UNIT 4 APPLICATION LAYER PROTOCOLS

Q: What is DNS and enlist needs/purpose of DNS?

1. DNS ensures the internet is not only user-friendly but also works smoothly, loading whatever content we ask for quickly and efficiently.
2. It allows the user to access remote system by entering human readable device hostnames instead of IP address. It translates domain name into IP addresses so browser can load internet resources.
3. It translates human readable domain names into the numerical identifiers associated with networking equipment, enabling devices to be located and connected worldwide.

Analogous to a network "phone book," DNS is how a browser can translate a domain name (e.g., "facebook.com") to the actual IP address of the server, which stores the information requested by the browser.

Q How to map domain name with IP address

- TCP/IP uses DNS client and DNS server to map a name to an address
- It can also perform reverse mapping

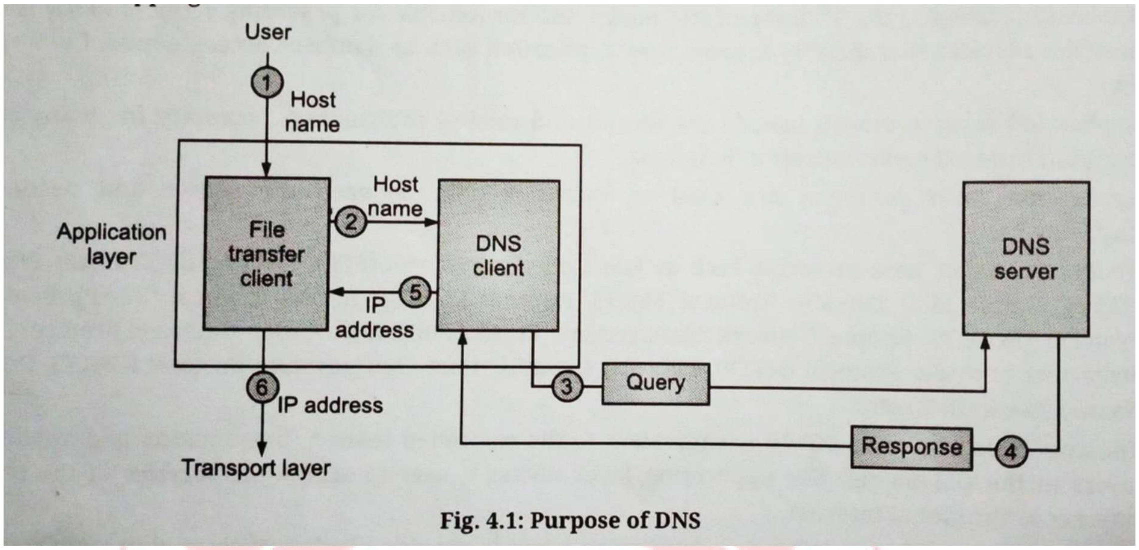
- 

Fig. 4.1: Purpose of DNS

- When user wants to use file transfer client to access corresponding file transfer server running on remote host
- If user knows only file transfer server name.
- But TCP/IP suite needs the IP address of file transfer server → to make connection
- Six steps to map host name to an IP address
- **Step 1**: The user passes host name to file transfer client
- **Step 2** : file transfer client passes the host name to DNS client
- **Step 3**: each computer after being booted knows the address of one DNS server. The DNS client send s a messages to DNS server with a query that gives file transfer server name using known IP address of DNS server
- **Step 4**: DNS server responds with IP address of the desired file transfer server
- **Step 5**: The client passes IP address to file transfer client
- **Step 6**: File transfer client now uses received IP address to access file transfer server
- DNS is a network protocol used to translate host names to IP addresses

Q: What is Domain Name Space (DNS)? Explain all terms in detail?

- DNS (**Domain Name System)**
- Domain name space consists of tree data structure
- Name space is abstract space or collection of - all possible addresses names, identifiers of objects on a network, inter network, internet.
- DNS should assign names unambiguously
- Names are bound with IP addresses
- Bound in two ways → Flat name space and hierarchal name space

  Name spaces in DNS

  1. Flat Namer space

  - Names are assigned to an address.
  - Name in this space is a sequence of characters without structure.
  - Names may or may not have a common section → if they do – it has no meaning
  - Advantage: Names are convenient and short
  - Disadvantage: It cannot be used for Large systems because it must be controlled to avoid ambiguity and duplication.

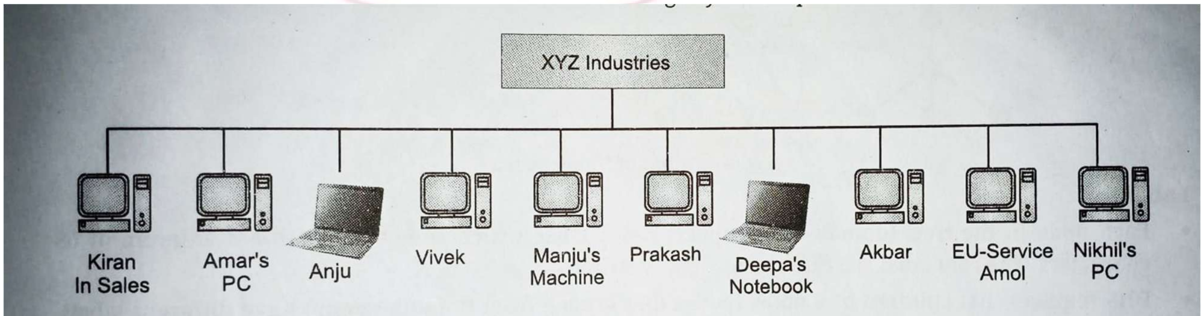


Fig. 4.2

2. hierarchal name space

- Each name is made of several parts
- First part defines nature of the organization
- Second part defines name of the organization
- Third part defines departments of the organization and so on
- Here authority to assign and control the namespaces can be decentralized
- Authority for names in each partition is passed to each designated agent

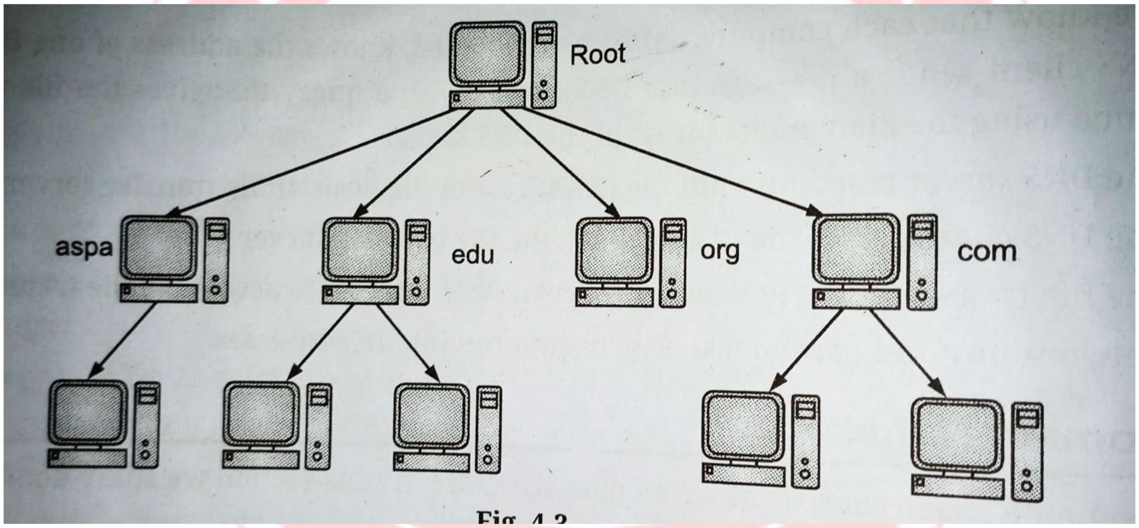


Fig. 4.3

- **Domain name space:**
- It was designed to have hierarchal name space,
- Names are defined in inverted tree structure with root at the top, tree can have only 128 levels: Level 0 (Root) to level 127
- It refers to hierarchy in internet naming structure

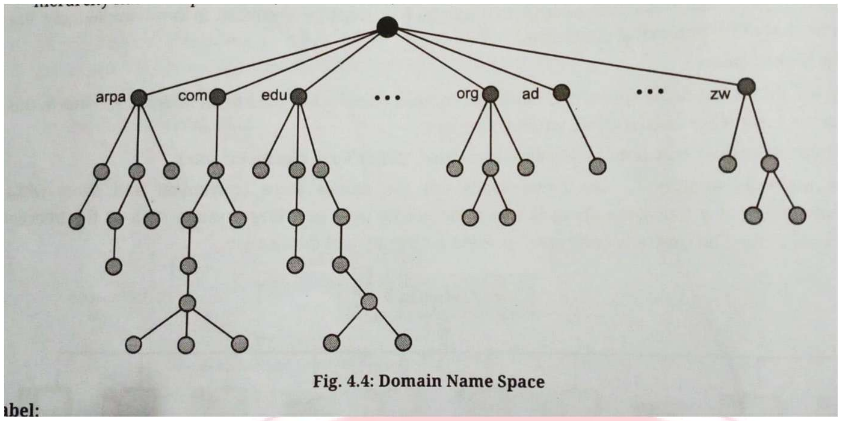
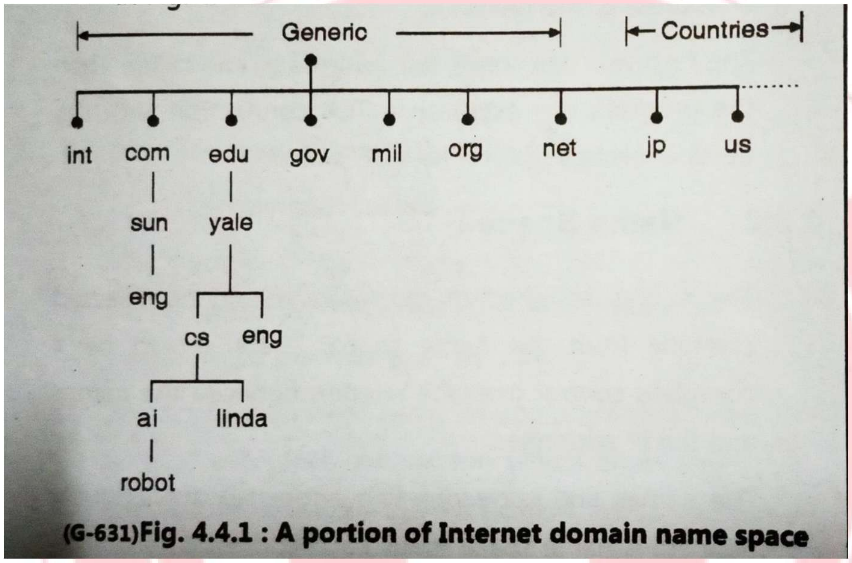


Fig. 4.4: Domain Name Space



(G-631)Fig. 4.4.1 : A portion of Internet domain name space

- **Label**
  - Each node has a label
  - It's a string with maximum 63 characters
  - The root label is null (Empty string)
  - Children of node should have different names → it guarantees the uniqueness of the domain name
- **Domain Name**
  - It's a symbolic string associated with an IP address
  - Each node has a domain name
  - It's a sequence of labels separated by dots
  - Domain names are always read from the node up to the root

- Last label is the root
- This means full domain name always ends with null label

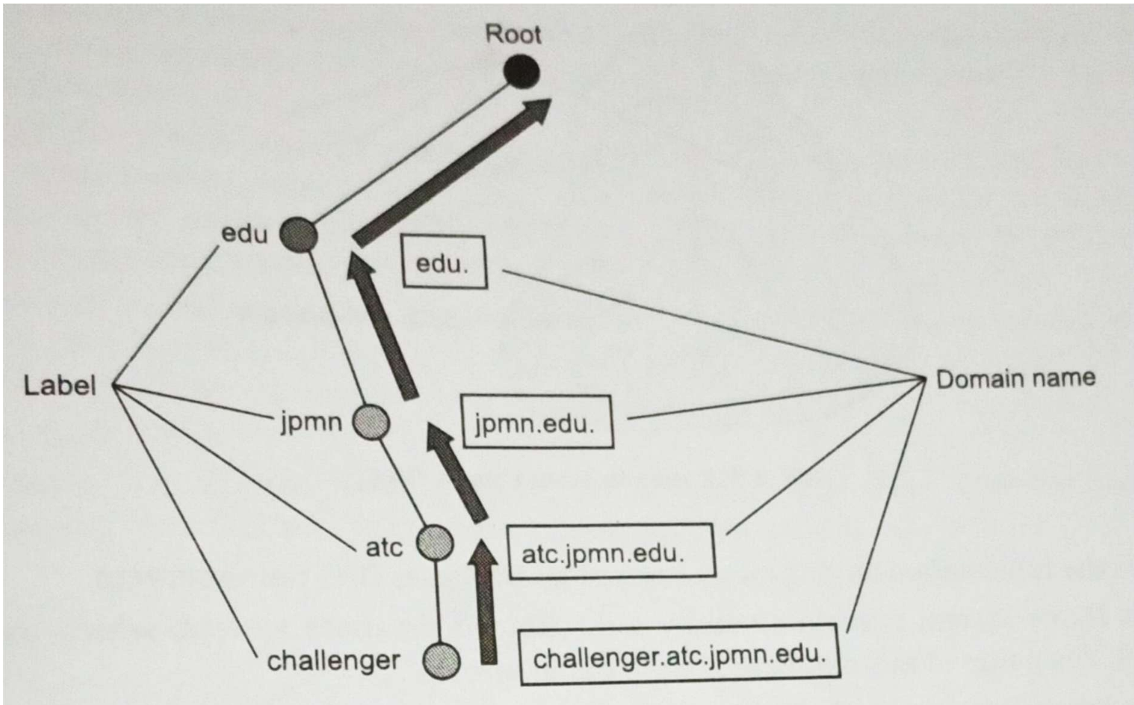


Fig. 4.5: Domain Names and Labels

The name of the domain is the nam

- **Domain:**
  - Domain is s subtree of domain name space
  - Name of the domain is the name of the node at the top of subtree
  - Domain is itself can be divided in to subdomains

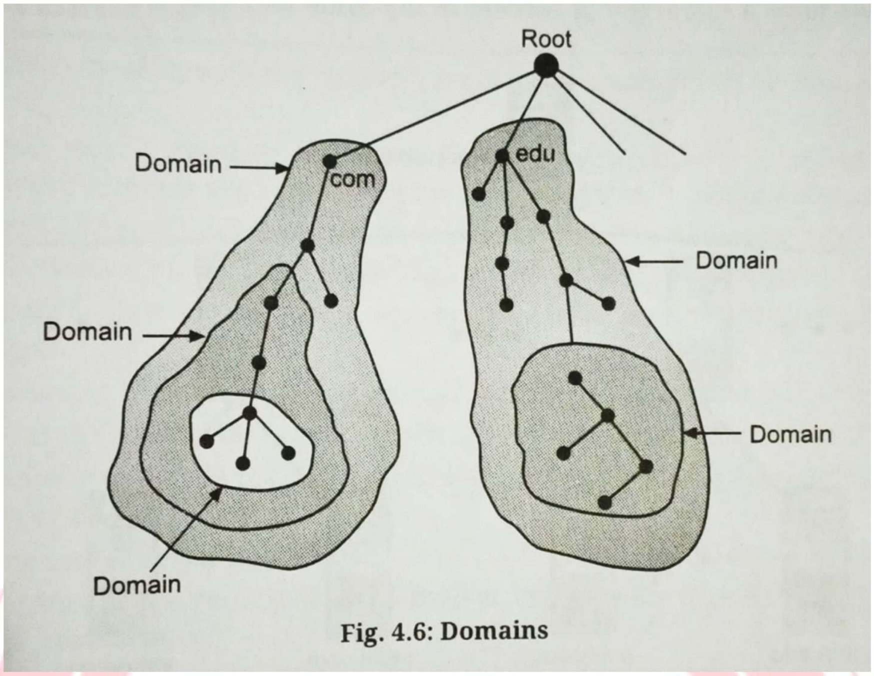


Fig. 4.6: Domains

- **Zones**
  - It's a collection od nodes under main domain
  - Server maintain database called zone file for every node
  - As complete name hierarchy cannot be stored on a single server → It is divided among many servers
  - If domain is not divided into sub domain → Domain and zone refer to the same thing
  - Information about nodes in the subdomain is stored in lower-level servers
  - Original server keeps reference to lower-level servers

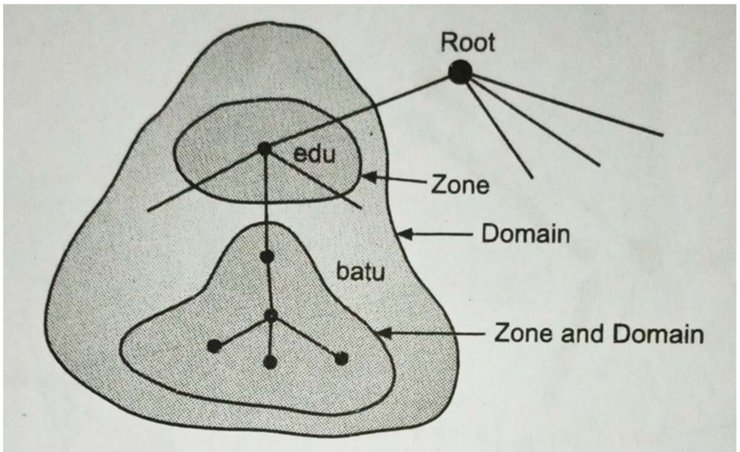


Fig. 4.7: Zones in Domains in DNS

Q: what is a name server? What are its different types?

