CSE 421 (Mid)

Network models & Protocol Architecture

Lecture: 01

* benefits of using layered model
  - ↳ fosters competition
  - ↳ tech change in one layer doesn't effect other
  - ↳ each layer has defined functions to act.
* Protocols
  - ↳ all communications are governed by protocols
  - ↳ Protocols are rules that communications will follow
  - ↳ rules vary depending on the protocol.
* Protocols → requirement
  - ↳ identified sender & receiver
  - ↳ common language & grammar
  - ↳ speed and time of delivery
  - ↳ confirmation or acknowledgement requirements.

* Common computer protocols must agree in →
  ↳ message encoding
  ↳ message formatting and encapsulation
  ↳ message size, timing, delivery option.

* Standards
  ↳ endorsed by the networking industry and approved by a standards organization.

* benefits
  ↳ create and maintain an open market and competitive market.
  ↳ ensured greater compatibility & interoperability.

* Categories.
  ↳ De facto → TCP/IP Protocol Model
  ↳ De jure → OSI Reference Model.

* Open Standards
  ↳ encourages →
    ↳ interoperability
    ↳ Competition
    ↳ Innovation.

* Standard organizations are
  ↳ vendor - neutral
  ↳ non-profit org
  ↳ established to develop and promote the concept of open standards.

* Internet Standards

↳ Internet Society (ISOC) → Promotes open development and evolution of internet

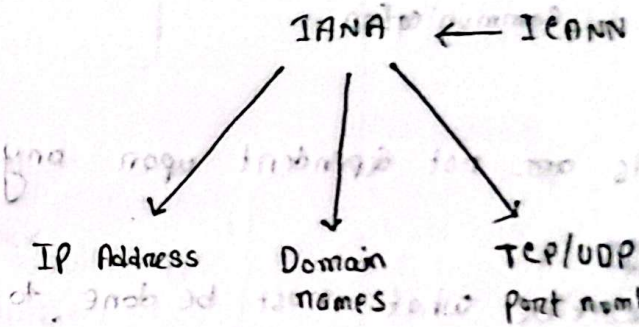
↳ Internet Architecture Board (IAB) → Responsible for management and development of internet standards.

↳ Internet Engineering Task Force (IETF) → Develops, updates and maintains internet and TCP/IP technologies.

↳ Internet Research Task Force (IRTF) → Focused on long term research related to internet and TCP/IP protocols.

↳ Internet Corporation for Assigned Names and Numbers (ICANN) → Coordinates IP address allocation, management of domain names, assignment of other info

↳ Internet Assigned Numbers Authority (IANA) → Oversees and manages IP address allocation, domain name management, and protocol identifiers for ICANN.



* Electronic and communications Standards

↳ IEEE → creating standards in power, energy, healthcare, telecommunication and networking

↳ EIA → Standards for electrical wiring, connectors, and 19 inch racks used to mount networking equipments.

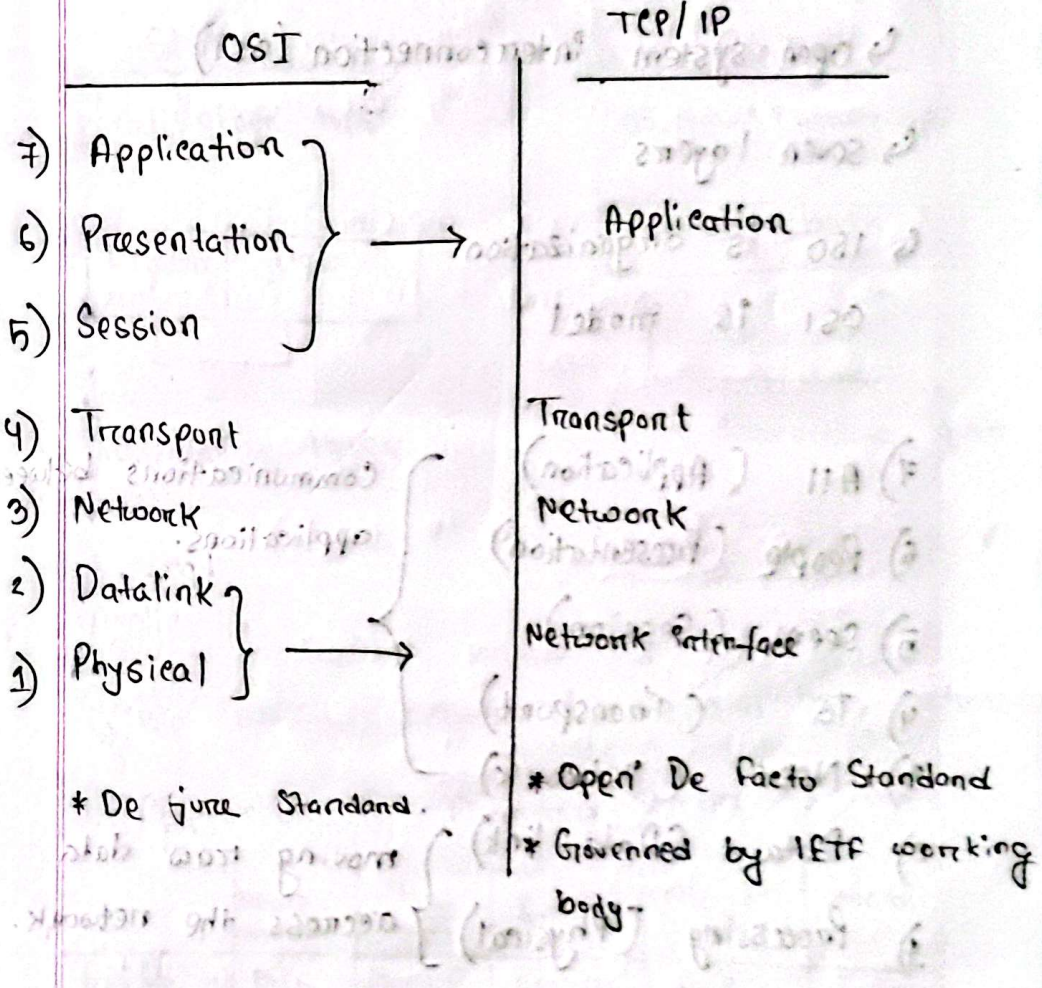
↳ TIA → Standards in radio equipments, cellular towers, VoIP devices, satellite communications.

