

Date ___ / ___ / ___

MOD 2

2.1 { ① Providing services .

② APP. layer paradigm

③ Client server

④ Peer to peer

2.2 { ③ Client server → APP. programming interface

④ Using services of transport layer.

2.4 { ⑤ Peer to peer → P2P networks , protocols }

⑥ Bit Torrent .

2.2 * ① Standard client - Server APP .

② DNS

③ HTTP

④ Terminal Network (TELNET)

⑤ Email (SMTP, MIME, POP3)

⑥ PTP

* ⑦ SNMP

Communication at Application layer is logical, not physical.

Providing Services:

- All communication net. made before internet were designed to provide services to network user.
 - The Internet was also designed for providing service to user around the world, but the TCP/IP protocol makes it more flexible.
 - When you remove a protocol from layer, care should be taken that next higher level protocol uses services accordingly.
- same when
when you add a protocol in a layer, care should be taken that it uses service provided by lower level.

- * * • The application layer is somehow different from all other layers in the suite.
- * • The protocols in app. layer do not provide any services to other protocols in the suite., they only receive services from transport layers
- * • Thus, protocol can be removed from this layer easily.
+ new protocol can be added as long as it can access service provided by transport layer.
- * app. layer provides service to internet users.

Date / /

Standard and Non-standard protocols

To provide smooth usage, the protocols for 1st four layers of TCP/IP should be standardized & included with OS.

App. layer can have both standard & non standard.

* Standard Application layer protocol

- There are several standard protocols of app. layers.
- Each standard protocol is a pair of computer programs that interact with user & transport layer to provide service to the user.
- In the case of app. layer, we should know, what kind of services the protocol provides.

* Nonstandard protocols :

- A programmer can create a non standard protocol, if the program can provide services to user by interacting with transport layer.
- No need of permission to use non standard protocols for private use.
- A pvt company can create a new N-S-P to communicate with all of its office around the world using services provided by first 4 layers of TCP/IP.

To use internet we require 2 app. programs to interact with each other

Date _____ / _____ / _____

Saathi

Application layer Paradigm:

- We know, to use internet we need ~~2 app~~ two application programs, one running on a computer & other on any other computer.
- The program needs to send message through Internet infrastructure.

* * Client Server Paradigm:

- traditional & most popular
- In this, service provider is an application program, called server process, it runs continuously, waiting for another application program, called client process, to make a connection through internet.
- server process → continuously running → waiting for client process to make a connect.
- Server process → continuously running
client process → when client needs something → needs to receive service.
- We cannot run a client - server process in vice-versa.
- Client communicates to Server to receive services.

Date / /

Problems (Client - server)

- The load is totally on server side process, thus server should be powerfull.
- Willing to accept the cost of make a powerfull server for specific service.
- * Client server paradigm is used by → HTTP, WWW, FTP & SSH & so on.

Peer to Peer paradigm

- A new paradigm (p2P) peer-to-peer has emerged.
- No need for server to be running all the time, waiting for client to make request.
- * The responsibility is shared between the computers i.e. peers
- * A computer can provide service at one time & receive service at one time & also can receive & provide service at both at same time.
- * Communication by phone is peer-peer, nobody needs to be waiting for other to call.
- * Another area in which peer-peer activity is seen is, when some computers are connected to internet & have something to share with each other.
eg: If one has to share a file, no need to be a server & keep running continuously, waiting till others retrieve.

* Peer 2 Peer is easily scalable & cost efficient.

Challenges

- ① Main issue is the security, it is difficult to create a secured communication.
- ② Not all applications can use this.

* Bit Torrent uses peer to peer.

#

Client Server Paradigm:

- In Client server system, server is the service provider, which keeps on running, to handle the requests sent by the client side for services. Server prepares a result & sends it back to client.
- ~~The server runs a server process~~
- Lifetime of server = infinite, running forever.
- Lifetime of client = finite, no. of requests.

Application Program Interface: (API)

* A process to be able to communicate with another process, we need a new set of instructions, to tell the lowest 4 layers of TCP / IP to open connection, send & receive data from other end, close connection.

* A set of instructions is called API → Application program interface.

first 4 layers of TCP/IP are built in OS

Saathi

Date _____ / _____ / _____

- * In this case, one entity is process at application layer other is the O.S. that encapsulates the ~~last~~ ^{first} 4 ~~last~~ layers of TCP/IP.
- * Computer manufacturer needs to build first 4 layers of TCP/IP in OS & include an API.
- * So as this is built in OS, process running at app layer is able to communicate with OS when sending & receiving messages through Internet.
- * Several APIs are designed for communication:
(socket interface, transport layer interface, stream)
- * Socket Interface: set of instruction that provide communication between Application layer of O.S.