- To distribute information among many computers DNS uses DNS name servers
- Domain name system is maintained by distributed database system
- Nodes in this system are the name servers
- Each domain has at least one DNS server → that publishes information about domain and the name servers belonging to it
- Root is standalone → can create many domains (subtrees) as first level nodes
- Created domain are very large so they are divided into subdomains
- Each and every server is responsible for either large or small domain

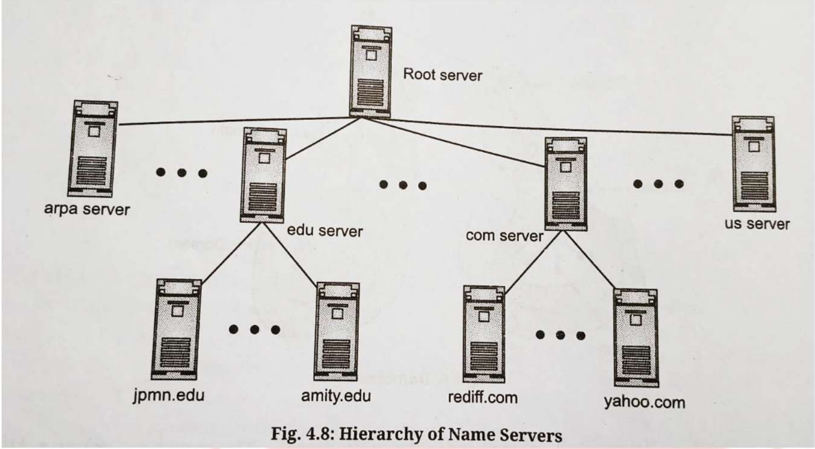
- 

Fig. 4.8: Hierarchy of Name Servers

Types of name servers

1. **Root server**
   - Usually does not store any information
   - it assigns authority to other servers
   - It keeps reference of other servers
   - There are several root servers → covering whole domain name space
   - Root servers are distributed all around the world
2. **Primary server**
   - It stores a file about the zone for which it is an authority
   - It is responsible for- creating, maintaining and updating the zone file
   - It stores zone file on a local disk
3. **Secondary server**
   - It’s a server that transfers complete information about a zone from another server →and stores the file on local disc
   - It neither creates nor updated the zone file
   - If updating is required it must be done by primary server
   - Primary server sends updated version to secondary server

Q Explain DNS architecture in detail

DNS in internet

- DNS tree is divided into three sections - Generic domains - Country domains - Inverse domain

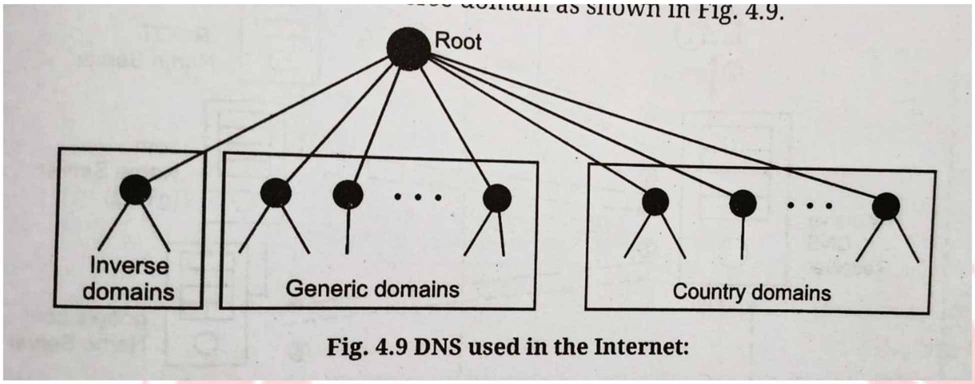


Fig. 4.9 DNS used in the Internet:

- "The process translating IP address to corresponding domain names through DNS is called Name resolution of DNS resolution"
- It begins with client's DNS request

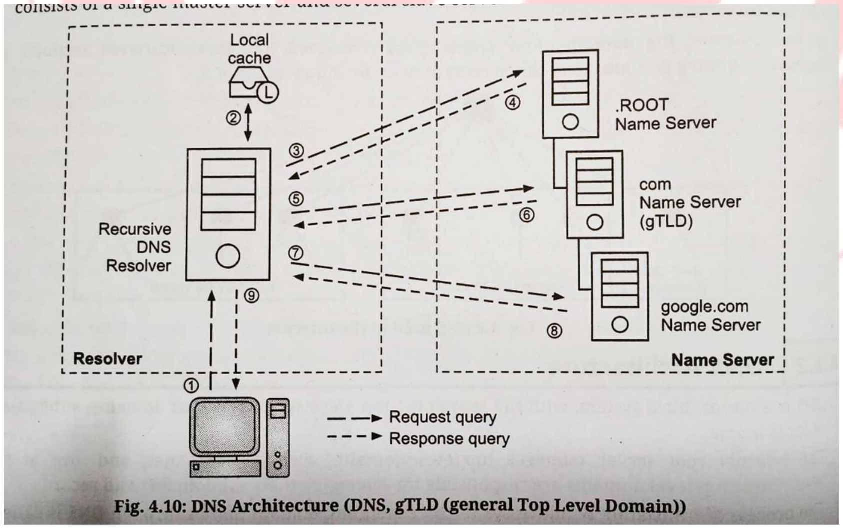
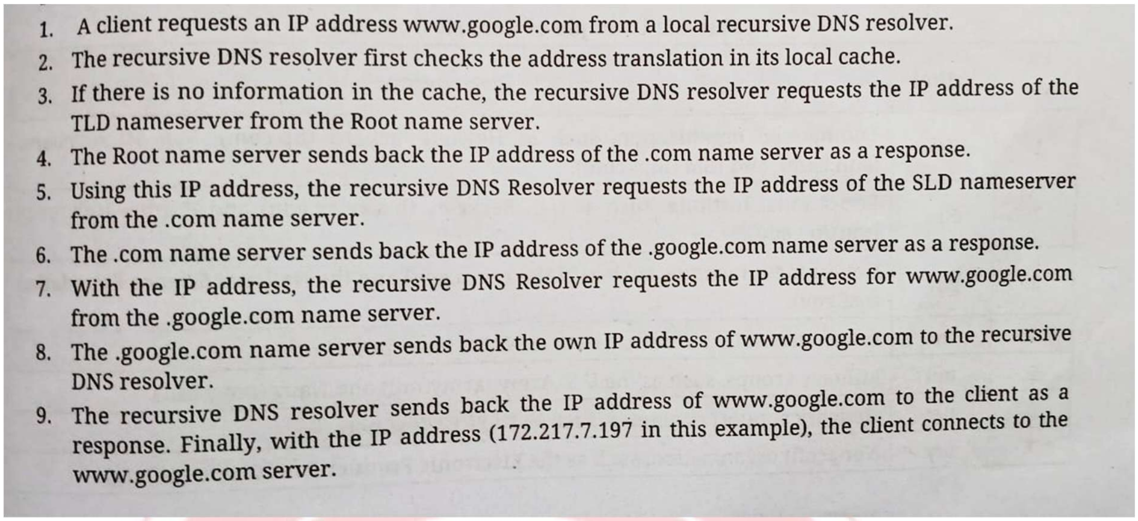


Fig. 4.10: DNS Architecture (DNS, gTLD (general Top Level Domain))

- Fig shows how client obtains IP address for a web server via DNS resolution→ allowing it to receive web services



1. A client requests an IP address www.google.com from a local recursive DNS resolver.
2. The recursive DNS resolver first checks the address translation in its local cache.
3. If there is no information in the cache, the recursive DNS resolver requests the IP address of the TLD nameserver from the Root name server.
4. The Root name server sends back the IP address of the .com name server as a response.
5. Using this IP address, the recursive DNS Resolver requests the IP address of the SLD nameserver from the .com name server.
6. The .com name server sends back the IP address of the .google.com name server as a response.
7. With the IP address, the recursive DNS Resolver requests the IP address for www.google.com from the .google.com name server.
8. The .google.com name server sends back the own IP address of www.google.com to the recursive DNS resolver.
9. The recursive DNS resolver sends back the IP address of www.google.com to the client as a response. Finally, with the IP address (172.217.7.197 in this example), the client connects to the www.google.com server.

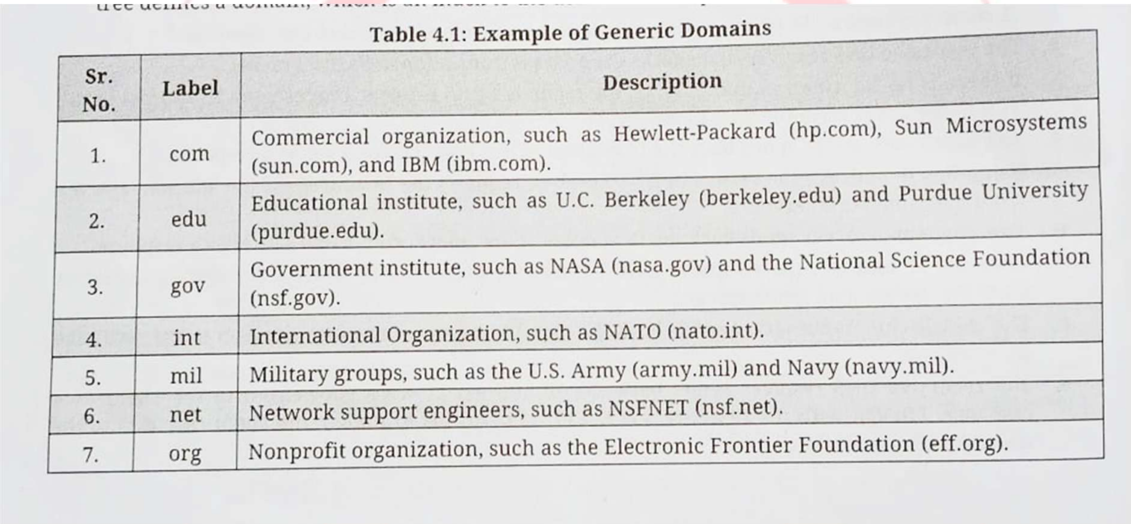
- DNS framework consists of three parts

  1. **Client**
     - They request address with domain name through a stub resolver (a client of DNS)
     - Client transmits a request to the local DNS server address set on its device
  2. **Local DNS server (recursive DNS resolver)**
     - They receive DNS query from client
     - They obtain IP address for the Domain name from domain name servers
     - The IP address once found is stored in memory for a certain period
     - So, it is called catching resolver
  3. **Domain name server (authoritative name server)**
     - They have IP addresses
     - IP addresses are managed for domain names as well as information related to IP address is also managed.
     - They are composed of three levels → root, TLD, Lower-level domain

- Each domain server consists of single master server and several slave servers

Q What are types of Domains in DNS? explain domain levels in DNS?

1. **Generic domains**
    - Generic domains define registered hosts according to their generic behaviour
    - Each node in the tree is defined as a domain
    - Node is an index to the domain name space database

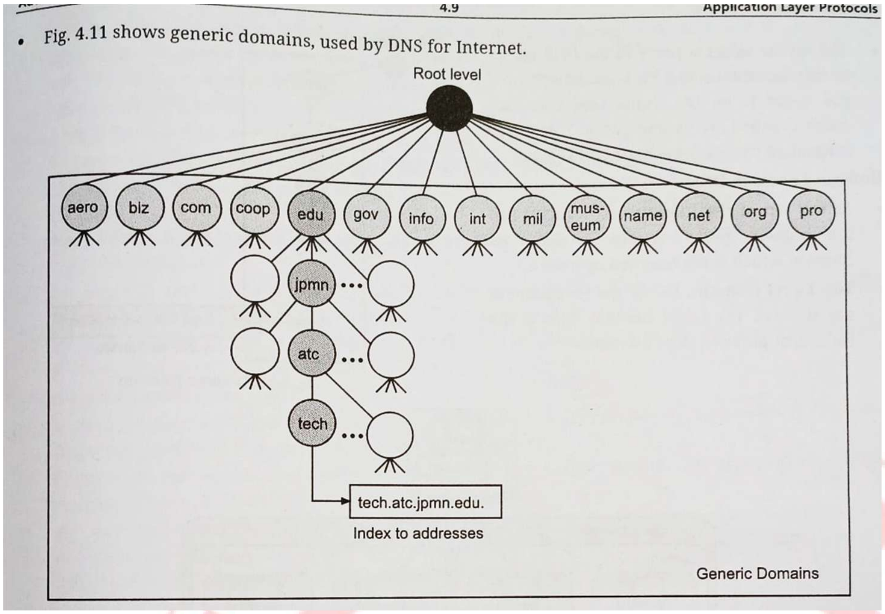


tree defines a domain, which is an index to the

Table 4.1: Example of Generic Domains

Sr. No.	Label	Description
1.	com	Commercial organization, such as Hewlett-Packard (hp.com), Sun Microsystems (sun.com), and IBM (ibm.com).
2.	edu	Educational institute, such as U.C. Berkeley (berkeley.edu) and Purdue University (purdue.edu).
3.	gov	Government institute, such as NASA (nasa.gov) and the National Science Foundation (nsf.gov).
4.	int	International Organization, such as NATO (nato.int).
5.	mil	Military groups, such as the U.S. Army (army.mil) and Navy (navy.mil).
6.	net	Network support engineers, such as NSFNET (nsf.net).
7.	org	Nonprofit organization, such as the Electronic Frontier Foundation (eff.org).

- **Fig shows generic domain used by DNS on internet**



- Fig. 4.11 shows generic domains, used by DNS for Internet.