↳ ITU-T → Standards for video compression, internet protocols Television, broadband communication.

* Protocols are not dependent upon any specific technology.
They describe what must be done to communicate but not how it is to be carried out.

Example: HTTP (it works across different hardware, operating systems, network technologies).

* Protocol Suites



* OSI Model

↳ Open system interconnection (OSI)

↳ seven layers

↳ ISO is organization
OSI is model

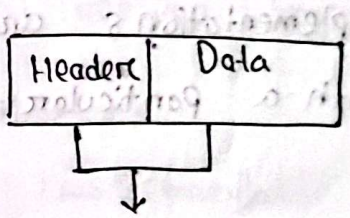
7) All (Application)
6) People (Presentation)
5) Seem (Session)
4) To (Transport)
3) Need (Network)
2) Data (Data-link)
1) Processing (Physical)

Communications between applications.

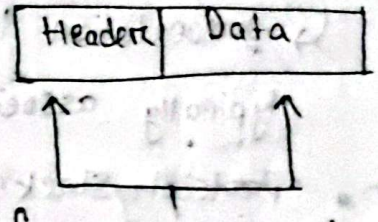
moving raw data across the network.

* Application layer (7th layer)

Sender from email/other app : **Receiver** to email/other app



to presentation layer



from presentation layer

* Applications

↳ the interface between human and Data network

↳ Responsible for providing service to user.

	Name system	Host config	Email	File Transfer	web
Application layer	DNS	BOOTP DHCP	SMTP POP IMAP	FTP TFTP	HTTP HTTPS

* Presentation layer (6th Layer)

↳ responsible for translation, compression and encryption ; ie. 3 primary functions.

↳ presentation layer implementations are not typically associated with a particular protocol stack.

* Session layer (5 th Layer)

↳ responsible for dialog control & synchronization

↳ it handles exchange of information

↳ to initiate dialogs

↳ keep them active

↳ to restart session that are disrupted or idle for a long period of time.

* Transport Layer (4th Layer)

↳ it is responsible for the delivery of message from one process (sender) to another (receiver).

↳ Transport Layer PDU is called Segments.

functions.

→ Segmentation and Reassembly

→ Adds port address & sequence number

→ Connection control

→ Flow and error control.

→ Multiplexing

* Functions - Segmentation / Reassembly

→ Segments data received from application layer into small parts.

→ Steps (sender)

↳ Segments into small parts

↳ Add a number to identify the application

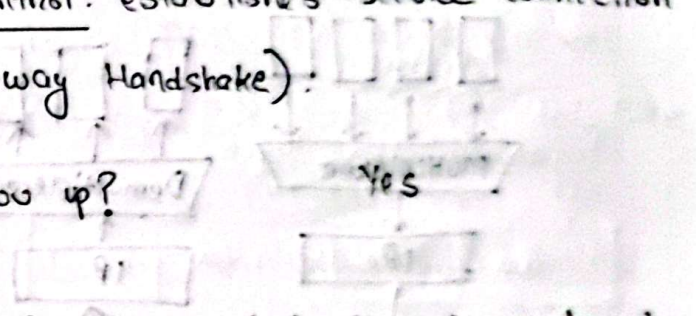
↳ Add a number sequence the segmented part.

→ (Receiver)

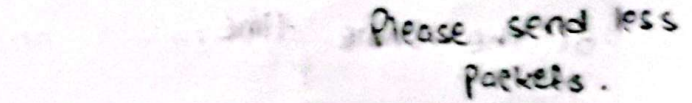
↳ uses the sequence number to order them sequentially, merge them and send them to upper layer.

* Functions

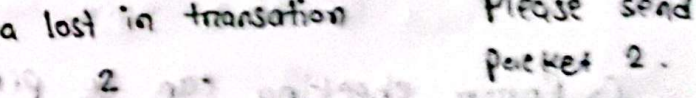
Connection control: establishes secure connection (TCP - Three way Handshake):



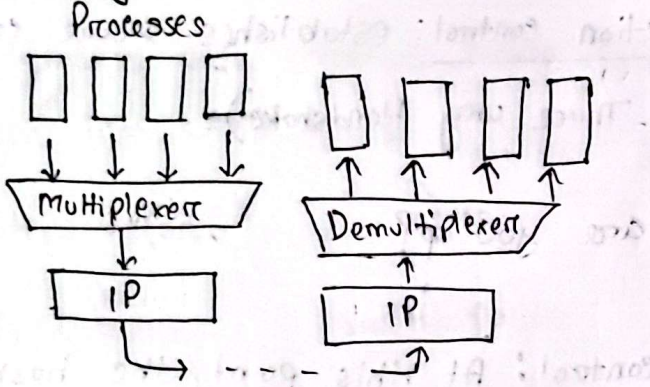
flow control: At this point, the host has too many packets to process, hence the buffer to store incoming packets overflows.



Error Control:



Multiplexing:



↳ segmentation allows session multiplexing - multiple applications can use the network at the same time.

↳ Data segmentation facilitates data carriage by the lower network layers.

↳ Error checking can be performed on the Data in the segment to check if the segment was changed during transmission.

* Network Layer (3rd Layer)

→ Network Layer PDU is called Packet.

→ The network layer is responsible for the delivery of individual packets from the source host to the destination host.

