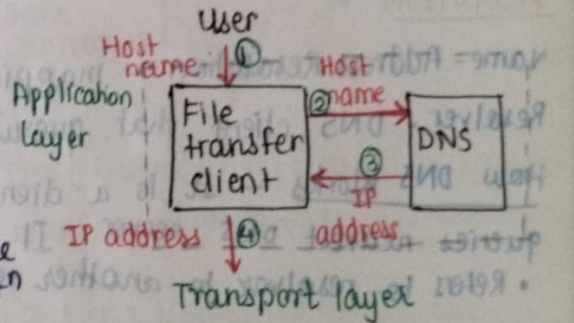
UNIT 1 : APPLICATION LAYER

APPLICATION LAYER PROTOCOLS :-

1. DOMAIN NAME SYSTEM (DNS) –

- TCP/IP uses IP addressing to identify entities.
- People prefer names over numeric addresses.
- A system is needed to map names to IP addresses & vice versa.
- Old method - Each host-file with name-address pairs (scalable ✗).
- Problems - Large file, difficult updates, high traffic.
- Solution - DNS :→ Info is divided into smaller parts & stored on multiple computers
  → A host queries the nearest DNS server for name-to-IP mapping.
  → Reduces traffic and improves efficiency.
- DNS : FUNctionality : Converts domain names to IP addresses.
  Works in a distributed & hierarchical manner.
  Ensures efficient & scalable mapping of names to addresses.

Definition - DNS is a hierarchical and distributed system, that translates human-readable domain names (eg. www.google.com) into IP addresses (eg. 142.250.183.260), allowing computers to locate & communicate with each other over the internet.

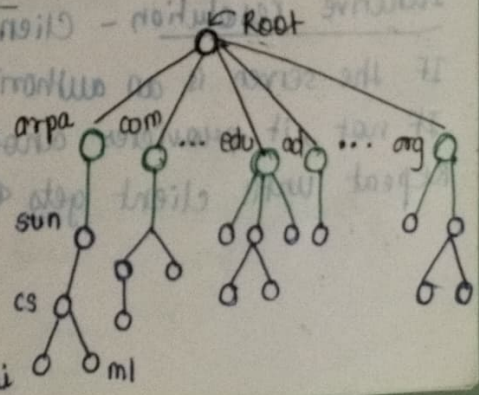


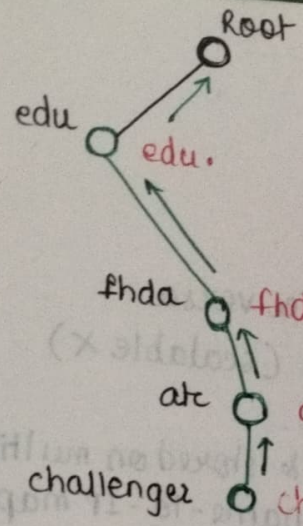
NAME SPACE –

- Every machine must have a unique name to avoid conflicts. (Need)
- 2 ways – Flat or Hierarchical.
- Flat Name Space : Each name is simple sequence of characters without structure. No relation bet^n names. Risk of duplication. Not suitable for large systems like the Internet (as it requires centrally control to avoid duplication)
- Hierarchical Name Space : Names are divided into multiple parts (like a tree structure) Helps decentralize name management. Eg. "challenger.fhda.edu", "challenger.smart.com", "challeger.berkeley.edu". – even if "challenger" is common, full names remain unique.

DOMAIN NAME SPACE –

- To have a hierarchical name space, domain name space was designed. In this design, the names are defined in an inverted-tree structure with the root at the top. Tree can have only 128 levels : level 0 (root) to level 127
- Labels - Each node in heirarchy has a label, (ie string of max. 63 characters). Root label is a null string. Children of node must have different labels (uniqueness)
- Domain Name - Each node has domain name. Domain Name is sequence of labels separated by dots (.). Always read from node upto the root.