Socket: is a Data structure created by & used by application program.

client

APP.

(socket interface.)

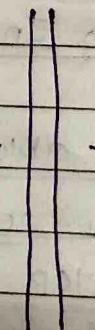
Transport

Network

Data-link

Physical

server.



OS

OS

IP → 32 bit integer address
port → 16 bit integer.

Date _____

Saathi

* We can use socket as a layer.

* Client to Server is a communication betn 2 sockets.

* Client → socket → server.

Socket Address:

(IP + Port)

• For two way communication we need address,
(sender address) & (receiver address).

∴ socket works as both sender & receiver.

∴ socket has two addresses. (pair of socket addresses)

Socket address is combination of 32 bit IP address
+ 16 bit port address

Finding socket Address

* The server needs a client address & server address
for communication.

SERVER

(sender)
↓
(server) # local socket address → provided by O.S. (IP add. of comp.)
& port address of server.

(Receiver)
↓
(client) # Remote socket address → since server can serve many
clients it does not have remote address beforehand.
this address comes from client → in the packet received.

CLIENT

(sender) # Local socket address : the local Client's socket address

↓
(server)

is provided by O.S.

- * The IP address of the computer + the port address of server.

** Remote server for (client) :

- When client starts the process, it should know the socket address of server it wants to connect.

both IP + port no. sometimes, user who starts the client process knows the both IP address + port address of computer on which server is running.

* • Each application has standard port no., we sometimes don't know IP address.

. In many cases we know the server for eg: gmail. so we know the server as (identifier), to the corresponding server socket address.

** So IP address can be obtained, using DNS. Domain Name System.

DNS is used in application layer to obtain IP address.

using services of Transport layer:

- A pair of processes, provides Internet Services to user.
- * • A pair of processes, needs to use the services provided by transport services layer for communication. As there is no physical communication at app. layer.
- * Three common transport layer protocols in TCP/IP:
① UDP, ② TCP ③ SCTP
- * We decide which type of protocol we need at transport layer while making an Application.

* UDP Protocol (User datagram protocol)

- Provides connectionless, unreliable datagram service.
- No logical connection betn two ends exchanging msg.
- Each msg is an independent entity called datagram,
- No relation betn datagrams' (entities).
- Not a reliable protocol, it checks if data is corrupted, but doesn't ask user to resend the data again
- * * advantage: it is msg oriented, boundaries to the msg.
- If app. is sending small msgs & speed is important then UDP is used.

* TCP Protocol :

- connection oriented, builds a logical connection betn 2 ends of data exchange.
- Reliable & has byte stream service.
- Also checks the flow, error & congestion control over the messages.

If data is lost, it asks sender for resending the data making TCP reliable.

- * • disadvantage: not msg oriented.

* reliability is seen & can send long msgs.

* SCTP

- combination of ~~TCP~~ two other protocols
- Provides connection oriented, reliable service.
- No byte stream oriented.
- Msg oriented like UDP, multistream.

* Best

- * Suitable for applications that need reliability & remain connected even if failure occurs.

Peer to Peer Paradigm#. P2P networks:

* Internet users becomes peers & form a network.

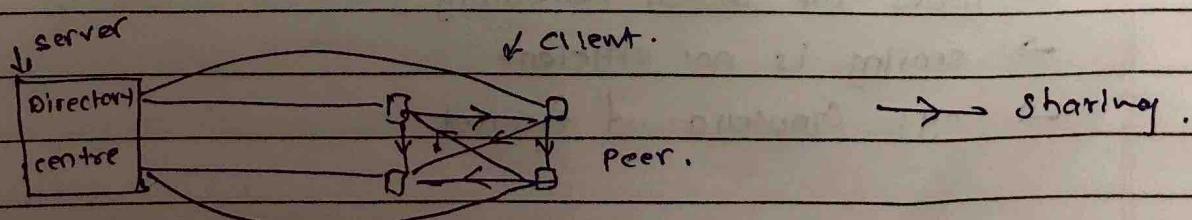
- * When a peer in the network has a file to share, it makes it available for rest of peers.
 - * An interested peer can connect itself to the computer where file is stored & downloads it. After that he can also let other peers download it.
- # Responsibility is shared among peers

Thus, the count of peer may gain or shrink; thus we need to divide P2P networks to 2 categories:

- ① Centralized
- ② Decentralized

* Centralized P2P network : (Client-Server)

- * In Centralized P2P network: the directory ~~sys~~ system of what listing of peers & what they offer is client-server
- Storing & downloading is done using P2P.