2. Country domain:

- It follows same format as generic domains
- But it uses 2-character country abbreviations in place of 3-character organizational abbreviations at the first level

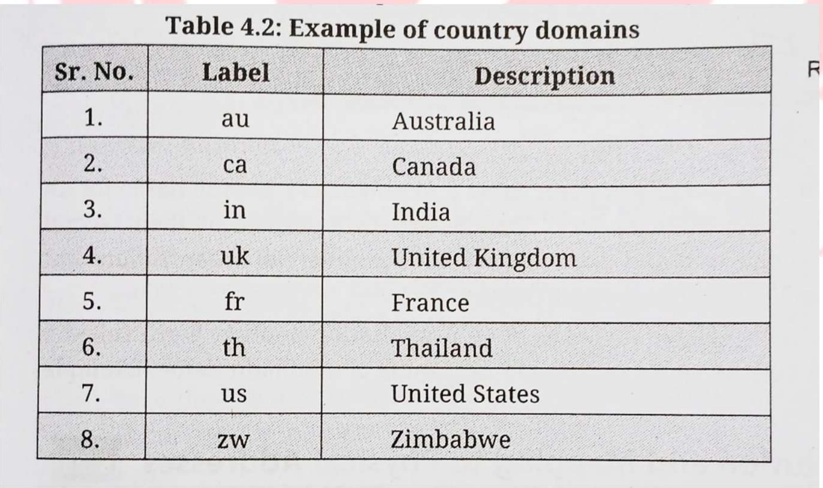


Table 4.2: Example of country domains

Sr. No.	Label	Description
1.	au	Australia
2.	ca	Canada
3.	in	India
4.	uk	United Kingdom
5.	fr	France
6.	th	Thailand
7.	us	United States
8.	zw	Zimbabwe

- Following fig shows country used by DNS for internet

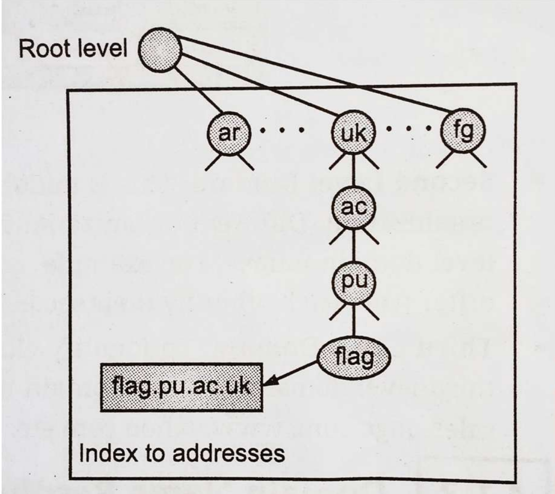


Fig. 4.12

3. **Inverse domain**
    - It is used to map address to a name
    - Client sends a request to a server for a particular task
    - Server finds a list of authorised clients
    - The list contains only IP address of client
    - Server send query to DNS server to map an address to a name → to determine if the client is in authorised list
    - This query is called inverse query
    - It is handled by first level node called ARPA

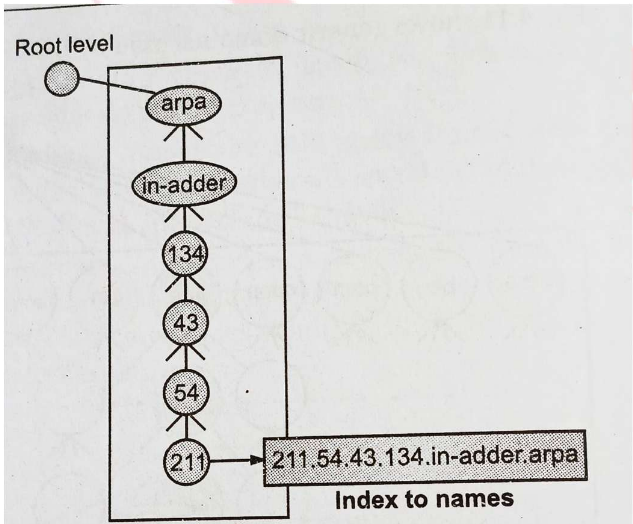


Fig. 4.13: Inverse Domain

Domain levels in DNS

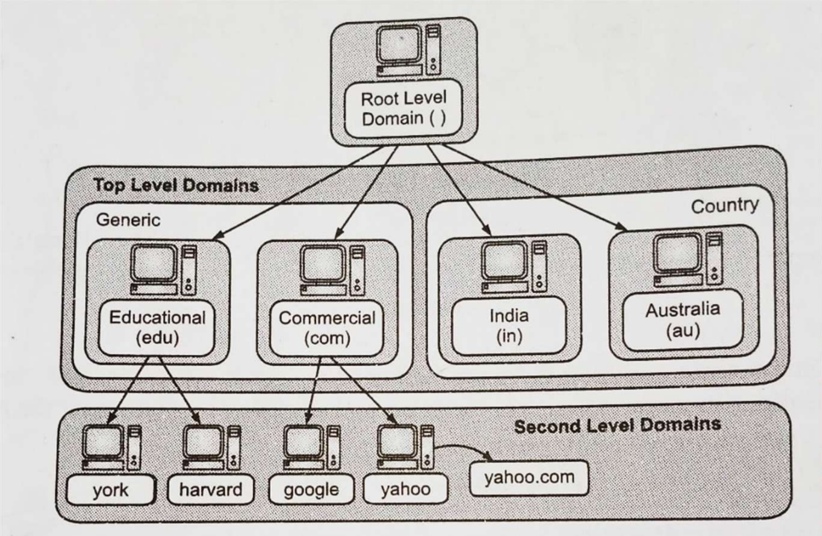


Fig. 4.14

- At the top of hierarchical structure of DNS root is situated

1. **Top level domain**
    - Top level domain is situated below the root
    - This the right most part of the domain name
2. **Second level domain**
    - This is the middle section of domain name
    - It indicates unique name for organization
    - Different organization with same top-level domains are distinguished by their second level domain names
    - E.g. Google.com, ONGC.com
    - Both are commercial organization but differ from each other by their second level domain name
3. **Third level domain**
    - To identify closely related division of an organization we use third level domains also called as subdomain
    - It is added at the beginning of the domain name
    - Eg. Sales.ongc.com, www.yahoo.com

4.1.3 Domain name resolution and mapping to physical addresses

- DOMAIN NAME RESOLUTION:" It is a process of converting a human readable domain name toa machine readable IP address
- E.g. www.google.com to 142.250.190.4
- Mapping name to address OR address to name → is called address resolution
- DNS is a client server model
- When a host needs to map address to name or name to address→it calls a DNS client (CALLED AS DNS RESOLVER)→ it further accesses closest DNS server with mapping request
- If server has the information, it informs the resolver → if it doesn't have the information it refers to another DNS server

Mapping names to addresses

- resolver gives domain name to server—asks for corresponding address
- if domain name is from generic domain server checks the generic domain section—resolver receives the respective domain (e.g. www.xyz.edu)
- query is sent by the resolver to local DNS server for resolution
- if local DNS server cannot resolve→ it refers or asks another DNS server directly
- if domain name is from country domain server checks the country domain section—resolver receives the respective domain (e.g. www.pqr.in)

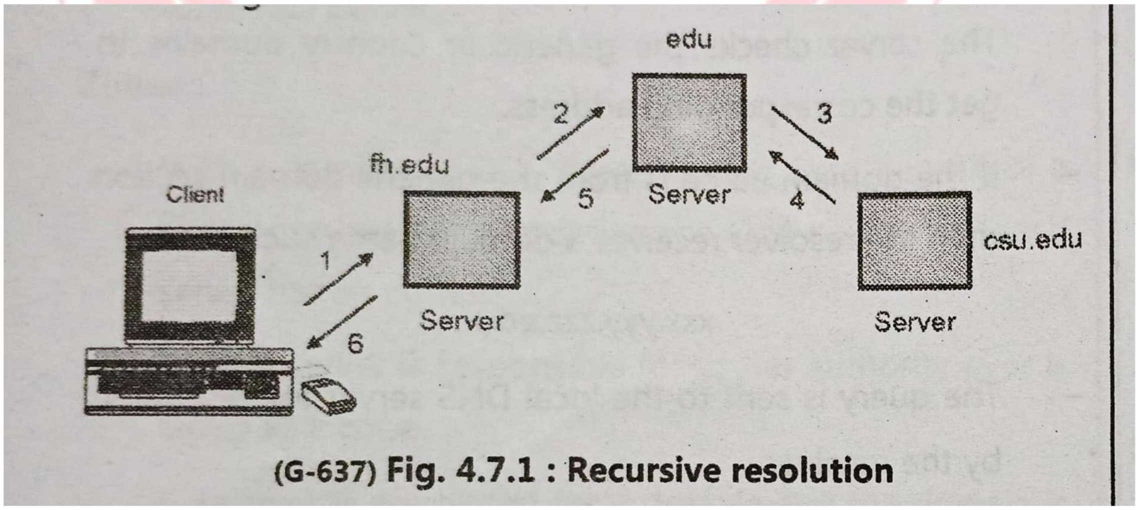
Mapping addresses to names

- client can send IP address to server—to map with domain name→ called a PTR query

- for such query DNS uses inverse domain
- in the request – IP address is reversed
- 2 labels in-addr and arpa – are appended to create a domain acceptable by inverse domain section
- E.g. if resolver receives IP address as- 132.34.45.121
- It is inversed first and added with labels as
    - → 121.45.34.132.in-addr.arpa
- It is then received by local DNS and resolved

Recursive resolution in DNS

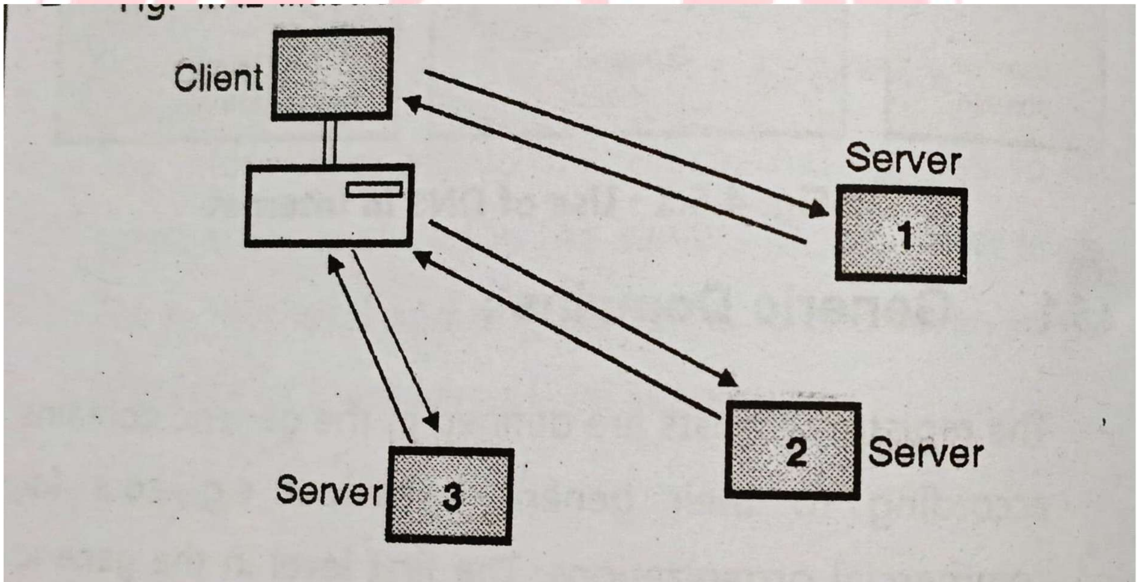
- Client can ask for recursive answer from name server
- Resolver here expects the server to supply final answer
- If server is an authority for domain names → it checks for domain names and responds
- If it is not the authority → it sends request to another server and waits for response
- When the query is finally resolved→ response travels back to requesting client



(G-637) Fig. 4.7.1 : Recursive resolution

Iterative resolution

- If client does not ask for recursive answer
- Mapping can be done iteratively
- If server is the authority for the name, it sends the answer
- If it is not it returns the IP address to the server → that it thinks can resolve the query.
- The client is responsible for repeating the query to this second server.
- If second server fails to answer the query → it returns IP address of new server to the client
- Client has to repeat the query to the third server → iterative process → query is repeated to multiple servers.



(G-638)Fig. 4.7.2 : Iterative resolution

Caching

- If every time server receives a query for a name that is not in it's domain → it needs to search its database for a server IP address

- This searching time can be reduced to increases efficiency
- DNS handles this with a mechanism called → caching
- When a server asks for a mapping from another server and receives the response → it stores this information in cache before sending it to client
- If same mapping is asked cache memory will help.
- To inform the client that→ response is coming from cache and not from authoritative source → server marks the response as unauthoritative
- **Operation of DNS**

Fig. 4.15 shows operation of DNS.

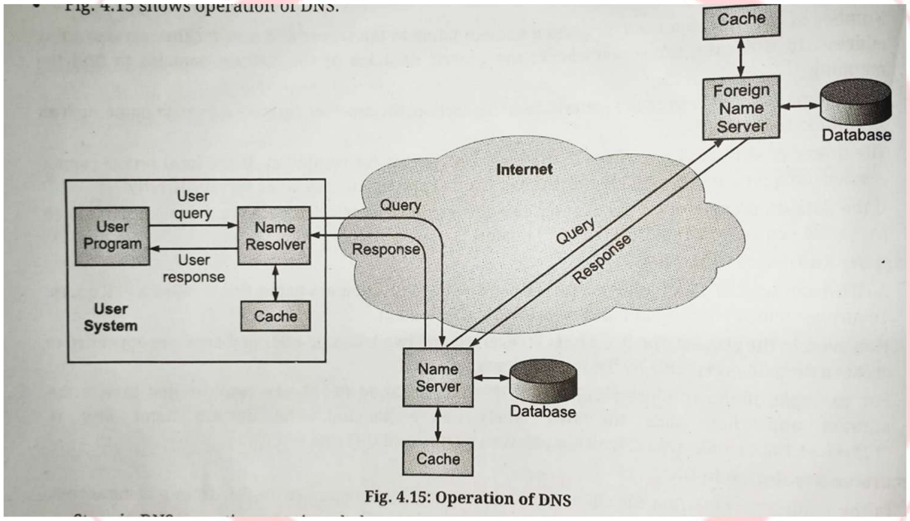


Fig. 4.15: Operation of DNS

Steps in DNS operation

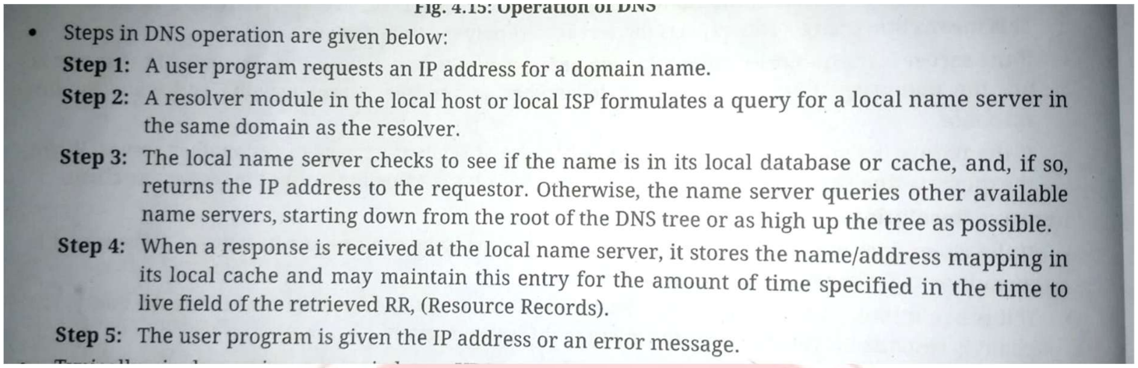


Fig. 4.15: Operation of DNS

- Steps in DNS operation are given below:

Step 1: A user program requests an IP address for a domain name.

