

# Computer Networks

## Network :

It is a group or system of interconnected people or items. Computers connected with each other with cables or wires is computer networks.

"A Network of computers is called computer networks."

???

Why do we need networks ?

Because we want to share resources, files, ppt's etc. (example) All the iphones and Apple users, are they can share files and everything using AirDrop. So they are connected via a network. They use wifi & bluetooth. Using these wifi or bluetooth they can share files seamlessly.

It means sharing of resources and communication is the main agenda.

## Internet :

In a nutshell, internet is a network of computer networks. Complex web of interconnected computer networks.

Advanced Research Project Agency

## History of Internet

1957

Soviet

sputnik

↑  
ARPA

↓  
US

1960's - 70's

↓  
communication system for  
ARPA's computer to  
talk

1969

→ ARPANET

→ 1980's → 1990's → They started using hyper  
 ↓ link based documents  
**CERN**  
**TCP/IP**

↳ Hyper text is a text which is comprises of Hyperlinks  
 One document is linked to other document

→ 1990's → Tim Berners Lee who introduces us with world wide web (WWW).

Then after some time's the browsers comes into the picture. i.e. Mosaic

Network

## Terminologies

\*

Protocols → It is a set of rules and regulations setup to communicate and share information over a network.

Ex. HTTP, UDP, TCP, SMTP

Packets

In order to share data, we can't send big chunks of data over the network. So, we divide the data in smaller chunks, these small chunks are called packets.

Having smaller chunks of data makes the overall network more manageable, more reusable and definitely move fast. So in this way, we can leverage the whole capacity of Network

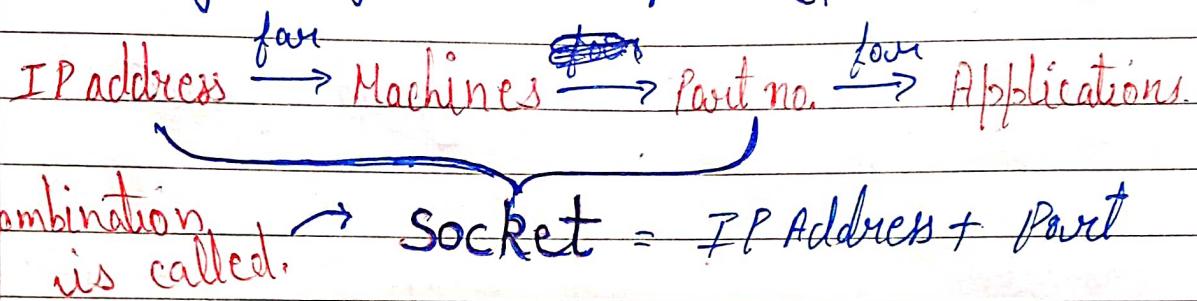
→ This is for the device.

## Address

Sending messages over the networks requires the destination details, this detail uniquely identify the end system is called address.

## Ports

Any machine could be running many networks applications in order to distinguish these apps for receiving messages we use ports. (port number)



- Ports help us get the packets to specific process on host

↳ Every process has 16 bit port number

$$0 - 2^{16} = 65535$$

↳ Range of port no

- \* 0 - 1023 → well known ports → reserved ports

These ports are Ex. port 80 → http  
system defined port 443 → https

- \* 1024 - 49152 → Registered ports

they are used by specific, potentially proprietary apps / processes that are known but not system defined.

Registered port means any third party application can use these range of ports →

1024 - 49152  
ports

Ex:- SQL server = 1433  
MongoDB = 27017

So applications, end user applications, proprietary software or open source software are going to use these ports.

We can change these range, there is no issue. Because these are not defined by the systems

\* 49152 - 63535 → Dynamic ports

## Access Networks

These are media using which end systems connect to the internet. Ex wifi, satellite, Telephone,

\* Network Interface Adapters :-

It enables a computer to attach to a network. As there are different types of network, it acts as a single suit to connect to any network.

\* DSL = (Digital Subscriber Line)

DSL uses the existing telephone groundwork lines for internet connection. Generally DSL is provided by same company which supplies telephone service.



## ISP (Internet Service provider)

It is just a company that provides end users internet connections.

Ex:- AT&T

### Network Protocol stacks

OSI (7 layer)

TCP/IP (5 layer)

Application → email, chat services

Application → combination of 3 layer

Presentation → Presentation of data,

Transport

Session → comprehension, encryption

Network

Transport → Imp. layer, divides big

Data Link → error/flow

chunks of data above to small chunks and manages

Physical → control, there is concept of multiplexing & demultiplexing

Network → these chunks

It also handles addressing which address to packets belong

Data arrives at the cable connections or satellite or dialup

connections network, this is the physical layer, cable carries on which the data actually comes in electronic form and transferred in the electronic form over in the form of signals or waves.

It handles routing of packets will be done on the network

## \* Application Layer

- Roles →

- 1) Writing / providing data off to the network
- 2) Reading the data from the user,
- 3) contains applications that helps users to interact on the network.
- 4) Error handling & recovery can also be done.

- Where it exists??