- Domain name
- Label

The last label is root (null).
Full Domain name always ends with root, (null label), so last character is a dot (.)

DNS → Domain Name Space
- Generic Domain → Classify host based on organizational type
- Country Domain → represent geographic locations using 2-character country codes
- Inverse Domain → maps IP addresses back to domain names

GD eg. .com / .edu / .gov / .net / .org
CD eg. .us / .in / .uk / .ca
ID eg. uses .arpa domain eg. 121.45.34.132.in-addr.arpa.

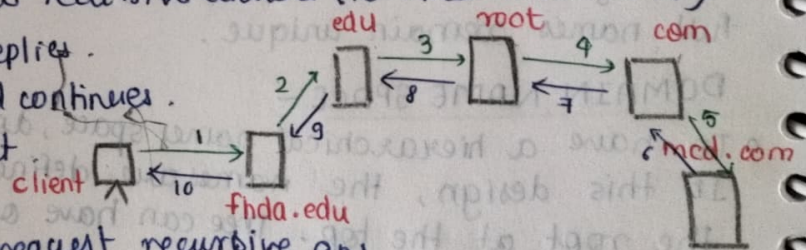
RESOLUTION –

Name-Address Resolution - mapping a name to address or address to name.
Resolver - DNS client that queries DNS server for mapping.

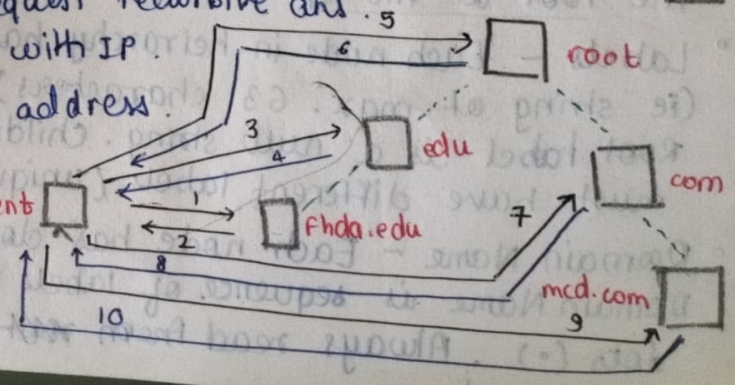
How DNS Works – It is a client-server system where a client (resolver) queries nearest DNS server. If server has mapping, it responds; otherwise
- Refers to resolver to another server & Asks another server to provide info.

Mapping Names to Address: The resolver gives domain name & asks for IP
The DNS server checks: Generic & Country domains. If local server can't resolve it queries other servers.

Recursive Resolution – Client request a recursive answer (ie final resolution)
If queried server knows the answer, it replies.
If not, it queries parent server, and continues.
When resolved, response back to client.



Iterative Resolution – Client does not request recursive ans.
If the server is an authority, it replies with IP.
If not, it provides another server's address.
Repeat until client gets IP.