Step 2: A resolver module in the local host or local ISP formulates a query for a local name server in the same domain as the resolver.

Step 3: The local name server checks to see if the name is in its local database or cache, and, if so, returns the IP address to the requestor. Otherwise, the name server queries other available name servers, starting down from the root of the DNS tree or as high up the tree as possible.

Step 4: When a response is received at the local name server, it stores the name/address mapping in its local cache and may maintain this entry for the amount of time specified in the time to live field of the retrieved RR, (Resource Records).

Step 5: The user program is given the IP address or an error message.

Example of DNS

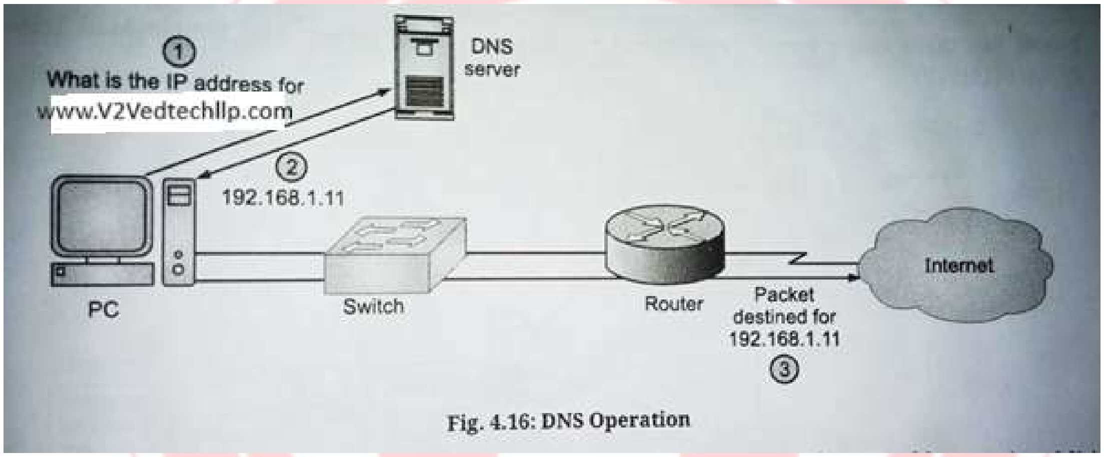

1
What is the IP address for www.V2Vedtechllp.com
DNS server
2
192.168.1.11
PC
Switch
Router
Packet destined for 192.168.1.11
3
Internet

Fig. 4.16: DNS Operation

Mapping Domain names to physical addresses

1. IP Address as Network Identifier

- Resolved IP address identifies specific device or server on a network

2. Address Resolution protocol (ARP)

- Within local network → ARP is used to map physical address od a device
- This process ensures data packets are delivered to correct hardware

3. Routing Across Networks

- Once IP address is resolved → router uses it to forward data packets
- It involves translating between IP address to MAC address

4.2. Electronic mail

Email has three major components:

1. User Agents,

2. Mail Servers and

3. Simple Mail Transfer Protocol (SMTP)

ocol (SMTP).

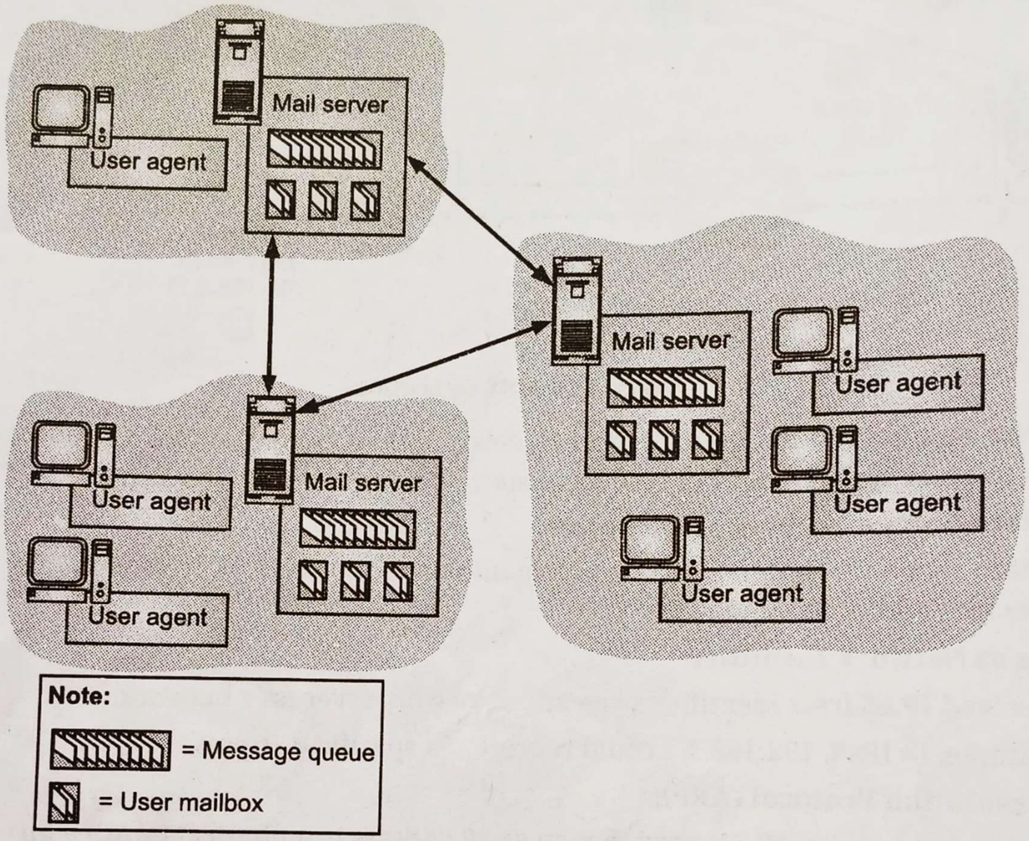


Fig. 4.17: High Level View of the Internet E-Mail System

Functions of E-mail

E-mail system supports the following **five** basic functions:

1. **Composition**: Composition refers to the process of creating messages and answers. Any text editor can be used. To the top of the message, when answering a message, the e-mail system can extract the originator's address from the incoming e-mail.
2. **Transfer**: Transfer refers to moving messages from the originator to the recipient.
3. **Reporting**: Reporting has to do with telling the originator what happened to the message. Whether e-mail is delivered or not delivered.
4. **Displaying**: Displaying incoming messages is needed, so user can read their e-mail.
5. **Disposition:** Disposition is the final step and concerns what the recipient does with the message after receiving it. Finally read and save or delete or forward the message.

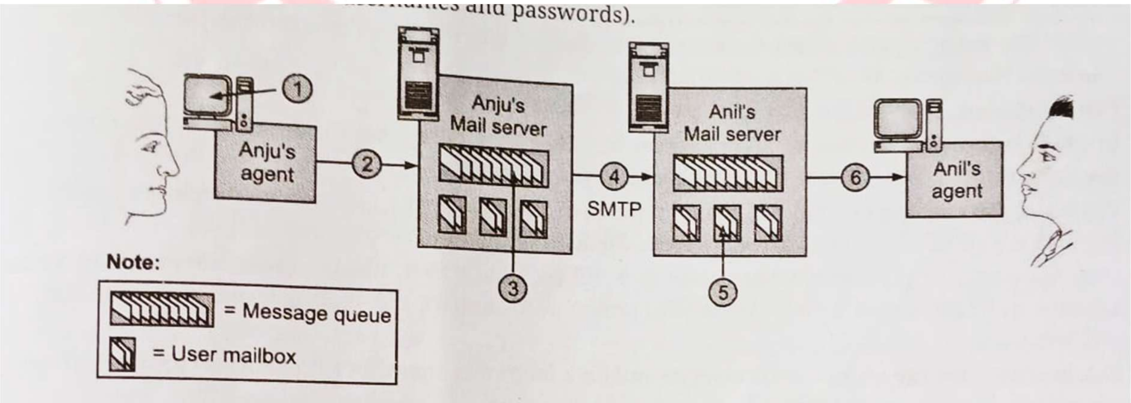


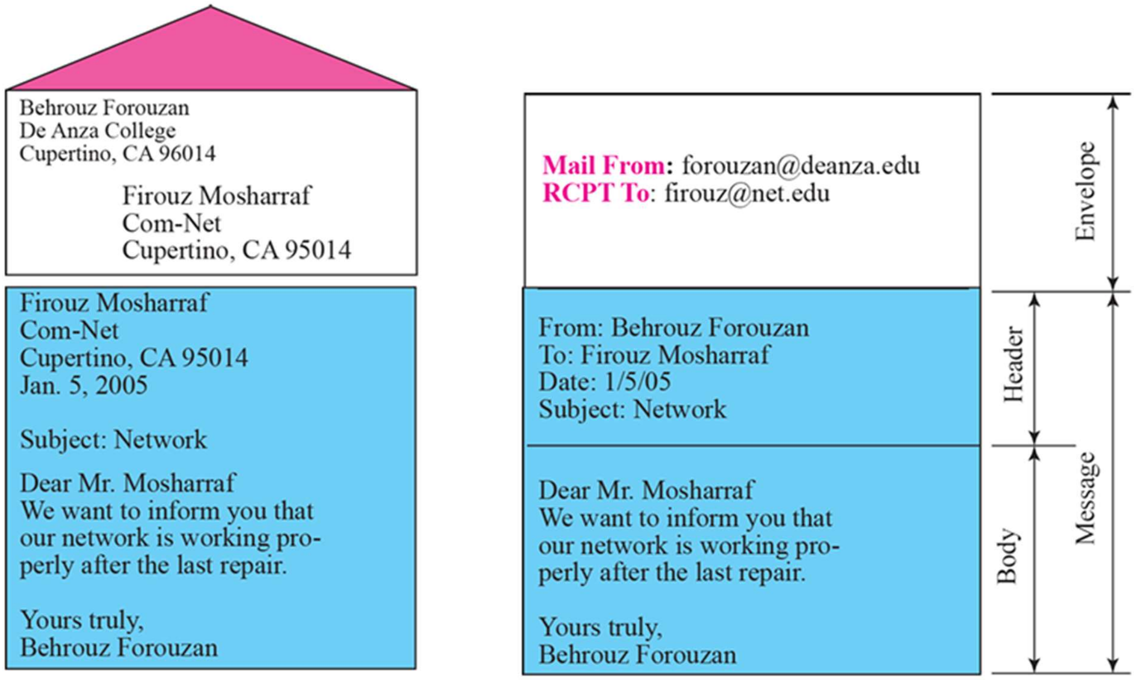
Fig. 4.18: Anju Sends a Message to Anil

 Example: Fig. 4.18 shows an example of e-mail.

- When Anju finishes composing her message, her user agent sends it to her mail server, where it’s placed in the outgoing message queue.
- When Anil wants to read a message, his user agent retrieves it from his mailbox on his mail server.
- Mail servers form the core of the e-mail infrastructure.
- Each recipient, like Anil, has a mailbox located on a mail server. This mailbox stores and manages messages sent to them.
- A typical message journey: starts in the sender’s user agent, goes to the sender’s mail server, then to the recipient’s mail server, and finally lands in the recipient’s mailbox
- When Anil wants to access the messages in his mailbox, the mail server containing his mailbox authenticates Anil (With Username and Password)
- Anju's mail server must also deal with failures in Anil's mail server. If Anju's server cannot deliver mail to Anil's server, Anju's server holds the message in a message queue and attempts to transfer the message later.
- Reattempts are often done every 30 minutes or so; if there is no success after several days, the server removes the message and notifies the sender (Anju) with an e-mail message.
- SMTP is the principal application layer protocol for Internet electronic mail. It uses the reliable data transfer service of TCP to transfer mail from the sender's mail server to the recipient's mail server.
- SMTP has namely, a client side, which executes on the sender's mail server, and a server side, which executes on the recipient's mail server. Both the client and server sides of SMTP run on every mail server.

- When a mail server sends mail to other mail servers, it acts as an SMTP client. When a mail server receives mail from other mail servers, it acts as an SMTP server.

Format of email (Email envelope)



Behrouz Forouzan
De Anza College
Cupertino, CA 96014

Firouz Mosharraf
Com-Net
Cupertino, CA 95014

Firouz Mosharraf
Com-Net
Cupertino, CA 95014
Jan. 5, 2005

Subject: Network

Dear Mr. Mosharraf
We want to inform you that our network is working properly after the last repair.

Yours truly,
Behrouz Forouzan

Mail From: forouzan@deanza.edu
RCPT To: firouz@net.edu

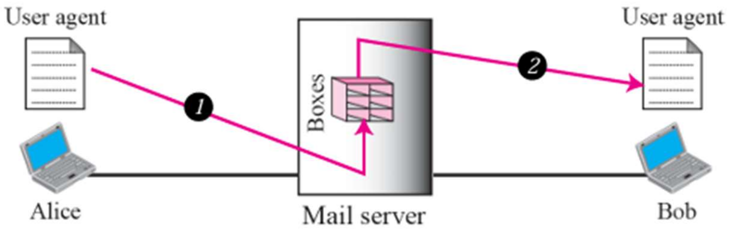
From: Behrouz Forouzan
To: Firouz Mosharraf
Date: 1/5/05
Subject: Network

Dear Mr. Mosharraf
We want to inform you that our network is working properly after the last repair.

Yours truly,
Behrouz Forouzan

Email architecture and services – 4 scenarios

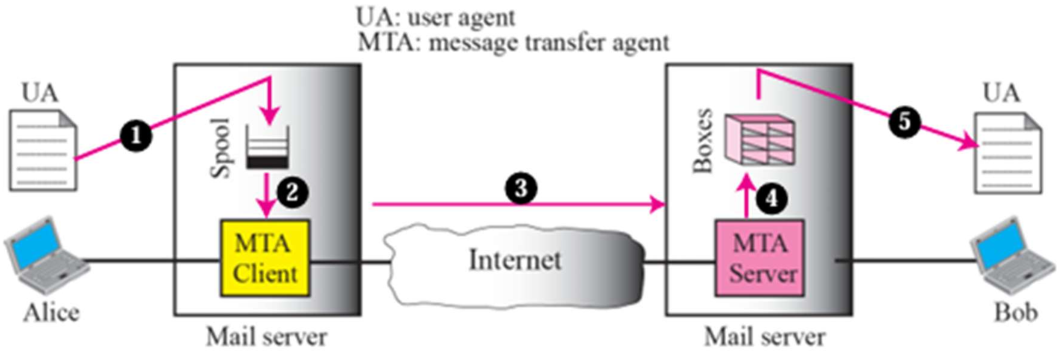
1. First scenario



Scenario 1: Both Users on the Same System

- **Setup:** Sender and recipient use the same computer or local system.
- **Flow:**
    - The sender composes the email using a User Agent (UA).
    - The message is stored directly in the recipient's mailbox on the same system.
- No need for a mail server or internet connection—this is a local delivery.