Date _____

~~working~~

- A peer looking for a file, sends request to directory server, server responses with IP address of nodes.
- Peer then contacts the nodes & downloads the files.
- Directory is updated as shrink & gain of peers.

Advantage / Disadv.

adv

- Makes maintenance easy

disadv.

- Increases traffic & slowdowns

dis

- Vulnerable to attacks

disadv.

- Whole system goes down

#

Decentralized P2P network

- No client - server system / directory system is seen.

#

- Peers arrange themselves on a logical ~~network~~ layer above the physical ~~layer~~ network.

- Depending on how the nodes are lying on logical network, decentralized P2P is classified into 2 types of structures: ~~unstructured~~ unstructured.

Unstructured

→ nodes are linked randomly

→ searching is not efficient

→ eg: Gnutella & freenet

Structured P2P networks

- uses a predefined logical network or set of rules to link nodes.
- Efficient & effective.
- DHT is used. in p2p file sharing.

#

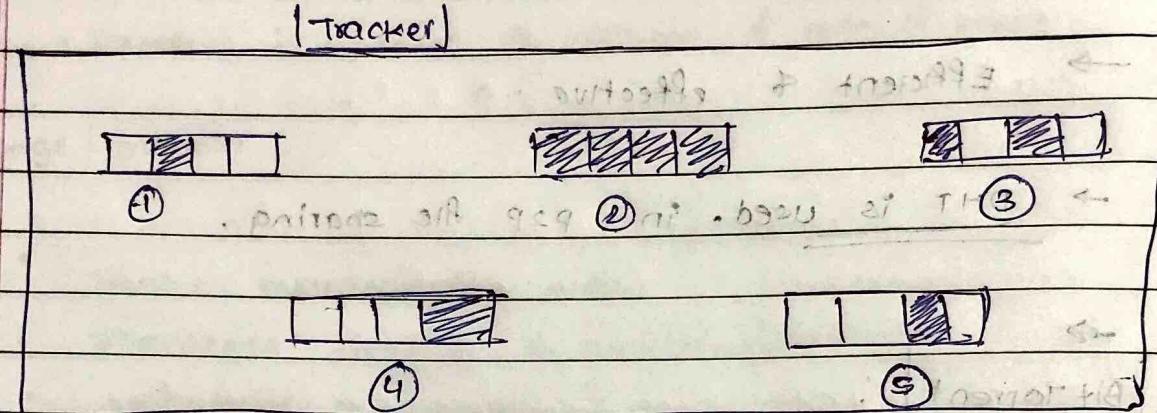
[BitTorrent]:

- BitTorrent is a p2p protocol, designed for sharing large files among peers.
- A group of peer takes part in the process where, all other peers in the group get the copy of the file.
- File sharing is done collaboratively → Torrent
- The set of all peers that take part in the file sharing are called swarms.
- 4 one who has complete file in the swarm = seed
- 4 one who has a part & wants rest = leech
- swarm is combination of seeds & leeches

Date ___ / ___ / ___

Bit-Torrent with a Tracker: (Tracker → central node)

→ Tracker tracks operations of swarms



2 → seed, 1, 3, 4, 5 → leecher

→ here 2 already has all the parts of file. thus

→ uploading & downloading continues;
some peer may leave, some may join.→ Now if a new peer joins the Bit-Torrent server,
with the file name.→ He will get a metafile & address of the
tracker, that will help him find the file
he wants to download [It gives address
of peers who have file to download].

→ It can leave the torrent anytime he wishes.

- To avoid overloading a single peer for download, it can restrict himself from no. of connection default → 4.
- BitTorrent protocols force to provide a balance b/w no. of pieces each peer may have by using strategy rarest first.

Trackless BitTorrent:

- If tracker fails, new peers cannot connect to network & updating is interrupted.
- * → Then the job of tracking is distributed among several nodes in the network.
- DHT can be used ~~to~~, here.
- If we use hash function of metadata as the key and hash function of list of peers, in swarm as value we can let some nodes in P2P play role of trackers.
- New peer joins & sends key of metadata to peer, if key matches → file is present.
The responsible node sends the value, i.e. actually list of peers in corresponding torrent.
- Now, peer can join correct torrent.