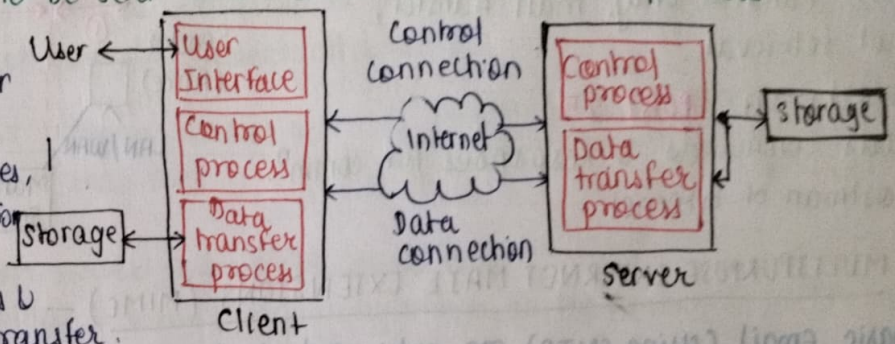


2. FILE TRANSFER PROTOCOL (FTP) & TRIVIAL FILE TRANSFER PROTOCOL (TFTP)

- FTP is a standard TCP/IP protocol used for transferring files between 2 computers over a network.
- Problems in transferring files → different file naming conventions, text & data in different formats, and directory structures of 2 systems maybe different.
- FTP provides solution to all these problems. It establishes 2 types of connections between the client & server, one for data transfer and the other for control information.
- Control connection (Port 21) – used for sending commands & responses
  – stays open throughout FTP session
  Data connection (Port 20) – used for transferring files
  – opens & closes for each file transfer.

Why 2 connection ? → Keeps control & data separate, making FTP more efficient. Allows commands to be sent while a file transfer is in process

1) Client initiates connection to the server on Port 21 (control).
2) When transferring files, a separate Data connection (Port 20) is created.
3) Data connection opens & closes for each file transfer.



Commands →
- Access commands → USER (user info) | PASS (Password) | QUIT (Log out of system)
- File management commands → DELE (delete file) | RMD (Remove directory) | MKD (create new directory)
- Data formatting commands → TYPE (Argument : A, E, I, N, T) (Define file type) | STRU (Define organization of data) | MODE (Define transmission mode)
- Port defining commands → PORT (Client chooses a port) | PASV (server chooses a port)
- File transfer commands → RETR (Retrive files) | STOR (store files) | STAT (Return status of files)

TFTP:

File transfer protocol that is designed for specific use cases where full features of FTP are not necessary. ie files transferring without authentication & without any separation of control info & data as in FTP.

Commands :- DATA - Transfer data file in blocks.
RRQ - read request (download)
WRQ - write request (upload)
ACK - acknowledge receipt of data block.
ERROR - indicates error during transfer process

	FTP	TFTP
Protocol used	TCP	UDP
Authentication	Yes (Password)	No
Control & data	Separated	Not separated
Ports	20 → data 21 → control	69
Data transfer	Reliable	Unreliable
Complexity	More	Simple, lightweight protocol
Speed	Slow	fast
Data transfer mode	Supports both ASCII & binary mode	Only binary mode.

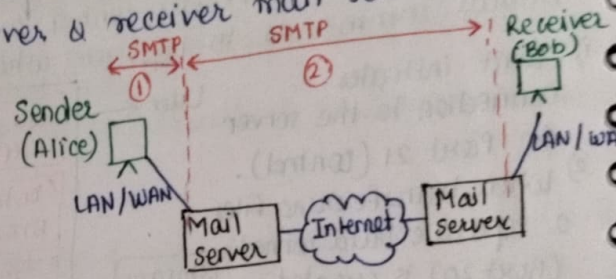
3. (SMTP) SIMPLE MAIL TRANSFER PROTOCOL —

- MTA (Message Transfer Agents); responsible for actual transfer of email betn mail servers.
- SMTP is an application layer protocol used for sending & relaying email msg over the internet.
- Sender writes an Email → Email send to SMTP server → Port 25 → DNS for finding address → SMTP relays (forwards) → Email receive to receiver ← Receiver picks/takes (POP3/IMAP) email
- A system should have client MTA to send & should have server MTA to receive.
- SMTP → defines MTA client & server in the internet.
  SMTP used 2 times → 1) betn sender & sender's mail server
  2) sender mail server & receiver mail server.
- SMTP handles only mail transfer, not retrieval.
- Works on PORT 25
- Uses Commands & Responses for comm^n.
- Postman of internet.

