS server:-

-> Root DNS

-> TLD (Top level DNS)

-> Authoritative DNS

in
↑
edu
↑
xim

DNS caching:-

It is a temporary DNS ~~server~~ storage on a device that contains DNS records of already visited domain names