→ Common Network Layer Protocol is called Internet Protocol (IP)

Functions

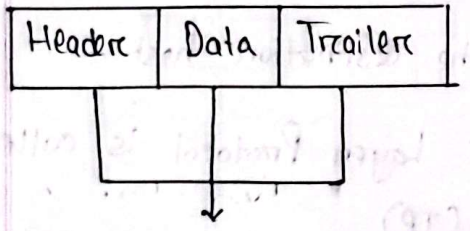
↳ Adds an address (logical address) to identify sender and receiver hosts.

↳ Decides which path to take (Routing).

* Data Link Layer (2nd Layer).

Sender

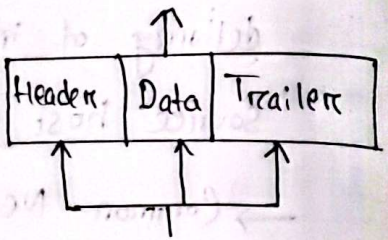
From network layer



To Physical Layer

Receiver

To Network Layer



From Physical Layer

→ Data link layer PDU is called frame

→ The Data link layer is responsible for moving frames from one hop (node) to next.

→ Protocol on this layer varies.

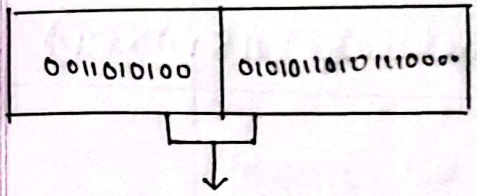
Functions

→ Framing
→ Physical Addressing
→ Flow Control
→ Error Control
→ Access Control

* Physical Layer (1st layer)

Sender

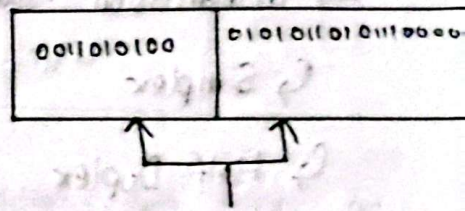
From Data link layer



To transport medium

Receiver

To Data Link Layer



From Transport medium

→ physical layer is responsible for movements of individual bits from one hop (node) to next.

Functions

→ Physical characteristics of interface & medium
→ Representation of bits
→ Data Rate
→ Synchronization of bits.

→ Physical Topology

↳ Bus, ring, mesh.

→ Transmission Modes

↳ Simplex

↳ Half Duplex

↳ full Duplex.

Layers	Transport	Network	Datalink
Address	Port number	IP Address	MAC Address
Represent	16 bit	32 bit	48 bit
Information	identify different app/ processes running in computer example: 80-web 23 - TCP	universal address, each host uniquely defined. example: 192.168.10.1	known as media access control address every interface, port/device has an unique mac number. example: 07:01:02:01:2C

TCP/IP Model

→ Developed by the US Defense Advanced Research Project Agency (DARPA) for its Packet switched network. (ARPANET).

→ Used by the global internet.

TCP/IP (PDU)

Application	⟶	Data
Transport	⟶	Segment
Network	⟶	Packet
Data-link	⟶	Frame
Physical	⟶	Bits

First: MAC, IP, PORT

Second: Destination then Source.

Physical address ⟶ change hop to hop.

logical & port ⟶ source to destination

Web Cache or Proxy Server.

Lecture :02

* Web Cache (proxy server)

↳ Satisfy client request without involving origin server.

→ User sets browser : Web accesses via cache.

→ Browser sends all HTTP requests to cache.

- Object in cache : cache returns object
- else cache requests object from origin server, then returns object to client.

→ A proxy server acts as both client & server.

- Server for original requesting client
- client to origin server.

→ Typically Proxy Servers are installed by ISPs

- university
- company
- Residential ISP.

Application Layer (HTTP) (Part:01)

TCP/IP OSI Lecture:02

Application
- Application
- Presentation
- session

→ DHCP, DNS, FTP, HTTP, HTTPS, POP, SMTP, SSH etc. (Data)

Transport
- Transport → TCP / UDP (segments)
  (port numbers)

Network / Internet
- Network → IP address : IPv4, IPv6 (Datagram).

Network Access
- Data link → MAC Address (frame).
- Physical → ethernet cable, fibre, wireless, coaxial etc

→ Application Layer Protocols

↳ Provide the rules and formats that govern how data is treated in the application layer.

→ Application Software

↳ The programs used to communicate over the network.

example:

Protocol → HTTP

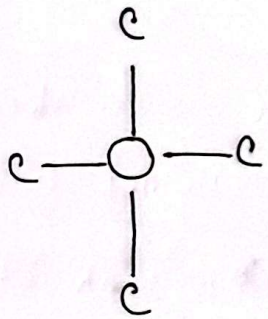
application → web browsers.

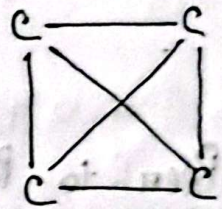
→ when accessing information on a device, the data may not be physically stored on that device.

→ if that is the case, a request must be made to the device where the data resides.

Two methods

- client / Server
- Peer - to - Peer





Client Server Model

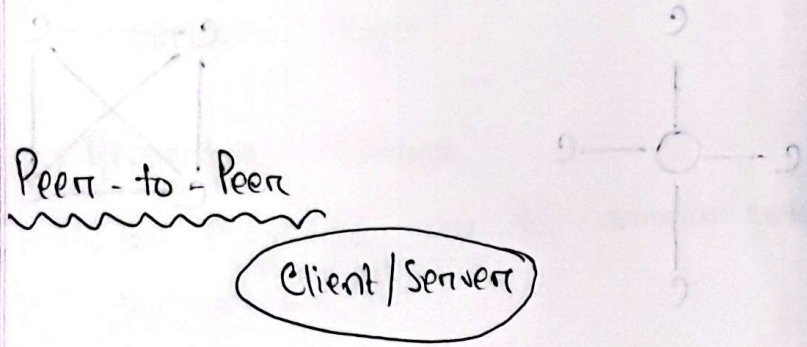
Client → hardware, software combination.