Same office = No mail carriers needed.
Diff office = Needs postal service (MTA)
Sender far away = Needs local courier + postal service
Both far away = local + postal + home delivery (MAA's)

uses MTA Message transfer Agent

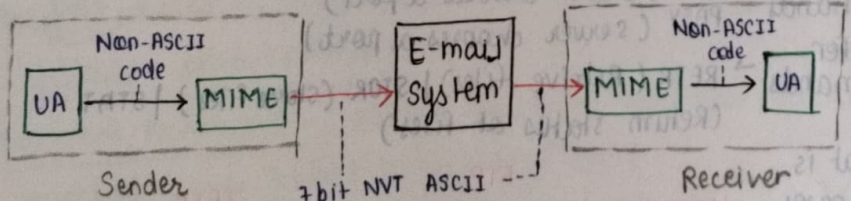
PUSH Protocol



4. MULTIPURPOSE INTERNET MAIL EXTENSIONS (MIME) —

- Basic Email (using SMTP) can only send text messages. It can't handle
  → Non-English languages
  → Pictures, videos or audio files.
- MIME solves this problem by converting these types of data into a format that can be sent & then converting back to their original form when received. (non ASCII → ASCII)

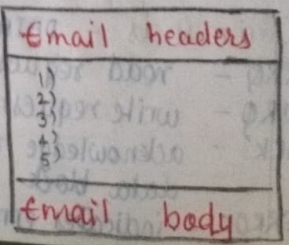
MIME extends SMTP



MIME defines 5 headers to email

1) MIME-version : Defines the MIME version (commonly 1.1)
2) Content-Type : Specifies what type of content is included in email. It has

Primary Type	subtype	
text	plain, html	simple text or HTML document content
multipart	mixed, parallel, alternative, digest	Emails with attachments / multiple files
image	jpeg, gif, png	Inline images / attachments
audio	basic	Sound files
video	mpeg	Video files.



3) Content Transfer-Encoding : Defines how email body is encoded for transmission

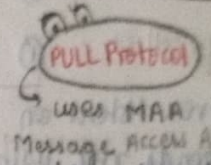
Encoding
- 7 bit → Default ASCII encoding
- 8 bit → supports special characters but requires handling
- base64 → used for img, video, PDF & binary Data.

SMTP only support 7bit ASCII, so other formats need encoding.

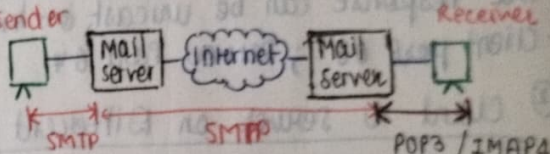
4) Content-Id : Assign a unique identifier.

5) Content-Description : defines whether whole body is image, audio or video.

5. (POP3) POST OFFICE PROTOCOL Version 3 AND (IMAP 4) INTERNET MAIL ACCESS P V4 -



- Simple and limited protocol for email retrival.
- Works on TCP Port 110.
- Downloads email from server to the user's device.
- Has 2 modes: 1) Delete mode → Emails are removed from server after download
  2) Keep mode → Emails stay on the server (useful when accessing from different devices)
- Limitations : 1) No option to organize emails on the server.
  2) Cannot preview email content before downloading.
  3) Cannot partially download emails.



INTERNET MAIL ACCESS PROTOCOL version 4 —

- More powerful & flexible than POP3.
- Works on TCP port 143.
- Keeps email stored on server instead of downloading.
- Supports multiple folders & allows organizing emails on the server.
- Features : 1) Checks email headers before downloading.
  2) Searches for keywords in email before downloading.
  3) Partially download large emails (useful for attachments).
  4) Organizes emails into folders directly on the server.
  5) Access emails from multiple devices.

Parameter	POP 3	IMAP 4
Protocol is defined at	RFC 1939	RFC 2060
TCP port used	110	143
e-mail is sorted at	User's PC	server
e-mail is read	offline	Online
Time required to connect	Small	Long
Use of server resource	minimal	Extensive
Multiple mail boxes	Not possible	Possible
Who backs up mail boxes	User	ISP
For mobile users	Not good	Good
User control over download	Little	Great
Partial msg download	No	Yes
Simplicity in implementation	Yes	No
Support	Wide-spread	Increasing.

POP3 ⇒ Where? used betn recipient's email client & mail server to download emails
When? when user wants to download emails on single device. & when internet access is limited.