↳ only in end systems

→ Instant messaging

→ WWW

→ VOIP (Voiced Over Internet based Protocols)

→ email

## \* How exactly Application layer works.

→ Client-Server Architecture ⇒

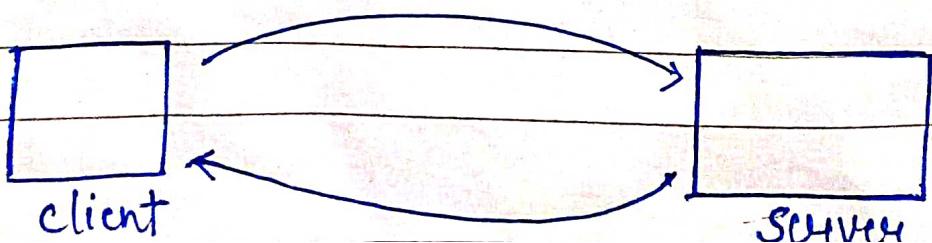
- It is a two level architecture

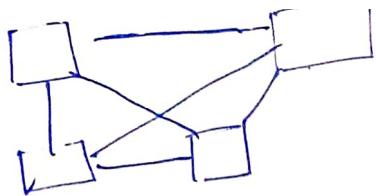
client side

server side

Server : This process controls access to centralised resource or service such as a website / Web app.

Client : It is like a frontent where user interacts.





} example  
Torrent

Date: \_\_\_\_\_  
Page: \_\_\_\_\_

\*

## P2P architecture (Peer to peer)

A P2P architecture consists of a decentralized network of peers - nodes that are both client and servers.

P2P networks distribute the workload between peers and all peers contribute and some consume resources within the network without the need for a centralized server.

\*

## Hybrid Architecture

Combination of client server and P2P architecture

Protocol

\*

## HTTP

- stands for hyper text Transfer protocol

\*

## URL (uniform resource locator)

- protocol
- hostname
- location of file/object
- arguments

http://flipkart.com/image/23.jpg?g=50

protocol

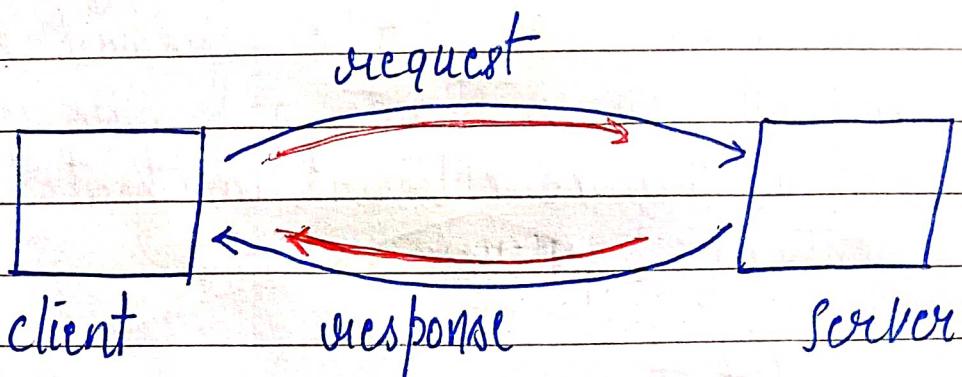
hostname

filepath

extra  
arguments

## \* HTTP

It defines the whole procedure on how client and servers will interact.



- Request:

Request is the part of information that client initiates or source initiates to the destination.

That OK, this is my request, this kind of resource I want to access, please give me the resource.

- Response:

And after processing the request, a response data is send from the server to the client or receiver, that OK, this is the resource that you want to access.

HTTP is also categorised in request-response protocols.

- the first msg → HTTP request
- the second msg → HTTP response

→ HTTP is stateless protocols

→ The server does not stores information about the client.

Xhr = XML http request  
kind of http request

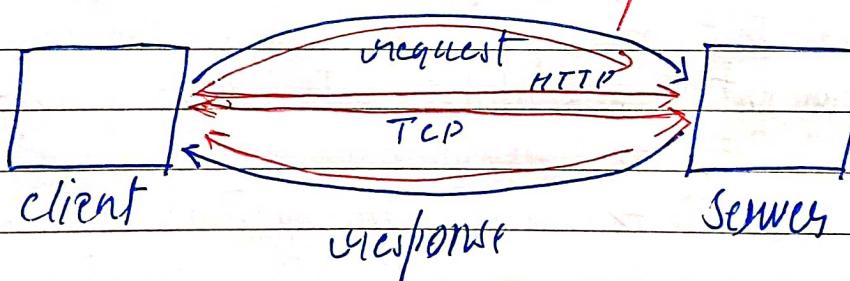
Data: \_\_\_\_\_  
Page: \_\_\_\_\_

So, a lot of application layer protocols depends on lower level protocols of transport layer.

⇒ In Transport layer there are 2 main protocols  
① TCP                    ② UDP

⇒ **HTTP depends on TCP**

Whenever we want to send payloads over HTTP, we can't send directly it. Every time, no matter what, first of all we have to establish TCP connection. The TCP connection once established, the moment over that TCP connection, we establish the HTTP connection, and then based on that, we send a request over that http connection and receive response



\* There are two types of HTTP connection: —

① Persistent HTTP

The moment we setup a TCP connection and then http connection after that we send a request or receive a response, this life cycle continuous.