2. Second scenario



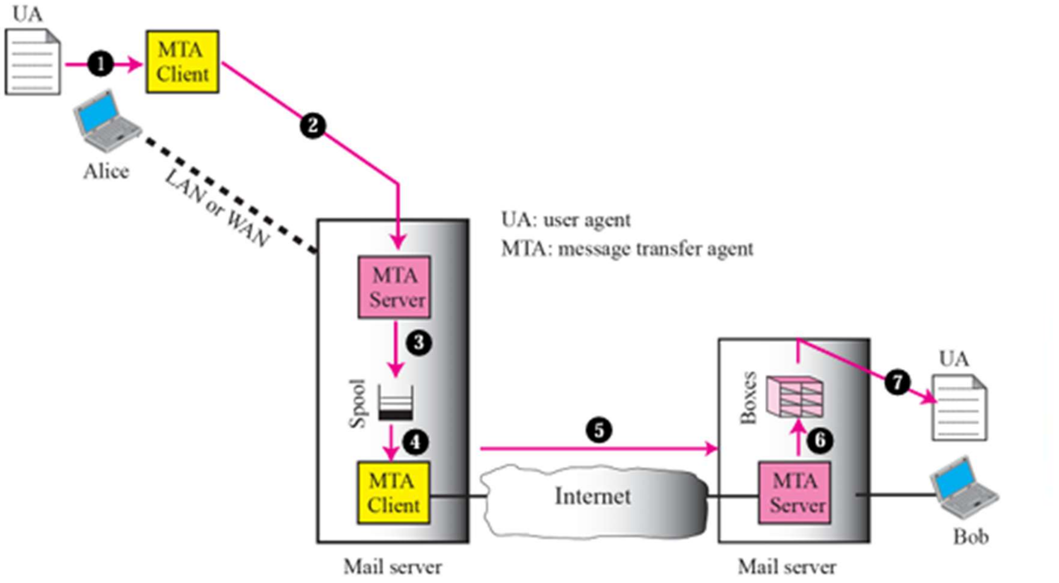
Scenario 2: Users on Different Systems

- **Setup:** Sender and recipient are on separate computers.
- **Flow:**
    - The sender's UA sends the message to their Mail Transfer Agent (MTA).
    - The MTA uses SMTP to send the message to the recipient's MTA.

  - The recipient retrieves the message using POP3 or IMAP via their UA.

- **Internet connection is required for communication between MTAs.**

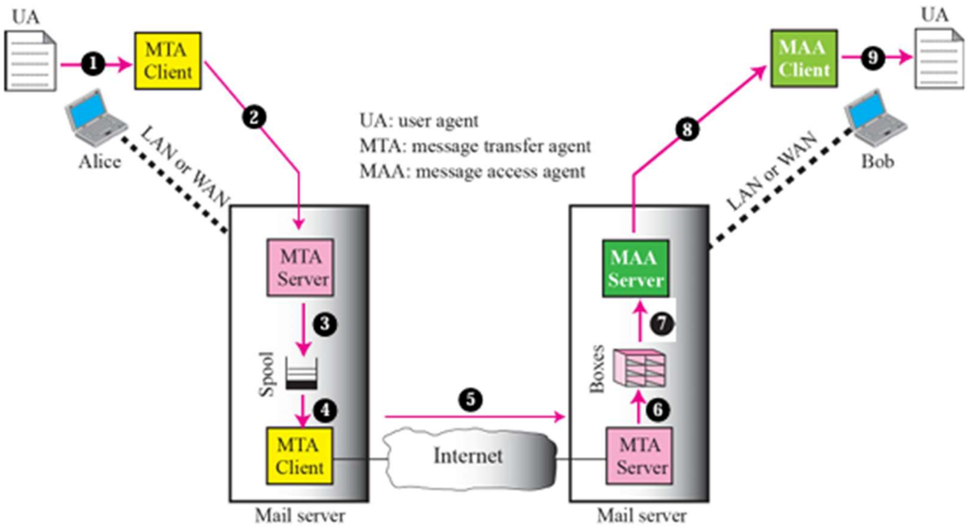
3. Third scenario



Scenario 3: LAN/WAN Connection

- **Setup:** Sender and recipient are on different systems connected via a Local Area Network (LAN) or Wide Area Network (WAN).
- **Flow:**
  - Similar to Scenario 2, but the MTAs may be within the same organization or network.
  - Faster delivery due to internal routing.
  - Often uses internal mail servers for routing and storage.

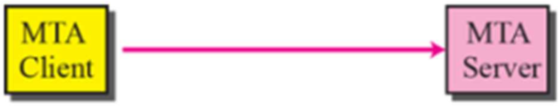
4. Fourth scenario



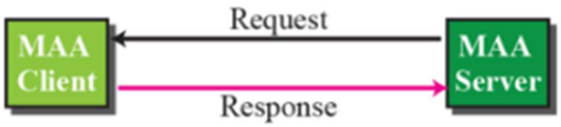
Scenario 4: Cloud-Based or Webmail Services

- **Setup:** Users use services like Gmail, Outlook, or Yahoo Mail.
- **Flow:**
  - The sender accesses the email service via a web interface or app.
  - The cloud-based mail server handles SMTP, POP3, and IMAP protocols.
  - The recipient accesses their mailbox through the same or a different service.
- Highly scalable, supports attachments, filtering, and spam protection.

Push versus pull:-



a. Client pushes messages



b. Client pulls messages

Some important terms:

- **User Agents (UA):** enable user to read and send e-mail
- **Message Transfer Agents (MTA):** move the messages from the sender to receiver

Advantages of using E-mail

1. **Speed**: The recipient can receive an e-mail within a very short time of sending the same.
2. **Cost**: No cost is involved in sending an e-mail message.
3. **Content**: A message can consist of a few lines or several lines and include any form of data like text, image, audio, video, etc. The message can be sent to several recipients simultaneously.
4. **Delivery**: A message can be delivered to the recipient anywhere in the world almost simultaneously.

5. **Reliability**: E-mail is a reliable mode of communication, as in case a message cannot be sent, the sender is notified with a failure message.

6. **Security**: Unauthorized persons cannot access your e-mail. Only the user with his password can open his account or mailbox and view his mails.

7. **Access**: In case of a Web mail service, a user can check his mails from any computer/mobile around the world, provided it is connected to the Internet.

Disadvantages of using E-mail

1. **Information Overload**: People daily receive many mails, majority of which can be junk mails. One can have to go through hundreds of mails to find out which mails are useful and which are not. Thus, spamming can be unproductive.

2. **Security Risk**: E-mails can be a source of computer viruses. Viruses can come as attachments to mails from unknown sources. When such mails are opened, the computers get infected by the virus. Using antivirus programs and firewalls can prevent such infection.

3. **E-mail Bombing**: It is the intentional sending of a large number of e-mails to a particular target address. This may cause network delay and can finally lead to a server crash.

4. **E-mail Spoofing**: This occurs when the e-mail header is altered so that the message appears to have originated from a different source. This is usually done to collect personal information of the recipient such as usernames, passwords, bank account and

credit card details, which can then be misused. As stated earlier, this is called Phishing (pronounced fishing).

5. **Distraction**: Checking and replying to vast number of e-mails can hamper productivity.

4.2.2 Message Transfer Agents (MTA)

Working of MTA

1. **Mail User Agent (MUA)** Software used by the user to compose, send, and read emails.
2. **Mail Submission Agent (MSA)** Receives email from the MUA and forwards it to the MTA after validation.
3. **Main Transfer Agent (MTA)** Transfers email between servers using SMTP across the internet.
4. **Message Delivery Agent (MDA)** Delivers the email to the recipient's mailbox for retrieval.

Functions of Message Transfer Agents (MTA)

1. Accepts emails from the Mail User Agent (MUA).
2. Selects the recipient's mail server using MX records and domain name.
3. Processes deferrals and tracks delivery status.
4. Sends auto-responses

Flow Summary: MUA → MSA → MTA → MDA → Recipient's MUA

Types of MTA Servers

1. **On- Premise MTA servers**

   - Hosted within an organization's infrastructure.
   - Offers full control over email systems.

- Requires investment in hardware and software.
- Stores data in an indexed internal database.

2. **Cloud based SMTP servers**

- Hosted by third-party providers (e.g., SendGrid, Mailgun).
- More cost-effective and scalable.
- Limited control over infrastructure and delivery mechanisms.

SMTP (Simple Mail Transfer Protocol)

- SMTP is an Internet standard for transmitting email across IP networks.
- It is a connection-oriented, text-based protocol.
- Communication occurs via commands and data streams over a TCP connection.
- SMTP is used for sending and receiving emails over TCP/IP architecture.

- Email systems rely on Message Transfer Agents (MTAs) to implement SMTP.
- MTAs handle sending, storing, and receiving emails on mail servers.
- SMTP defines the formal protocol for client-server communication between MTAs.
- It ensures reliable email transmission across the Internet.

Characteristics of SMTP

1. SMTP is a push protocol and uses port number 24.
2. It operates over TCP at the transport layer.
3. Uses persistent TCP connections, allowing multiple emails to be sent together.

4. It is a connection-oriented protocol.
5. SMTP is an in-band protocol.
6. It is a stateless protocol.

- SMTP is described as a two-band protocol:

  Between the sender and sender's mail server & Between the sender's mail server and receiver's mail server

- A different protocol is used between the receiver's mail server and the receiver (typically POP3 or IMAP).
- SMTP is a simple ASCII-based protocol that establishes a TCP connection between the sender and port 24 of the receiver.
- No checksums are required because TCP ensures reliable delivery.
- Once all emails are exchanged, the connection is released.

nail, the connection is released.

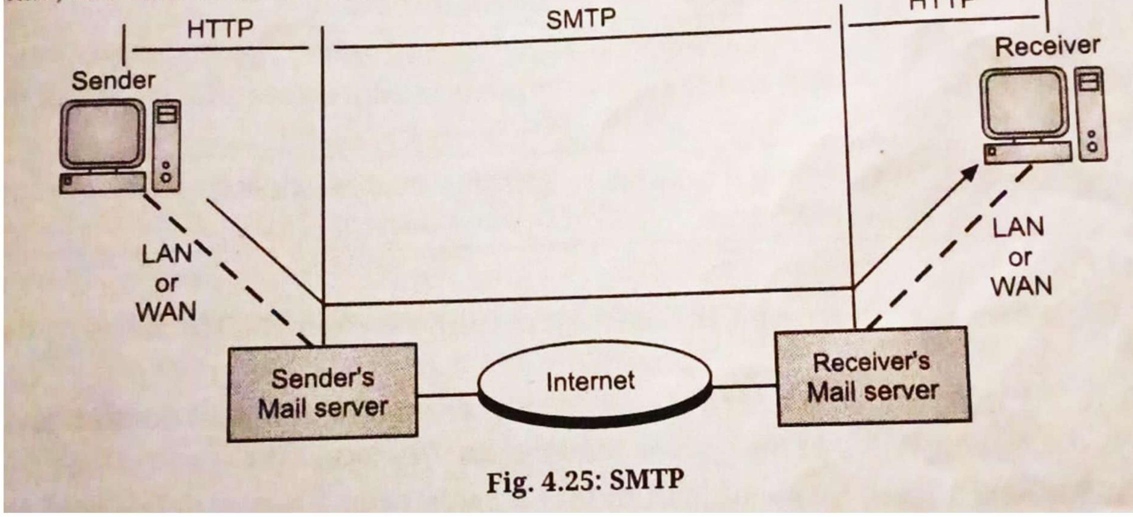


Fig. 4.25: SMTP

Commands and Responses

Most useful SMTP Commands

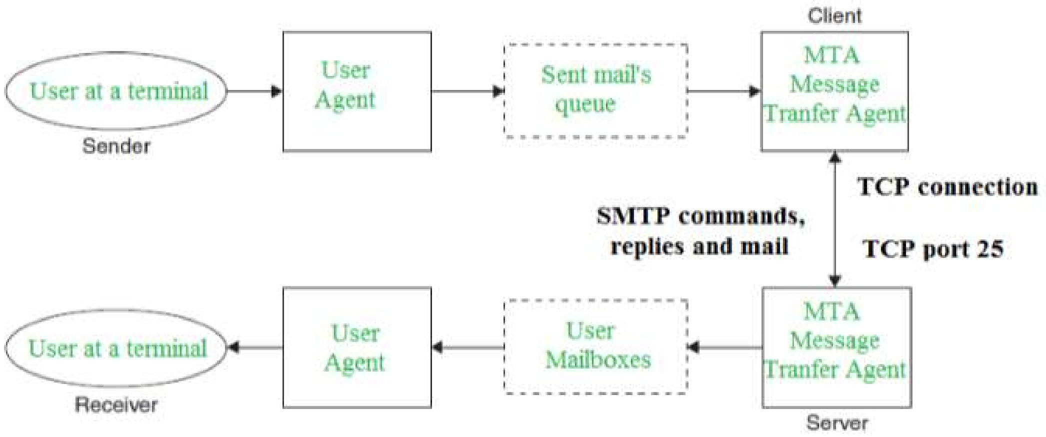
Here's a list of the most useful SMTP commands that form the backbone of email transmission between servers:

Essential SMTP Commands

Command	Purpose
HELO	Initiates conversation with the server, identifying the sender's domain
MAIL FROM	Specifies the sender's email address
RCPT TO	Specifies the recipient's email address
DATA	Begins the transfer of the email content (headers + body)
RSET	Resets the current session
VRFY	Verifies if an email address exists on the server
NOOP	Sends a no-operation command; used to keep the connection alive
EXPN	Expands a mailing list to show all recipients (often disabled)
HELP	Requests help or a list of supported commands
QUIT	Ends the SMTP session gracefully

Model of SMTP

- The user interacts with a User Agent (UA) like Microsoft Outlook, Netscape, or Mozilla to send emails.
- The Message Transfer Agent (MTA) is responsible for exchanging mail over a TCP connection, typically on port 25.
- Users do not directly deal with the MTA; it's managed by the system administrator.
- The MTA maintains a mail queue to retry delivery if the recipient is temporarily unavailable.
- Once delivered, the MTA stores the email in the recipient's mailbox.
- The recipient accesses the email using their User Agent (UA) at their terminal.



Components of SMTP

1. Mail User Agent (MUA)

- Interface used by the user to compose, send, and read emails.

- Examples: Gmail, Outlook, Thunderbird.

2. Mail Submission Agent (MSA)

- Accepts email from the MUA.
- Validates and forwards it to the MTA for delivery.

3. Mail Transfer Agent (MTA)

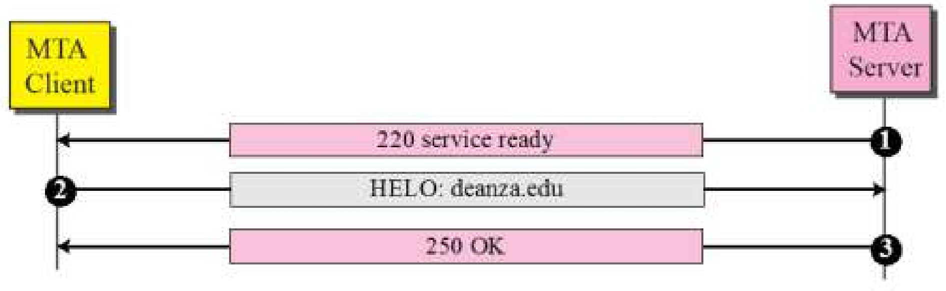
- Transfers email between servers using SMTP.
- Handles routing, queuing, and retries.

4. Message Delivery Agent (MDA)

- Delivers the email to the recipient’s mailbox.
- Works with POP3 or IMAP for message retrieval.

Working or operation of SMTP

Step 1: connection establishment



Server Greeting

- The MTA Server sends a message: 220 service ready This indicates that the server is ready to begin an SMTP session.