Server → resources are Stored.

Client begins the exchange by making a request for data.

The server responds with one or more streams of data.

Server may also require control information. User authentication or identify a file to be transferred.



↳ Two or more computers are connected via a network and can share resources (such as printers and files) without a dedicated server.

↳ End devices (peers) can function as either a server or client depending upon the required service.

Web and HTTP (Part: 02)

WWW - The Web.

↳ A web page or webpage is a document that is viewed in an internet browser.

↳ Contains text, graphics, hyperlink etc.

↳ webpage contains objects.

↳ Contains base HTML-file which includes several referenced objects.

↳ Objects can be HTML file, JPEG image, Java applet, audo file etc.

↳ A web page can be accessed by entering a URL address into a web browser's address bar.

http://www.nytimes.com/tech/index.html

- http: application transfer protocol
- www: host name
- nytimes: domain name
- com: top level domain
- tech/: path
- index.html: file

case sensitive.

* HTTP messages

2 types

→ HTTP Request message

→ HTTP Response message.

HTTP method types

HTTP/1.0

- GET
  - ↳ Primarily gets information only
  - ↳ Retrieve data
- POST
  - ↳ Creating new data
- HEAD
  - ↳ also retrieve but asks server to leave requested object out of response.

HTTP/1.1:

- GET, POST, HEAD
- PUT / PATCH
  - ↳ update data
  - ↳ replace existing object
- DELETE
  - ↳ Deletes data.

Uploading from input

POST method:

- web page often includes form input
- input is uploaded to server in entity body.

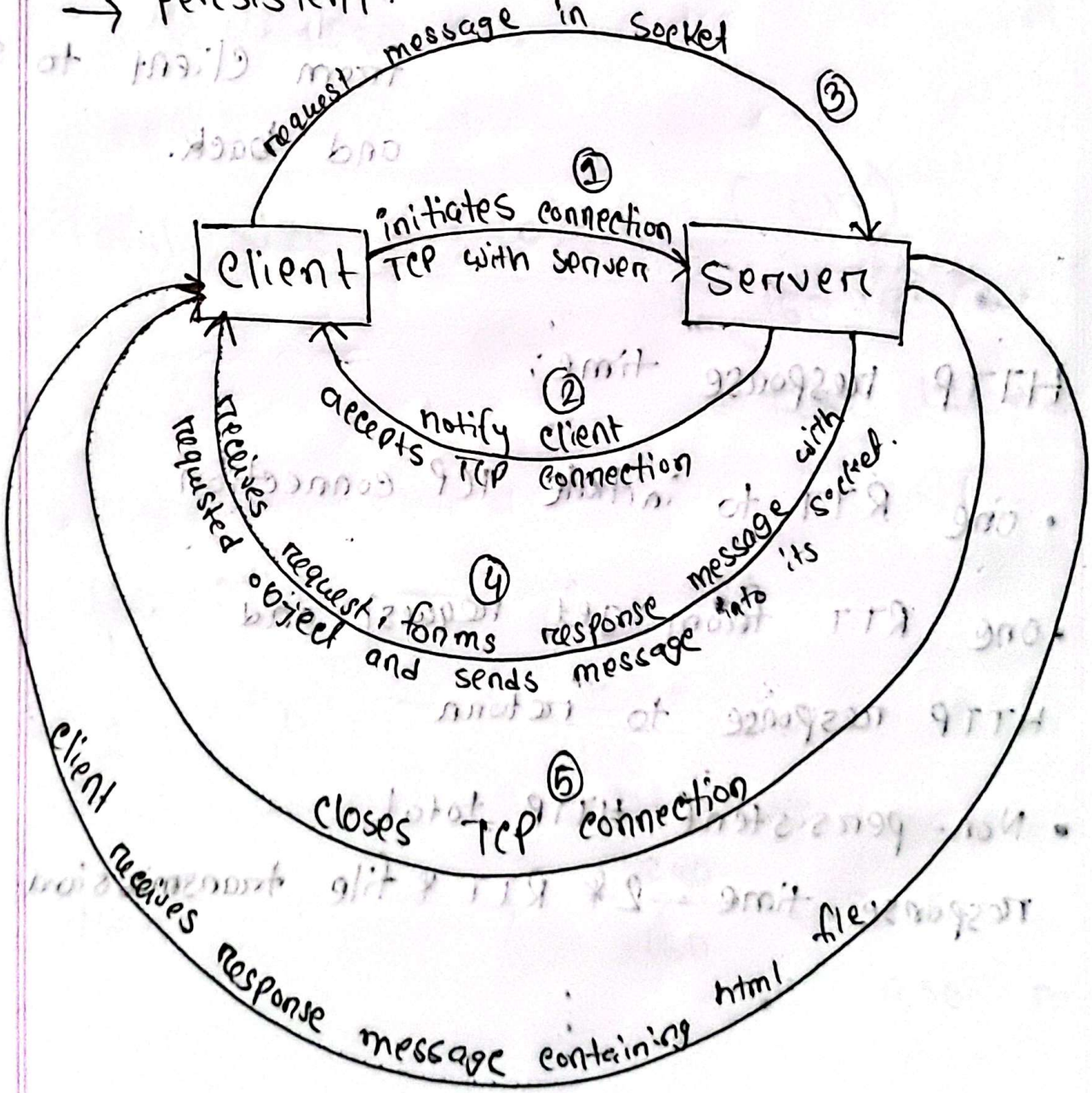
URL method:

- Uses GET method
- Input is uploaded in URL field of request

HTTP connections (Part : 03)

→ Non persistent

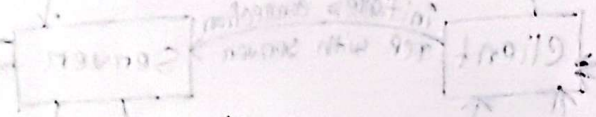
→ persistent?



non-persistent.