IMAP 4 ⇒ Where? used betn recipient's email client & mail server to read emails without downloading them permanently.
When? i) when want to access emails from multiple device.
ii) organize emails on server
iii) need to search or preview emails before downloading.
Eg. Gmail, Yahoo, Outlook, etc

6. DYNAMIC HOST CONFIGURATION PROTOCOL (DHCP) –

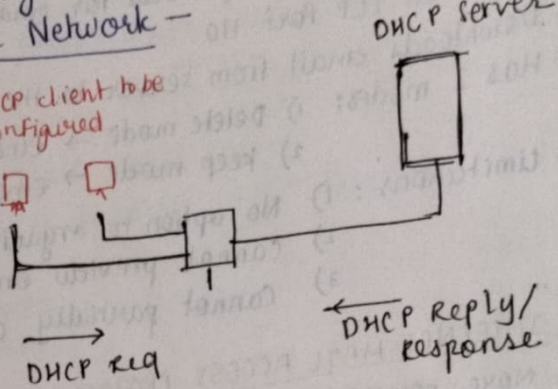
- DHCP is a client-server protocol that assigns IP addresses dynamically to devices on a network.
- It is successor to BOOTP & is backward compatible it.
- Helps in automatic configuration of devices, eliminating the need for manual IP assignment

Why is DHCP Needed ?
1) Avoids manual IP config. errors.
2) Efficiently manages IP address allocation in large network.
3) Supports both static & dynamic address allocation.

DHCP Operation :

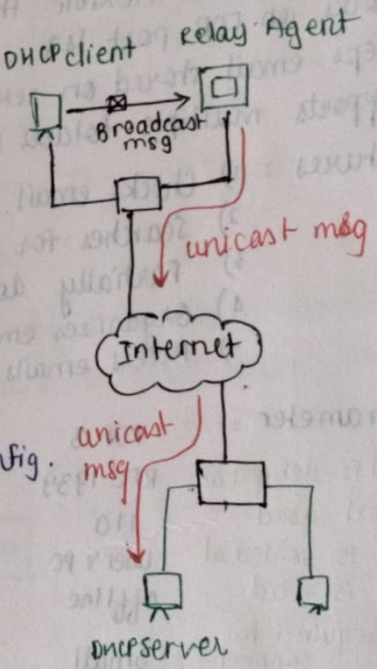
① Client & Server on Same Network –

- DHCP server listens on UDP port 67.
- Client sends req using UDP port 68.
- The client uses 0.0.0.0 as its source IP & 255.255.255.255 as destination IP.
- The server responds with an IP address & configure settings.
- The response can be unicast or broadcast
- Client port 68 ; Server Port 67



② Client & Server on Different Networks –

- A router or another device acts as a relay agent.
- The relay agent forwards the client's broadcast request to the server as a unicast message (direct comm^n).
- The server replies to the relay agent, which then sends the response back to the client.



DHCP Message Types :
1) DHCP DISCOVER – client broadcasts for an IP.
2) DHCP OFFER – Server responds with an available IP & config.
3) DHCP REQUEST – Client requests the offered IP.
4) DHCP PACK – server confirms & assigns the IP.
5) DHCP NAK – Server rejects the req.
6) DHCP DECLINE – Client declines the offered IP.
7) DHCP RELEASE – Client releases its assigned IP.
8) DHCP INFORM – Client req. config. settings without an IP.

7. (TELNET) TELEcommunication NETwork –

Telnet is a protocol that allows users to remotely access & control another computer or device over the internet or a local network. It works through a text based interface, enabling users to interact with the remote system as if they were physically present.

Cookies

WWW was a stateless system ie once client request & server responds, the connection ends - there is no memory of past interactions.

Modern website need state-ful interaction

- Shopping websites
- User authentication
- Advertising

So cookies were introduced.

Cookie is small piece of data stored on user's device by a website It helps remember info about user across multiple visits.