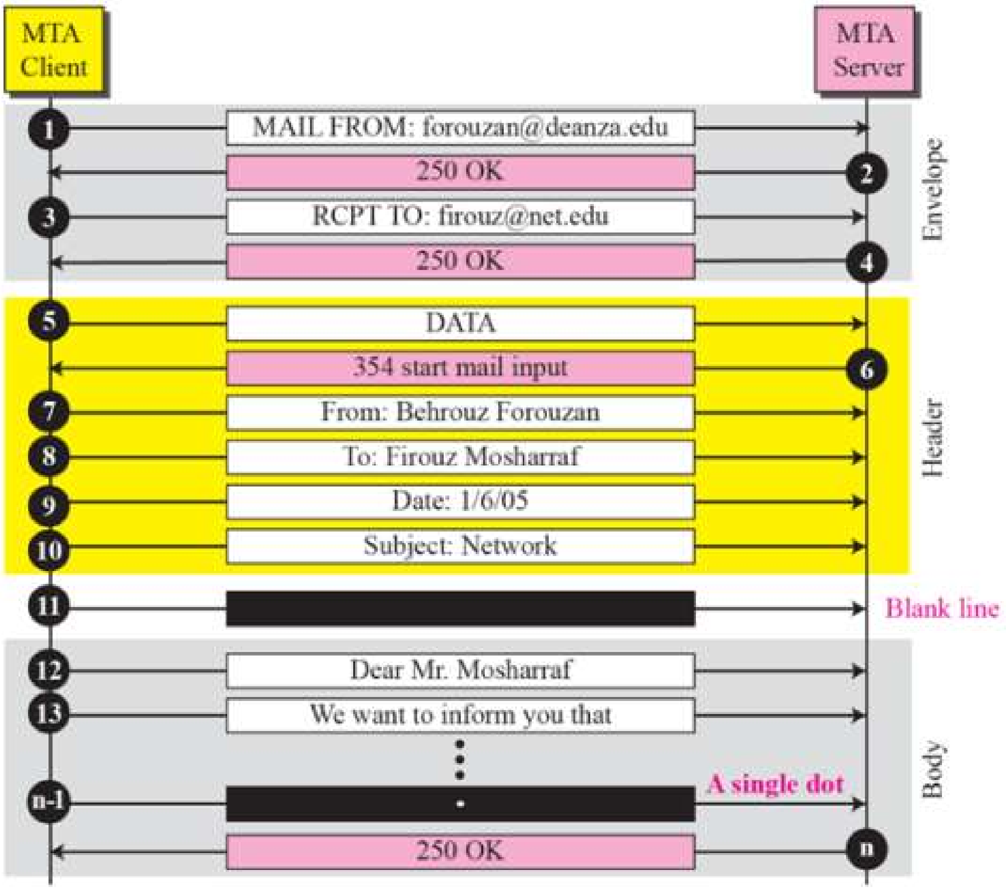
Client Introduction

- The MTA Client responds with: HELO: deanza.edu This command identifies the client to the server using its domain name (deanza.edu).

Server Acknowledgment

- The MTA Server replies: 250 OK This confirms that the server has accepted the client's identity and is ready to proceed with further commands (like MAIL FROM, RCPT TO, etc.).

Step 2: Message Transfer



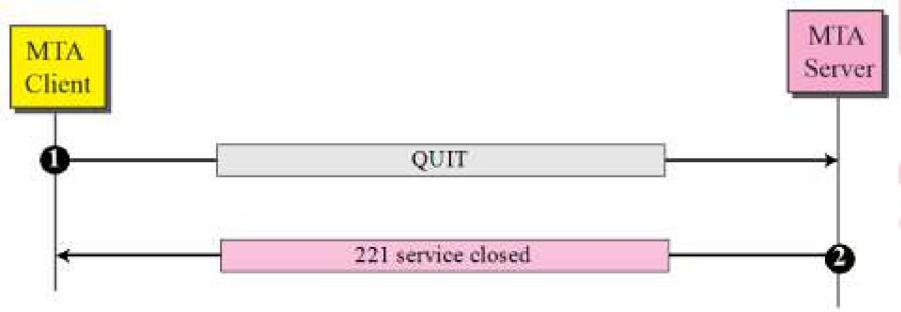
\

1. **Envelope Section**
   - **MAIL FROM:** specifies the sender’s email address.
   - **RCPT TO:** specifies the recipient’s email address.

- Server responds with 250 OK to confirm each command.

2. **Header Section**
   - Begins with the DATA command.
   - Server replies with 354 Start mail input.
   - Includes fields like From, To, Date, and Subject.
3. **Blank Line**
   - Separates the header from the body of the email.
4. **Body Section**
   - Contains the actual message content.
   - Ends with a single dot (.) on a new line to signal completion.
   - Server replies with 250 OK to confirm successful delivery.

Step 3: Connection Termination or Closing



SMTP Connection Termination

1. **QUIT Command**
   - The MTA Client sends the QUIT command to the MTA Server.

- This signals that the client has finished sending emails and wants to close the session.

2. Server Response

- The MTA Server replies with 221 service closed.
- This confirms that the server is ending the connection and releasing resources.

Advantages of SMTP

1. SMTP is a simple, text-based protocol that allows specifying multiple recipients and supports message text along with encoded objects.
2. It is easy to implement and offers high speed in email transmission.

Disadvantages of SMTP

1. SMTP cannot transmit executable files or binary objects directly.
2. Some SMTP servers may reject messages that exceed a certain size limit.
3. SMTP gateways that convert between ASCII and EBCDIC may use inconsistent code page mappings, leading to translation issues.

4.2.4 Message Access Agent (MAA)

- MAA is used to retrieve messages from the recipient's mailbox.
- It transfers the received message to the recipient's mail server.
- MAA is involved in the final delivery of the email to the recipient.

- SMTP is the Message Transfer Agent (MTA) used for sending mail.
- POP and IMAP are the Message Access Agents (MAA) used for receiving mail.

- **POP-Post Office Protocol**
  - POP is an application-layer protocol used by email clients to retrieve emails from a mail server.
  - POP3 is the current standard version.
  - POP3 supports two modes:
    - **Delete mode:** Emails are removed from the server after download.
    - **Keep mode:** Emails remain on the server after download.
  - POP3 allows users to download emails to a local computer and read them offline.
  - Client POP3 software is installed on the recipient's device.
  - Server POP3 software runs on the mail server.
  - Communication starts when the client connects to the server via TCP port 110.

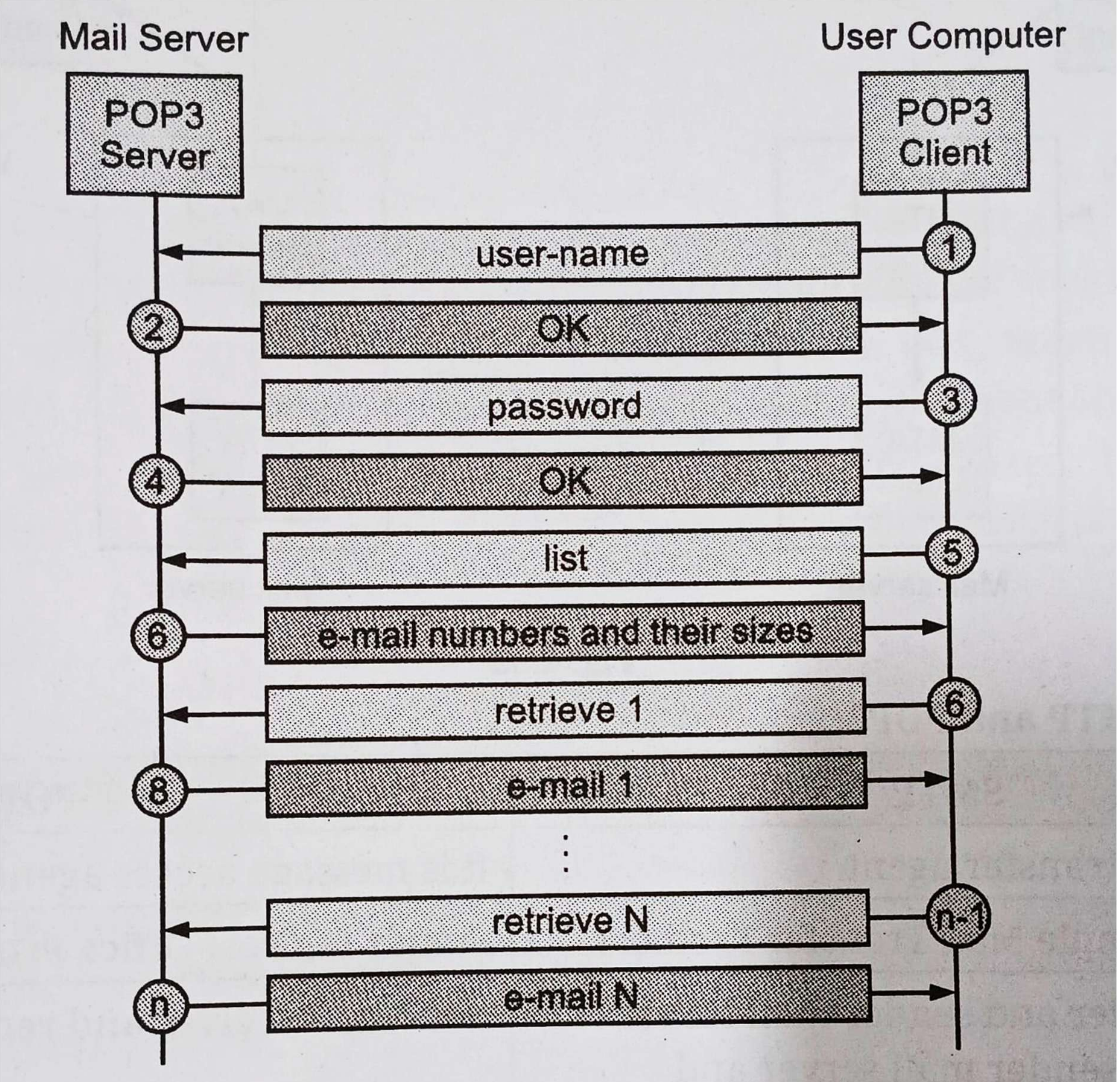
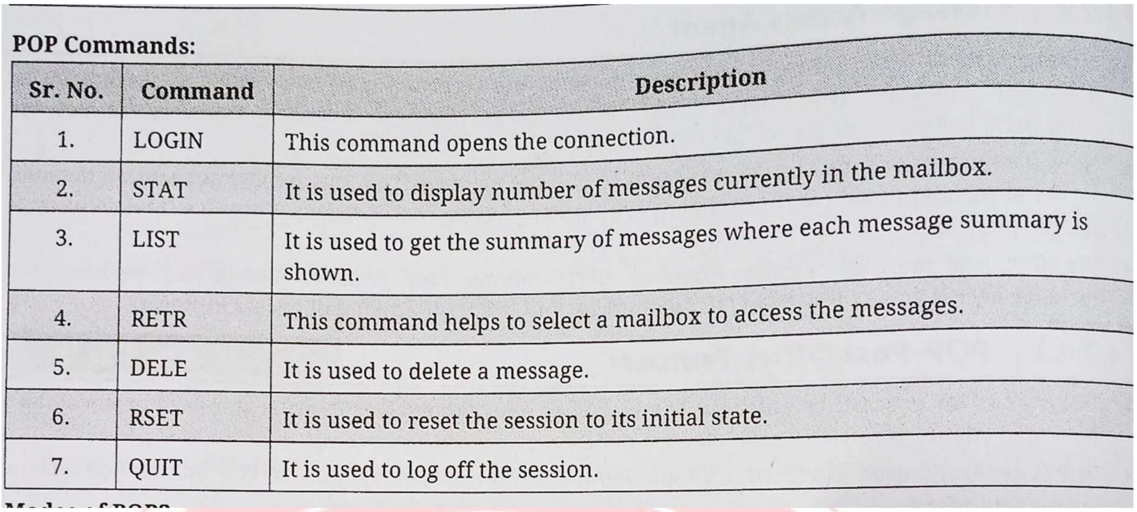


Fig. 4.31: POP3

- **POP Commands**



POP Commands:

Sr. No.	Command	Description
1.	LOGIN	This command opens the connection.
2.	STAT	It is used to display number of messages currently in the mailbox.
3.	LIST	It is used to get the summary of messages where each message summary is shown.
4.	RETR	This command helps to select a mailbox to access the messages.
5.	DELE	It is used to delete a message.
6.	RSET	It is used to reset the session to its initial state.
7.	QUIT	It is used to log off the session.

- **Modes of POP3**
- **POP3 has two modes namely, the delete mode and keep mode**
- **Delete Mode**
  - Emails are deleted from the mailbox after retrieval.
  - Best for users accessing mail from a permanent computer.
  - Allows saving and organizing emails locally after reading or replying.
- **Keep Mode**
  - Emails are retained on the server even after being read.
  - Ideal for users accessing mail from a non-permanent device (like a laptop).
  - Enables later retrieval and organization from different locations.

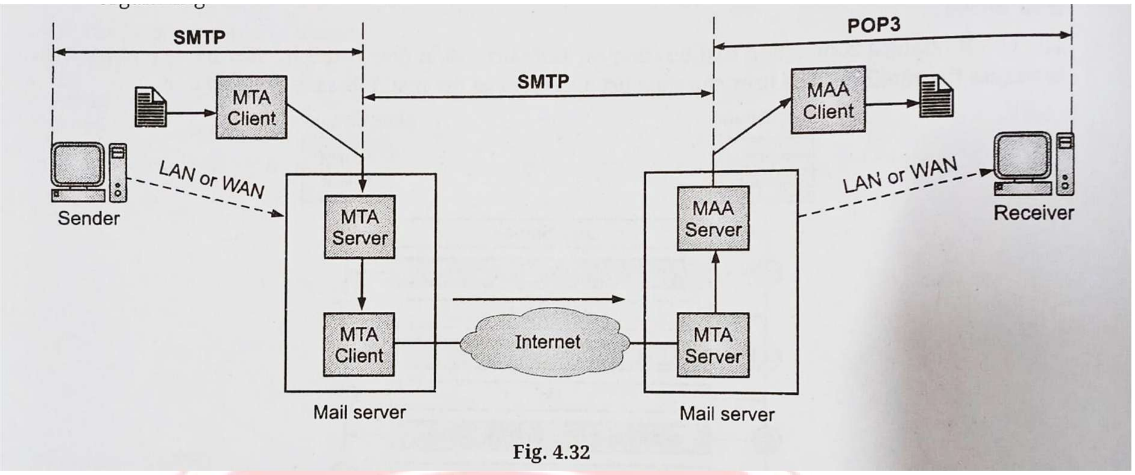


Fig. 4.32

- **Difference between SMTP and POP3**

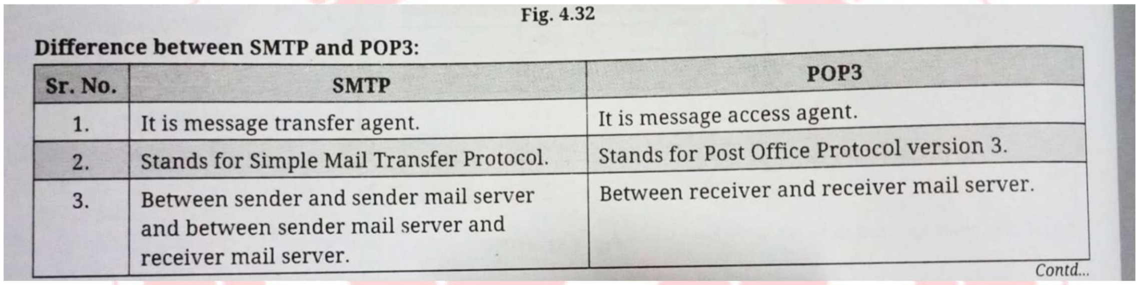
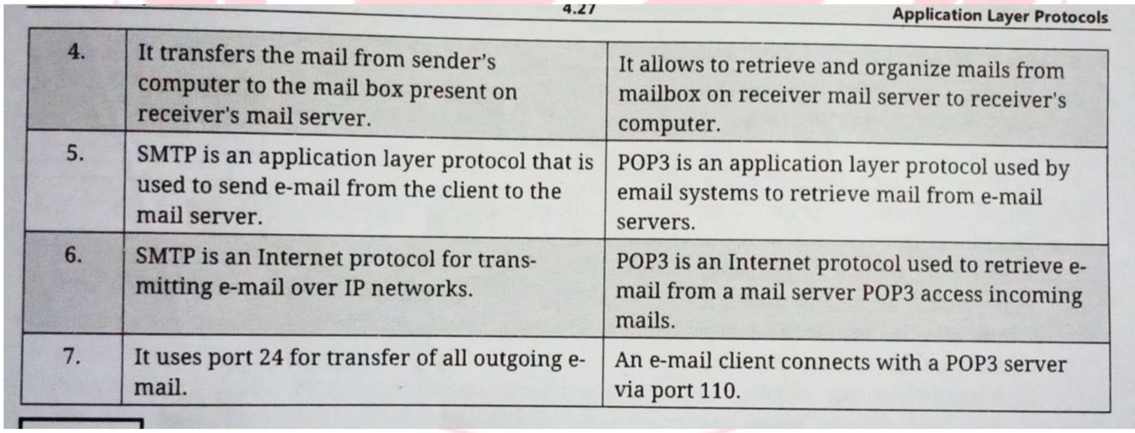