Non Persistent HTTP : response time

RTT (Round Trip Time) : time for a small packet to travel from client to server and back.



HTTP response time :

- one RTT to initiate TCP connection
- one RTT from get request and HTTP response to return
- Non-persistent HTTP total response time = 2 * RTT * file transmission time.

Quiz question

i) RTT = 2 x RTT x no. of object. { .04x2 = .08 = 80 msec

= 2 x 80 x 8

= 1280 ms.

ii) Total size of object = (20x8)

= 160 x8

= 1280 Mbps.

160 Mbps ⟶ 1 sec

1 " ⟶ $\frac{1}{160}$ "

1280 " ⟶ $\frac{1280}{160}$ = 8 sec

= 8000 ms.

∴ file transmission time = 8000 ms.

iii). Total response time = $T_{RTT} + f_T$

$= 1280 + 8000$

$= 9280$ ms.

* Persistent HTTP

↳ Server leaves connection open after sending response

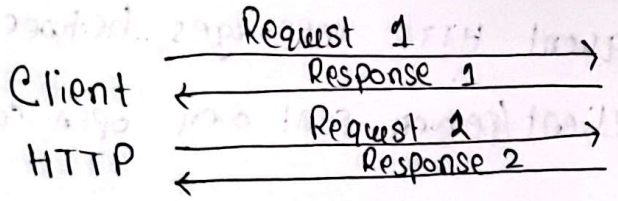
↳ Subsequent HTTP messages between same client/server sent over open connection.

↳ Client sends requests as soon as it encounters a referenced object.

↳ As little as one RTT for each of the referenced objects.

Cookies (Part: 04)

Stateless HTTP



Don't remember previous request response chain.

HTTP is 'Stateless'

↳ Server maintains no information about past client requests.

User - server state : cookies.

many websites use cookies: Example:

1) Cookie header line of HTTP response message
- Susan always access from PC

2) Cookie header line in next HTTP request message.
- visits specific e-commerce site for first time

3) Cookie file kept on user's host, managed by user's browser.
- when initial HTTP requests arrives at site, site creates unique id, entry in backend DB for ID.

4) Backend database at website.

What cookies can be used for?

→ Authorization

→ Shopping carts

→ Recommendations

→ User session state (web email)

How to keep 'state'

- protocol endpoints: maintain 'state' at sender/receiver over multiple transactions.

- cookies: http messages carry state.

Cookie & Privacy

- Cookies permit sites to learn a lot about us.

* Web Cache or Proxy Server (Part: 05)

→ Satisfy client request without involving origin server.

→ User sets browser: Web accesses via cache

→ Browser sets all HTTP request to cache

- object in cache: cache returns object
- else cache requests object from origin server, then returns object to client.

→ A proxy server acts as both client & server.

- Server for original requesting client
- client to origin server.

→ Typically Proxy servers are installed by ISPs. • University • company • residential ISP.

Advantage of Web Caching

→ reduce response time for client request.

→ saves bandwidth (prevents downloading of same content multiple times)

→ helps log usage, block unwanted traffic

→ Internet dense with caches:

↳ enables 'poor' content providers to effectively deliver content.

example:

i) avg obj size 1 Mbits

ii) Avg req rate from browsers to origin server 15/sec

iii) bandwidth 100 Mbps

iv) RTT from router to any origin server : 2 sec

v) Access link rate : 15 Mbps.

What is average response time?

✓ Avg response time = LAN Delay + Access Delay + Internet Delay.

✓ Traffic intensity on LAN = $\frac{(\text{Avg req/sec} * \text{Avg Obj Size})}{\text{Transmission link bandwidth}}$

$= \frac{15 \times 1000000}{100000000}$

$= .15$

Traffic intensity on the Access link = $\frac{15 \times 1000000}{15 \times 1000000}$

$= 1.$

Intensity Delay = 2 sec (RTT)

Consequence

- LAN utilization : 15%.
- Access link utilization: 100%.

Solution of reducing delay:

i) increase the bandwidth of access link

ii) install a web proxy server on web cache at the network.

When access link speed increases. Cost ↑

Stale Cache

→ one problem of using Proxy server

↳ the object housed in web server may have been modified since the copy was cached at the client

→ HTTP has a mechanism that allows a cache to verify that its object are up to date. This mechanism is called 'conditional GET'.

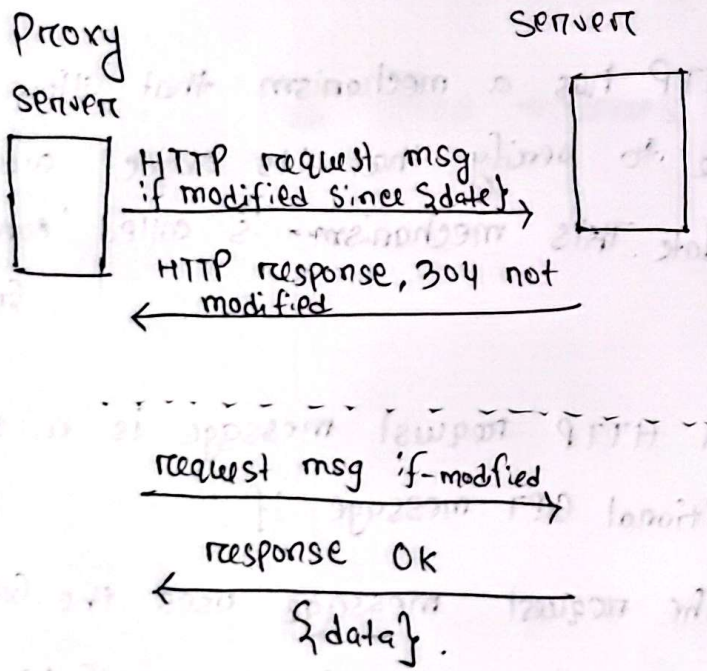
→ An HTTP request message is a so-called conditional GET message if

i) the request message uses the GET method

ii) The request message includes an If-Modified-Since header line.