Ex:- Web socket

② Non-Persistent HTTP

The moment one request/response cycle is done, the http connection breaks. It's not persistence, everytime it has to send a new request



## HTTP req and response messages

HTTP Req Msg  $\Rightarrow$

Any http message are plain ASCII text

$\hookrightarrow$  host

$\hookrightarrow$  method

$\hookrightarrow$  status code

$\hookrightarrow$  ~~Referer~~ policy

So lot of details combined makes a new HTTP request



header  $\Rightarrow$  header is like some extra piece of information that you want to send with your request or response.

$\Rightarrow$  There are multiple http methods

1) Get  $\rightarrow$  request some data

2) Post  $\rightarrow$  put some data on the server

3) Put  $\rightarrow$  update data on the server

4) Patch  $\rightarrow$

5) Delete  $\rightarrow$  delete an object at a given url.



Cookies :

$\hookrightarrow$  These are mainly concerned towards privacy

$\hookrightarrow$  http is a stateless protocol but a lot of time user session is required.

???

How cookies work ???

Cookies are unique identifier strings. These are set by the server through http headers, as soon as a cookie is stored, it is sent

along with subsequent http req to the same server. This allows server to know who is contacting it and hence serve the content accordingly.

- **Set cookie headers** ⇒

When a server wants to set a cookie it includes "Set-cookie : value" in the http response. This value is stored in the cookies file of browser.

\* **SMTP** → Email  
(Simple Mail Transfer protocol)

⇒ for executing the functionality of email, SMTP is used. One more protocol named POP3 is used in combination with SMTP.

One is used to send emails that are stored in the users inbox and other is used to retrieve emails sent to a user.

Similarly as SMTP also uses TCP protocol from transport layer.

Connection for SMTP is setup on port 25.

Mail client gives the actual UI for end users to send and receive mail via gmail, outlook etc.

## Ques \* How SMTP works?

- when an email is sent, it is sent to the sender's SMTP server using SMTP protocol.  
(Also the SMTP server is configured in the mail clients.)
- The SMTP server places the email on a message queue.
- Then SMTP server initiates a connection with receiver's SMTP server and conducts an initial SMTP handshake.
- Then finally it sends the email to recipient's SMTP server.
- The mail is downloaded from receiver's SMTP server and then the client shows the mail.

 **SMTP** → Push protocol  
Sending the email

 **POP3/IMAP** → Pull protocols

 **downloading the email.**

- If recipient SMTP server is offline, the sender SMTP server tries again and again after some delta mins. There is a set threshold after which it stops sending the email & marks it delivered.

Latest version is 3, that's why it is called POP3

Date: \_\_\_\_\_  
Page: \_\_\_\_\_

## \* POP (Post office Protocol)

→ It downloads emails in 4 phases.

1. Connect
2. Authorize
3. Transaction
4. Update

### • Two modes of POP

↳ download and keep it

↳ download and delete.

## \* IMAP (Internet Message Access protocol)

- Emails are kept on the server & not deleted.  
Local copies of the emails are cached on each device.
- If an email is deleted by user manually then only gets deleted from server.

## \* Transport layer

It plays an important role in establishing the communication between application layer and network layer. So it acts as a sandwich layer in b/w app. and netw. layer.

Whenever we suppose to transfer the data from the application layer to network layer, that is, in the case of sender, then transport layer plays an imp. role in that.

Also when network layer is going to transfer data from to the application layer, in that case also when we are in the receiver end.

# Transport layer and its protocols reside at end systems.

??? because we are going to take the data from the application layer and then provide it to the corresponding network layer.

And at the receiver's end the network layer is going to give the data and we need to manipulate the data so we can transfer it to the application layer.

??? TCP : — (Transmission Control Protocol)  
↳ also acknowledgement based protocol  $\xrightarrow{\text{Transport layer protocol}}$

# What TCP does?

① appropriate Send data  $\rightarrow$  Segment = collection of bits

② Segment data

③ transmit Congestion control

④ write Identity and sequence number.

App data

segment data

Transmit

Acknowledgement based protocol  $\Rightarrow$  It gives us the acknowledgement of whether the delivery of the corresponding data packets was successful or not.

So there is a feedback mechanism which acts as an acknowledgement for the sender, whatever packet send to the receiver has been successfully received at the receiver.

**Example****Application :**

1. FTP : File Transfer Protocol
2. SSH : Secure Shell Protocol
3. Email :
4. Web browsing : HTTP/HTTPS

It allows us secure connection to the a remote host over an unsecure host network.

**UDP :-**

User Datagram Protocol  
It is one of the transport layer protocols.

For TCP, we have packets called Segments and Here, packets will be called datagram.

**Why UDP?**

- ① faster
- ② reliability can be built separately
- ③ small header size gives an edge in terms of reduced transmission overhead & quicker transmission times.

## # Applications

- Xbox
- Name translation of DNS

**IP : (Internet Protocol)**

IP is Network layer protocol