Fig. 4.32

Difference between SMTP and POP3:



Sr. No.	SMTP	POP3
1.	It is message transfer agent.	It is message access agent.
2.	Stands for Simple Mail Transfer Protocol.	Stands for Post Office Protocol version 3.
3.	Between sender and sender mail server and between sender mail server and receiver mail server.	Between receiver and receiver mail server.
4.	It transfers the mail from sender's computer to the mail box present on receiver's mail server.	It allows to retrieve and organize mails from mailbox on receiver mail server to receiver's computer.
5.	SMTP is an application layer protocol that is used to send e-mail from the client to the mail server.	POP3 is an application layer protocol used by email systems to retrieve mail from e-mail servers.
6.	SMTP is an Internet protocol for transmitting e-mail over IP networks.	POP3 is an Internet protocol used to retrieve e-mail from a mail server POP3 access incoming mails.
7.	It uses port 24 for transfer of all outgoing e-mail.	An e-mail client connects with a POP3 server via port 110.

Contd...

IMAP

- IMAP is an application layer protocol used to retrieve emails from a mail server.

- It was designed to allow multiple email clients to manage the same mailbox.
- Emails remain on the server until the user explicitly deletes them.
- IMAP is more commonly used on internal networks than on the public Internet.
- The current version is IMAP4, first proposed in 1986.

Versions of IMAP

1. Original IMAP
2. IMAP2
3. IMAP2bis
4. IMAP3
5. IMAP4

- IMAP allows users to manipulate emails directly on the server without downloading them.
- Users view copies of messages in their email client while the originals stay on the server.
- It supports remote access, making it ideal for multi-device usage.

Characteristics of IMAP

- IMAP is a pull protocol.
- Uses TCP at the transport layer.
- Operates on port 143 (or 993 for secure IMAPS).
- Follows a client-server model.
- It is an application layer protocol.

- IMAP is an in-band protocol (commands and data share the same channel).
- Supports distributed mail storage across multiple servers.

Functions of IMAP4

- Users can check email headers before downloading full messages.
- Users can search email content for specific strings before downloading.
- Supports partial download of emails—useful for large attachments or limited bandwidth.
- Users can create, delete, or rename mailboxes directly on the server.
- Allows creation of a hierarchy of mailboxes, similar to folders for organizing emails.

IMAP Commands

Command	Description
IMAP_LOGIN	Opens the connection and authenticates the user.
SELECT	Selects a mailbox to access its messages.
CREATE	Creates a new mailbox with a specified name.
DELETE	Permanently deletes a mailbox.
RENAME	Renames an existing mailbox.
LOGOUT	Ends the session; server sends a BYE response before closing connection.

Comparison between POP and IMAP

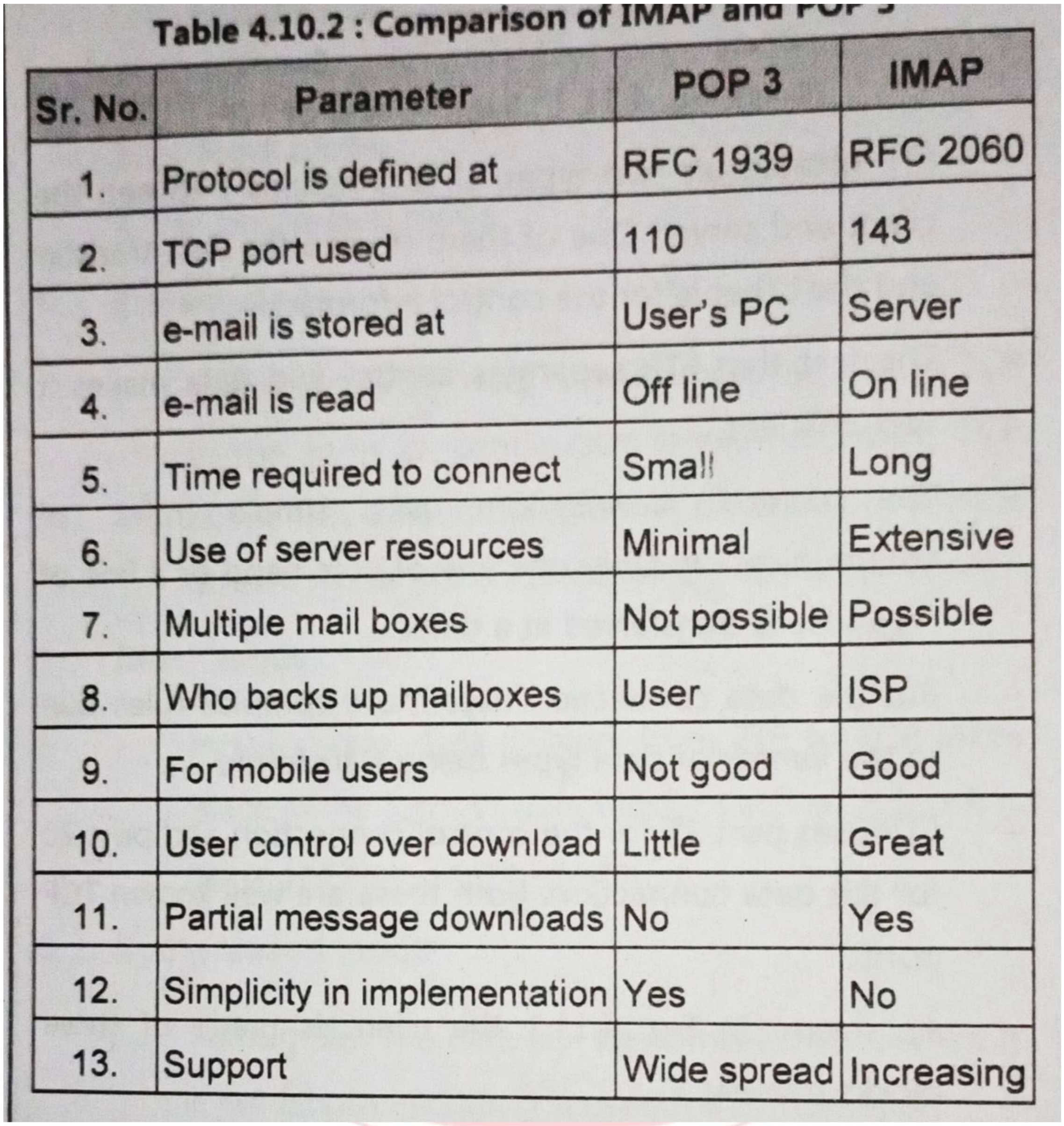
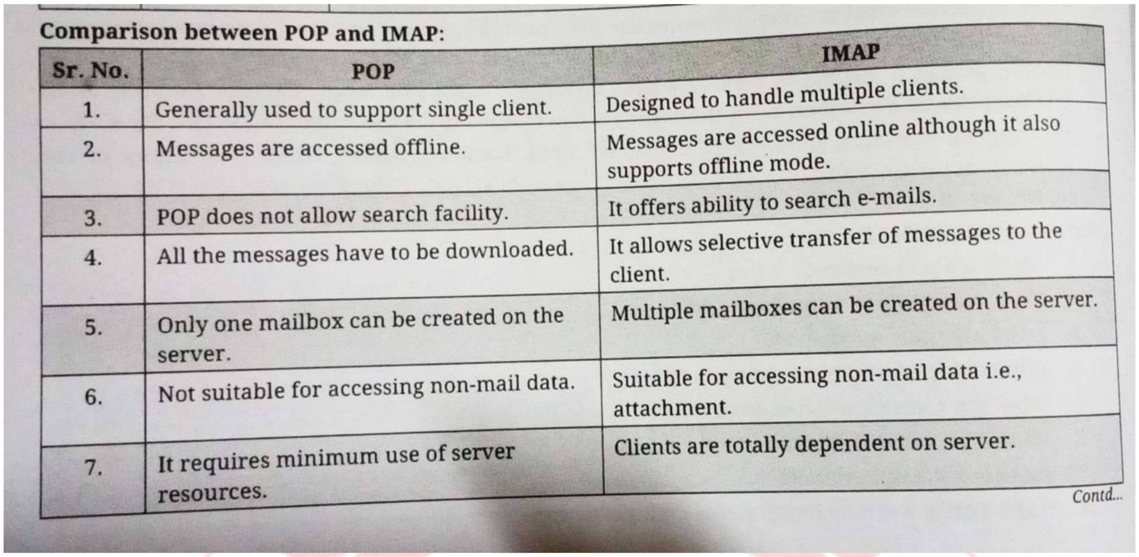


Table 4.10.2 : Comparison of IMAP and POP 3

Sr. No.	Parameter	POP 3	IMAP
1.	Protocol is defined at	RFC 1939	RFC 2060
2.	TCP port used	110	143
3.	e-mail is stored at	User's PC	Server
4.	e-mail is read	Off line	On line
5.	Time required to connect	Small	Long
6.	Use of server resources	Minimal	Extensive
7.	Multiple mail boxes	Not possible	Possible
8.	Who backs up mailboxes	User	ISP
9.	For mobile users	Not good	Good
10.	User control over download	Little	Great
11.	Partial message downloads	No	Yes
12.	Simplicity in implementation	Yes	No
13.	Support	Wide spread	Increasing



Comparison between POP and IMAP:

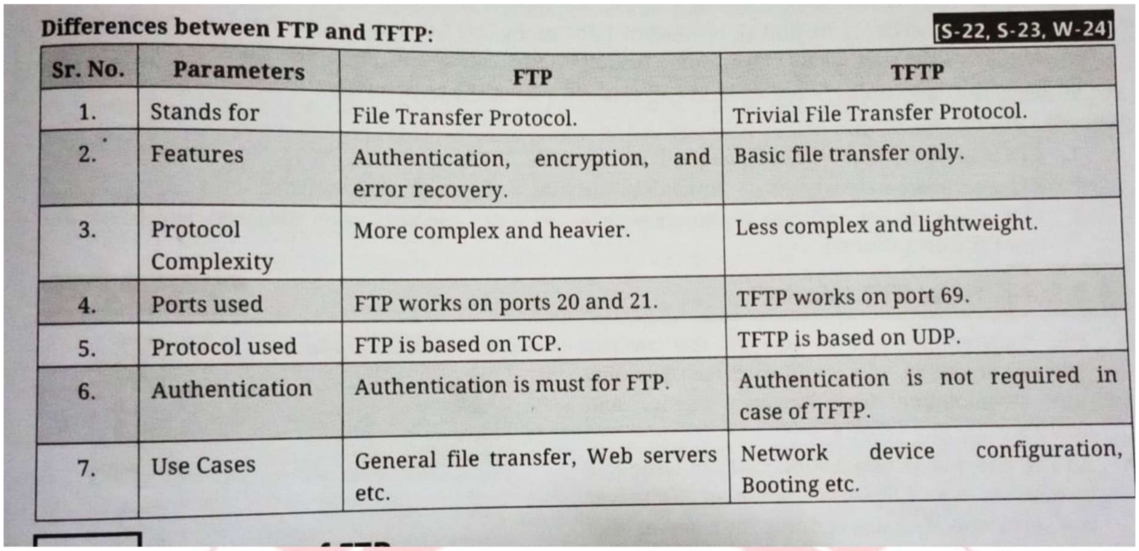
Sr. No.	POP	IMAP
1.	Generally used to support single client.	Designed to handle multiple clients.
2.	Messages are accessed offline.	Messages are accessed online although it also supports offline mode.
3.	POP does not allow search facility.	It offers ability to search e-mails.
4.	All the messages have to be downloaded.	It allows selective transfer of messages to the client.
5.	Only one mailbox can be created on the server.	Multiple mailboxes can be created on the server.
6.	Not suitable for accessing non-mail data.	Suitable for accessing non-mail data i.e., attachment.
7.	It requires minimum use of server resources.	Clients are totally dependent on server.

Contd...

1.3 Q: what is File Transfer Protocol (FTP)

- **FTP (File Transfer Protocol)** is a standard network protocol used to transfer files between a client and a server over a TCP/IP network. It allows users to upload, download, rename, delete, and manage files on remote servers.
- Transfer files between computers
- Upload website content to a web server
- Download software, documents, or media from servers
- **Port 21**: Used for control commands
- **Port 20**: Used for data transfer (active mode)
- **Client** initiates a connection to the FTP server.
- **Authentication** is done using a username and password.
- Files are transferred using **control** and **data** channels.
- The user can perform file operations like upload, download, rename, etc.

Q: Difference between FTP and TFTP



Differences between FTP and TFTP: [S-22, S-23, W-24]

Sr. No.	Parameters	FTP	TFTP
1.	Stands for	File Transfer Protocol.	Trivial File Transfer Protocol.
2.	Features	Authentication, encryption, and error recovery.	Basic file transfer only.
3.	Protocol Complexity	More complex and heavier.	Less complex and lightweight.
4.	Ports used	FTP works on ports 20 and 21.	TFTP works on port 69.
5.	Protocol used	FTP is based on TCP.	TFTP is based on UDP.
6.	Authentication	Authentication is must for FTP.	Authentication is not required in case of TFTP.
7.	Use Cases	General file transfer, Web servers etc.	Network device configuration, Booting etc.

Architecture of FTP

- Server has 2 major components
- Client also has 2 major components
- Control connection is made between control processes at client and server
- Data connection is made between data transfer processes
- Control connection remains open during entire FTP interactive sessions
- Data connection is opened when user wants to transmit file
- Closed after file transfer

Benefits of FTP

- Used to transfer files throughout network
- Provides accessing to both directories and files with certain operations
- Used to list and manipulate directories, type file contents, copy files between hosts and other file operations

How FTP work?