Conditional GET

Goal: Don't send object if cache has up to date cached version.



HTTPS (Part: 06).

→ more secure version of HTTP

→ allows ~~entering~~ transferring the data in an encripted form.

→ use an enenyption protocol known as Transport Layer Security and oficially it is referred to as 'Secure Sockets Layer (SSL)'.

→ ~~transport~~ transmit data over 443 port num.

Criteria and advantage

→ websites to have HTTPS, needs signed SSL Certificate.

→ SSL enenypts data, client ⟶ Server.

→ is a transport layer protocol.

→ SEO Advantage: Gives preference to whose website that used HTTPS.

Application Layer (E-mail & DNS)

Lecture :03

Electronic Mail

* 3 major component
  - ↳ user agent
  - ↳ mail servers
  - ↳ simple mail transfer protocol (SMTP)

User Agent

↳ software agent that is used for composing, editing, reading, forwarding mail messages.

example: Outlook, iphone mail client, web browser.

Mail Servers

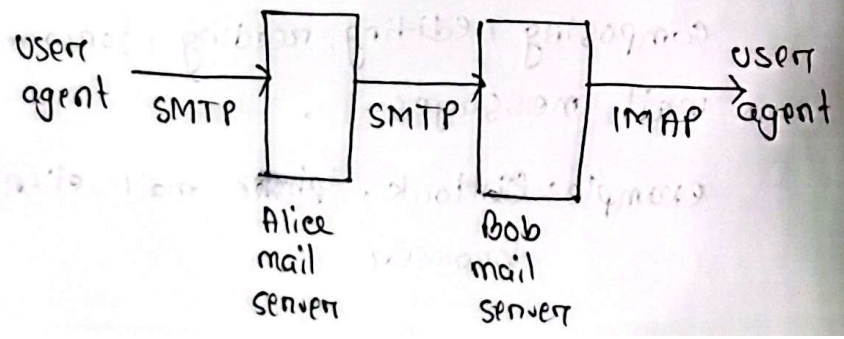
mailbox → contains incoming messages for user

message queue → of outgoing (to be sent) mail messages.

* SMTP protocol between mail servers to send email messages:

client : sending mail server

server : receiving mail server.



* ## Email : SMTP

→ uses TCP to reliably transfer email messages from client to server, port 25.

→ direct transfer : sending server (acting like client) to receiving server.

→ Three phases of transfer

- SMTP handshaking (greeting)
- SMTP transfer of messages
- SMTP closure

SMTP : final words

→ SMTP uses persistent connection.

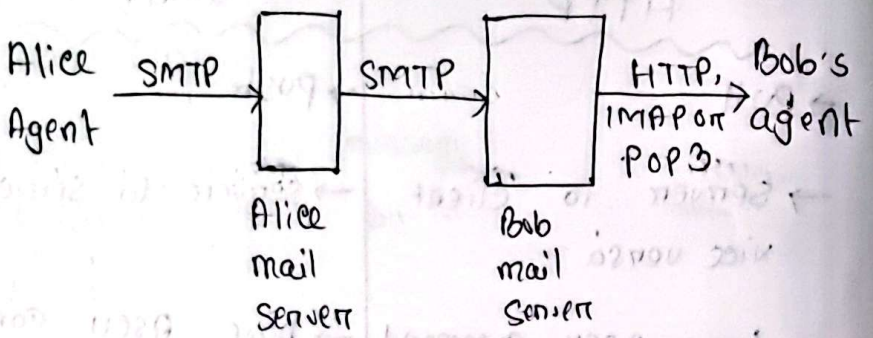
→ SMTP requires message (header & body) to be in 7 bit ASCII

→ SMTP server uses CRLF.CRLF (\r\n.\r\n) to determine end of message.

SMTP with HTTP

HTTP	SMTP
→ pull	→ push
→ server to client, vice versa	→ server to server
→ have ASCII command	→ have ASCII command
→ each object encapsulated in its own response message	→ multiple objects sent in multipart message

* Mail Access Protocol



→ SMTP : delivery/storage to receiver's server.

→ Mail access protocol: retrieval from server

- POP → authorization download (Port : 110)
- IMAP → more feature including manipulation of stored messages on server (Port: 143)
- HTTPS → Web based (Gmail, Hotmail, Yahoo! Mail etc).

* Domain Name System (DNS)

→ DNS is an automated client/server service

→ internet programs requiring domain name look up send a resolution request to the DNS resolver in the local operating system.

→ the resolver in turn handles the communications required.

Centralized DNS? No

Reasons

↳ Single point of failure

↳ traffic volume

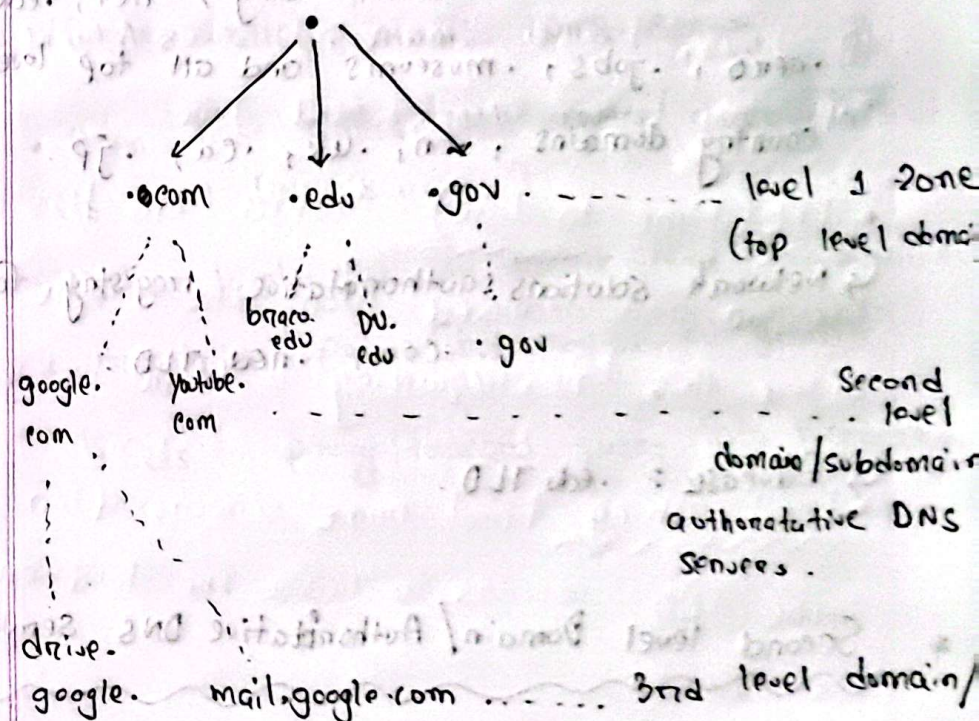
↳ Distance centralized database

↳ Maintenance

↳ Doesn't scale

↳ Solution: Distributed Database

DNS: a distributed, hierarchical database.



* Top-Level Domain, and authoritative servers.

↳ responsible for .com, .org, .net, .edu, .aero, .jobs, .museums and all top level country domains, .en, .uk, .ca, .jp.

↳ Network solutions: authoritative registry for .com, .net TLD.

↳ Educase: .edu TLD.

* Second level Domain/ Authoritative DNS servers:

↳ Organization's own DNS server(s), providing authoritative hostname to IP mapping for organizations named hosts.

↳ can be maintained by organization or service provider.

* Local DNS name Server

- each ISP has one.
  ↳ also called 'default name server'
- When host makes DNS query, query is sent to its local DNS server.
  ↳ has local cache of recent name-to-address translation pairs (but may be out of date)
  ↳ Acts as proxy, forward query into hierarchy.

* DNS : Caching, updating records

- Caching → When a name server learns an IP address for a domain, it saves (caches) this info for faster access next time.
- TTL (Time to Live) : Cached info has a lifespan and removed from the cache after this time, requiring a fresh lookup.
- Local caching: Local name servers often cache TLD records to reduce the load on root servers.
- Out-of-Date cache: May be out of Date if a domain's IP changes. The change only becomes known accross the internet once all caches with the old IP expire.
- Updating Standard : Has a standard IETF for updating DNS record to address this issue, allowing quicker updates across servers.

DNS Record

DNS : Distributed database Storing resource record (RR)

RR format: (name, value, type, ttl)

web server → A

wrong name → CNAME

Authoritative → NX

mail server → MX.

if name is same, then using these 4 type we can make difference and get the IP address.

Introduction to Transport Layer

Lecture : 04

Transport layer protocol

→ UDP : User Datagram Protocol

→ TCP : Transmission Control Protocol.

Transport Layer : Logical communication between processes.

Network Layer : Logical communication between hosts.

Segments : Transport layer PDU.

* Function of Transport Layer:

UDP + TCP:
- → segmenting the data and managing each piece.
- → reassembling the ~~data~~ segments into streams of application data.
- → identifying the different applications.
- → Multiplexing.

TCP only (Reliability):
- → initiating and terminating a session
- → performing flow control between end users
- → enabling error recovery.

* Some applications need their data to complete with no errors or gap.

But can accept a slight delay to ensure this.

(Reliable, TCP)

email

* Some applications can accept occasional errors or gaps in the data.

But they cannot accept any delay.

(Fast, UDP)

online games.

* UDP : User Datagram Protocol

↳ best effort service

↳ used by applications that requires no delay in data delivery.

* How does UDP deliver fast?

↳ no connection establishment (which can add delay)

↳ small header size

↳ no error or flow or congestion control

* UDP used by

↳ streaming multimedia apps

↳ SNMP.

* reliable transfer over UDP:

↳ add reliability at application layer

↳ application specific error recovery

Function: 01

* The transport layer divides the data into pieces and adds a header for delivery over the network. (Segmentation & Reassembly).

Application layer Data

Piece 1 Piece 2 Piece 3

UDP Datagram

Header	Piece 1
Header	Piece 2
Header	Piece 3

TCP Segment

Header	Piece 1
Header	Piece 2
Header	Piece 3

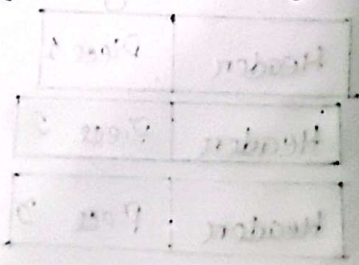
Function :02

Different diagram and segments may take different routes at the time of sending to destination.

When destination receives the datagram or the segments it get (reassembly).

Datagram doesn't rearrange and lost datagrams are not re-sent.

Segments gets rearranged and segments are not lost.



TCP and UDP Headers

* TCP header has - (20 - 60 byte).
  - → Source port address (16 bit)
  - → Destination port address (16 bit)
  - → sequence number (32 bit)
  - → Acknowledgement number (32 bit)
  - → HLEN (4 bit)
  - → Reserved bit (6 bit)
  - → Flag (6)
  - → Window size (16 bits)
  - → Checksum (16 bits)
  - → Urgent pointer (16 bits)
  - → options and padding (16 bits).
* UDP header has - (8 bytes).
  - → Source port number (16 bits).
  - → Destination port number (16 bits)
  - → Total length (16 bits)
  - → Checksum (16 bits)

function: 03

Port numbers / addresses are used to identify different applications / processes running in a computer.

16 bits in length.

→ represented as a single bit decimal

→ range 0 - 65535

→ web - 80

→ SMTP - 25

→ Spotify - 4070

Port Numbers.

well known → 0 to 1023

registered → 1024 to 49151

private or dynamic → 49152 to 65535.

25 → SMTP

110 → POP 3

80 → HTTP

53 → DNS

143 → IMAP

8008 - alternate HTTP

4070 - Spotify

3306 - MySQL

Client can use any random (dynamic) port numbers, but servers can't.

Servers must use well known port numbers.

If two sessions in the same server then, two sessions will have different port numbers but server will have the same.

(Identifying different applications).

function :04

multiplexing, demultiplexing : based on segment, datagram header field values :

UDP : demultiplexing using port number

TCP : demultiplexing using 4 tuple: ~~source~~

- ↳ source IP
- ↳ destination IP
- ↳ source port number
- ↳ destination port number.

Transport Layer (TCP)

Lecture : 05

* Byte number

→ The bytes of data being transferred in each connections are numbered by TCP.

→ Number starts with an generated number in range of 0 and $2^{32}-1$

example:

1st byte number — 1067

last byte number — 4066

* Sequence Number

→ The sequence number of the first segment is the ISN (initial sequence number) which is a random number (byte number).

→ Suppose, a TCP connection is transferring a file of 5000 bytes. The first byte is 10001. Data sent in 5 segment carrying 1000 bytes.

Solution :

Segment 1 : 10001 - 11000

Segment 2 : 11001 - 12000

Segment 3 : 12001 - 13000

Segment 4 : 13001 - 14000

Segment 5 : 14001 - 15000

* Acknowledgement Number

→ if receiving host TCP receives uncorrupted data, then

→ it is acknowledged using acknowledgement number

→ the value of acknowledgement field in a segment defines the number of the next byte the receiver expects to receive.

acknowledgement number 1001 means:

1000 data got. Ready to get data from 1001.

→ the acknowledgement number is cumulative.

→ receiver acknowledges multiple data segments in one acknowledgement.

* Control Bits

6 flags (6 bit)

→ URG : Urgent pointer is valid

→ Ack : Acknowledgement is valid

→ PSH : Request for push

→ RST : ~~Request~~ Reset for connection

→ SYN : Synchronize sequence numbers

→ FIN : Terminate the connection.

→ One or more bits can be set at a time.

→ These bits help indicate connection establishment and termination, flow control.

* Window Size

→ this field defines size of data in bytes of the sending TCP process.

→ the maximum size of window is 65,535 by

→ normally referred to receiving window (rwnd)

→ the sender must obey the dictation of the receiver in this case.

* Checksum

→ 16 bit to detect errors in the transmitted segment while travelling through the network.

→ also presented in UDP header

→ mandatory in TCP not in UDP.

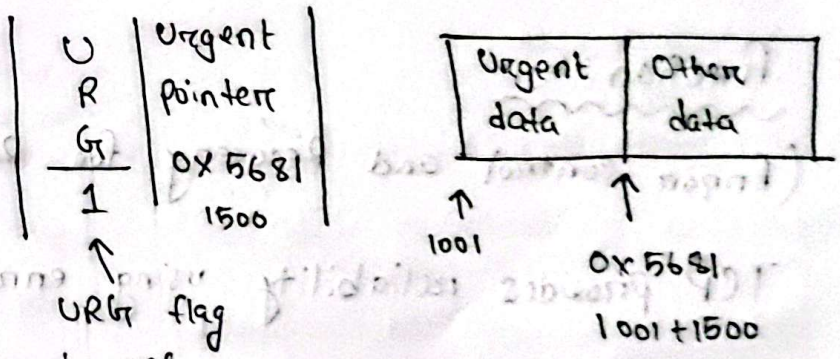
→ Same process for both.

* Urgent Pointer

→ this 16 bit field, which is valid only if the flag is set.

→ when used, it means segment has urgent data

→ it defines a value that must be added to the sequence number to obtain the number of the last urgent byte in the data section of the segment.



* Function : 5

(Connection establishment and termination for reliability)

TCP sets up a connection between end hosts before sending data. This is known as 'Three Way Handshake'

After connection is established the hosts can send data.

* function : 6

(Error control and Recovery for Reliability)

TCP provides reliability using error control

error mechanism includes

↳ detecting & responding corrupted segments.

↳ resending lost segment.

↳ storing out of order segments until missing segment arrives.

↳ detect & discard duplicate data.

* Error Control in TCP through

→ Checksum

→ Acknowledgement

→ Time out and retransmission.

Checksum

→ each segment includes a checksum field, which is used for checking a corrupted segment.

→ If a segment is corrupted, as detected by an invalid checksum, the segment is discarded.

Acknowledgement

→ Using acknowledgement number to confirm the receipt of data segments.

→ To confirm control segments that carry no data, but consume a sequence number.

→ ACK segments do not consume sequence numbers and are not acknowledged.

Retransmission

→ When segment is send, it is stored in queue until acknowledgement.

→ retransmission of segment will occur.

After retransmission Time Out

- the sending TCP maintains one retransmission time out (RTO) timer for each connection.
- When the timer matures TCP resends the segments in the front of the queue if segment is not acknowledged.

Out of Order Segments

TCP stores segment manually Doesn't send out of order segment. Waits, flags them out as out of order segments until missing one arrives. Out of order segment doesn't allowed by TCP.

Function: 07

(flow control and recovery for reliability)

→ TCP uses a sliding window for flow control.

What is the window?

→ indicates the size of the device's receive buffer for the particular connection.

→ how much data a device can handle from its peer at one time before it is passed to application process.

* Different TCP Sliding Window Protocols

• Selective Repeat Protocol

→ only those segments are re-transmitted which are found lost or corrupted.

→ keep track of out of order segments at the receiver side.

→ efficient for noisy channel.

→ widely used in TCP.

• Go Back N Protocol

→ if sent segment is found corrupted or lost then all the segments are re-transmitted from the lost segment to the last segment.

→ Don't keep track of out of order segments.

→ efficient for less noisy channel.