- FTP is used for uploading and downloading files between client and server
- FTP client can issue FTP command to FTP server→ FTP server responds to it
- FTP command is used to→ change directories, change transfer mode between binary to ASCII, upload files, download files
- FTP uses TCP for communication

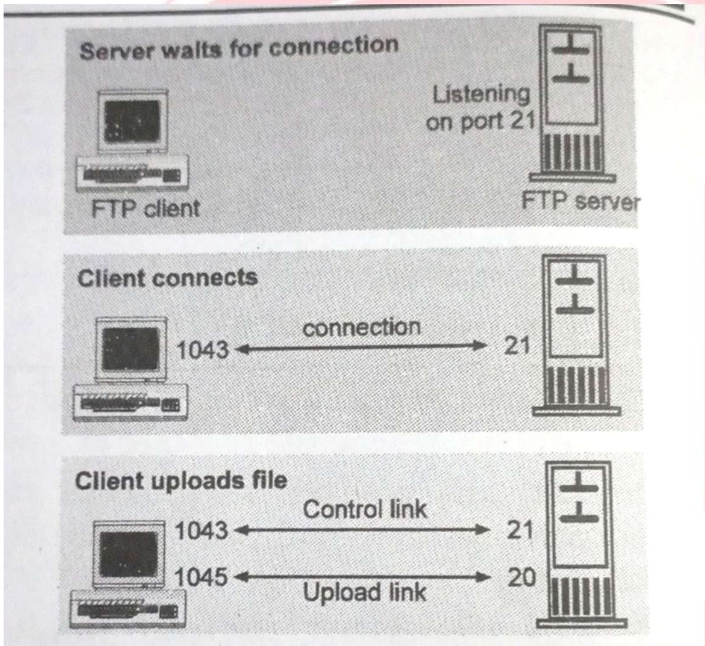


Fig. 4.35: Working of File Transfer Protocol (FTP)

- Port no 21 on FTP server listens for connection attempt from client
- It is also used as control port fro establishing connection
- Once connection is established server opens port 20
- Forms new connection with client to transfer data
- 2 types of protocols required: FTP and TFTP

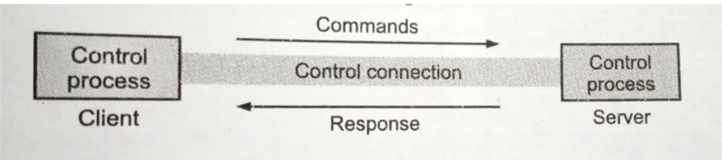
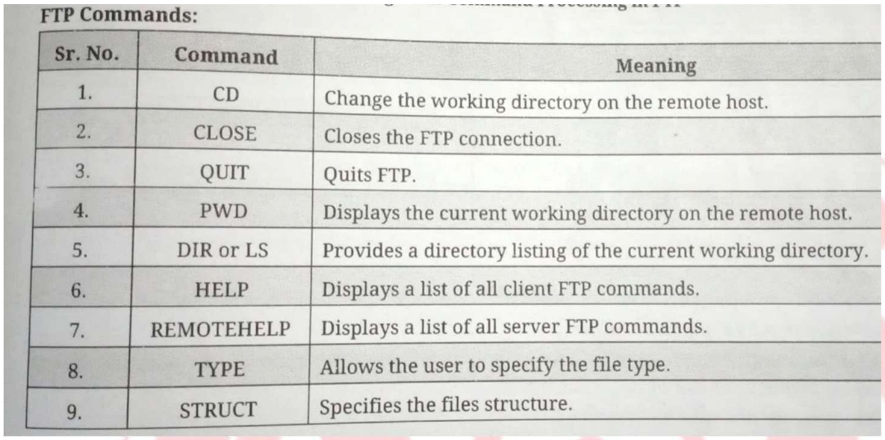


Fig. 4.36: Command Processing in FTP

FTP Commands



FTP Commands:

Sr. No.	Command	Meaning
1.	CD	Change the working directory on the remote host.
2.	CLOSE	Closes the FTP connection.
3.	QUIT	Quits FTP.
4.	PWD	Displays the current working directory on the remote host.
5.	DIR or LS	Provides a directory listing of the current working directory.
6.	HELP	Displays a list of all client FTP commands.
7.	REMOTEHELP	Displays a list of all server FTP commands.
8.	TYPE	Allows the user to specify the file type.
9.	STRUCT	Specifies the files structure.

File transfer in FTP

- File transfer occurs over data connection
- File transfer means following 3 things

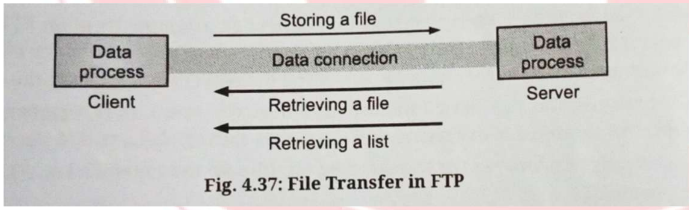
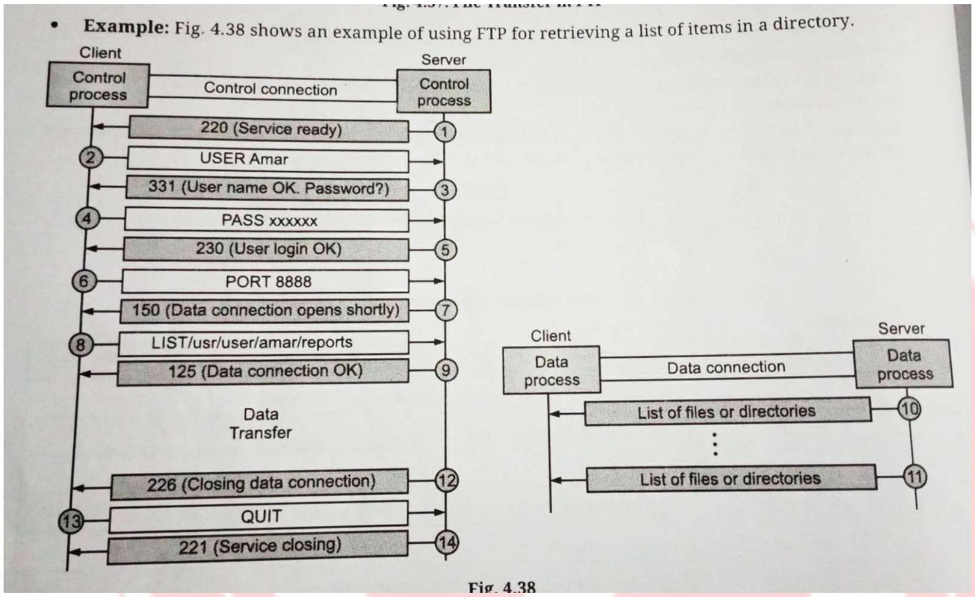


Fig. 4.37: File Transfer in FTP

- **RETR command**
- File is to be copied from server to client→ download
- Also called retrieving of file→ done under RETR command
- **STOR command**
- File is to be copied from client to server→ upload
- Also called as storing of file→done under STOR command
- **LIST command**

- List of directories or file names are to be sent from server to client
- Done under supervision of LIST command
- The list is sent over data connection

e.g.



- **Example:** Fig. 4.38 shows an example of using FTP for retrieving a list of items in a directory.

Fig. 4.38

Steps in file transfer in FTP

Step 1: After the control connection to port 21 is created, the FTP server sends the 220 (service ready) response indicating that it is ready to accept a connection.

Step 2: The client sends the USER command.

Step 3: The server responds with 331 (user name is OK, password is required).

Step 4: The client sends the PASS command.

Step 5: The server responds with 230 (user is OK, proceed).

Step 6: The client issues a passive open on an ephemeral port for the data connection and sends the PORT command (over the control connection) to give this port number to the server.

Step 7: The server does not open the connection at this time, but it prepares itself for listening at the active open on the data connection between port 20 (server side) and the ephemeral port received from the client. It sends response 140 (data connection will open shortly).

Step 8: The client sends the LIST message.

Step 9: The server responds with 124 and opens the data connection.

Step 10: The server then sends the list of the files or directories (as a file) on the data connection. When the whole list (file) is sent, the server responds with 226 (closing data connection over the control connection).

Step 11: The client now has two choices. It can use the QUIT command to request the closing of the control connection or it can send another command to start another activity (and eventually open another data connection). In our example, the client sends a QUIT command.

Step 12: After receiving the QUIT command, the server responds with 221 (service closing) and then closes the control connection.

FTP file transfers file in three difference modes

1. **Stream Mode**

   - Default mode used in most FTP transfers.
   - Data is sent as a continuous stream of bytes.
   - The end of the file is marked by closing the data connection.

- Ideal for simple and fast transfers, especially for text files.

2. **Block Mode**

- Data is sent in blocks, each with a header that describes the block size and type.
- Useful for structured data or when error recovery is needed.
- Allows for more control and flexibility during transmission.

3. **Compressed Mode**

- Data is compressed using algorithms like Run-Length Encoding (RLE) before transmission.
- Reduces the amount of data sent, which is helpful in low-bandwidth environments.
- Especially beneficial for large files or repetitive data.

Application of FTP

1. Uploading webpages to web servers for publishing on the Internet.
2. Browsing and downloading files from public software sites.
3. Organizations use FTP to allow employees to share files across different locations and branch offices.
4. Transferring large files among two parties that are too large for email attachments.
5. Downloading and uploading content like university assignments via an FTP server.
6. Distributing the latest revisions of software by software vendors.
7. Employees use FTP to securely share files with coworkers and external business partners.

8. IT teams use FTP to transfer data back to disaster recovery (DR) sites.

1.3.2 Anonymous File Transfer Protocol

- A network protocol used for transmitting files over TCP-based networks.
- Operates at Layer 7 of the OSI model (Application Layer).
- Allows users to transfer files anonymously between computers.
- ⍰ Users can access files without a designated username or password.
- Typically used to access publicly available data on websites or servers.
- The user ID is "anonymous", and the password may be:
    - Provided by the server
    - Entered by the user (often an email address)
    - Left blank
- Enables access to remote servers or archive sites without identification.
- Users have limited access—usually read-only—to public files.
- The server owner controls what information is accessible.
- Anonymous FTP is ideal for public file distribution, such as software or documents.

How do Anonymous FTP sessions work?

4.4. Remote logging

Remote Logging is the process of collecting log data from multiple systems or devices and sending it to a centralized server over a network. It helps in monitoring, troubleshooting, and securing systems by analysing logs from different sources in one place.

How remote logging works?

1. **Log Generation**

- Logs are created by:
  - Operating systems
  - Applications
  - Network devices (routers, switches, firewalls)
- These logs contain:
  - System events
  - Error messages
  - Security alerts
  - User activities

2. **Log Transmission**

  - Logs are sent over the network to a central logging server.
- Common protocols used:
  - Syslog – Standard protocol for sending logs
  - SNMP (Simple Network Management Protocol) – Used for network monitoring
  - SSH (Secure Shell) – Secure transmission of logs
  - HTTP/HTTPS – Web-based log transmission
  - FTP/SFTP – File-based log transfer

3. Log Storage and Analysis

- Logs are stored in:
  - Log files
  - Databases
  - Cloud-based storage
- Analysis tools are used to:
  - Search and filter logs
  - Detect unusual activity or errors
  - Generate reports and alerts

TELNET

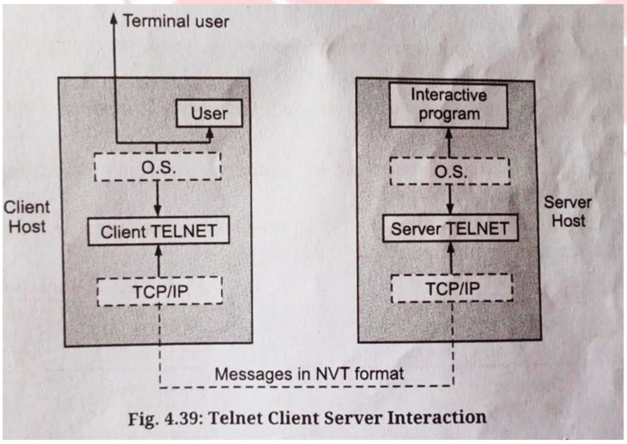
- Terminal Network
- Protocol used for log in to remote computer on internet
- It's an application layer protocol
- Bidirectional, connection oriented, client server protocol
- Uses port no 23
- 

Fig. 4.39: Telnet Client Server Interaction
- Accessing routers, switches, and servers

- Testing open ports and services
- Educational purposes for learning remote access
- No encryption (data sent in plain text)
- Replaced by **SSH (Secure Shell)** for secure communication
- Not suitable for modern secure networks

Logging in TELNET

2 parts of logging process

1. **Local Login**

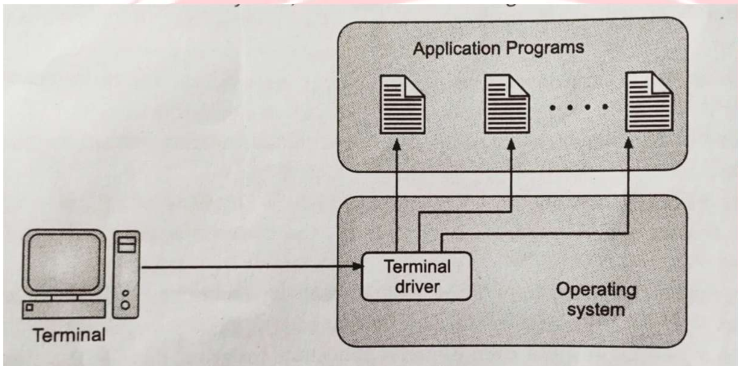
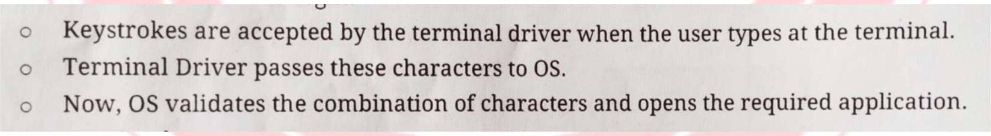


Fig. 4.40: Local Login

- **The Procedure of Local Login**



  - Keystrokes are accepted by the terminal driver when the user types at the terminal.
  - Terminal Driver passes these characters to OS.
  - Now, OS validates the combination of characters and opens the required application.

2. **Remote Logging**

- User can log in to remote site → computer or services available on remote computer
- User can transfer the result of local processing from remote computer to local computer

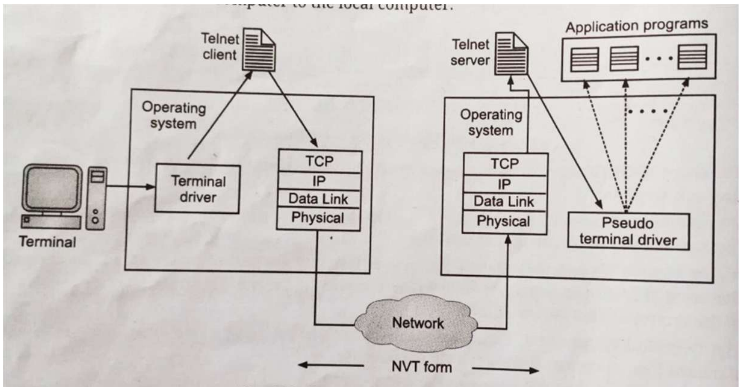
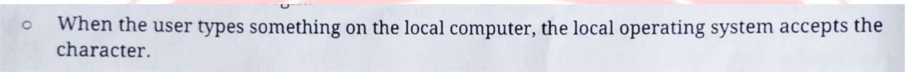
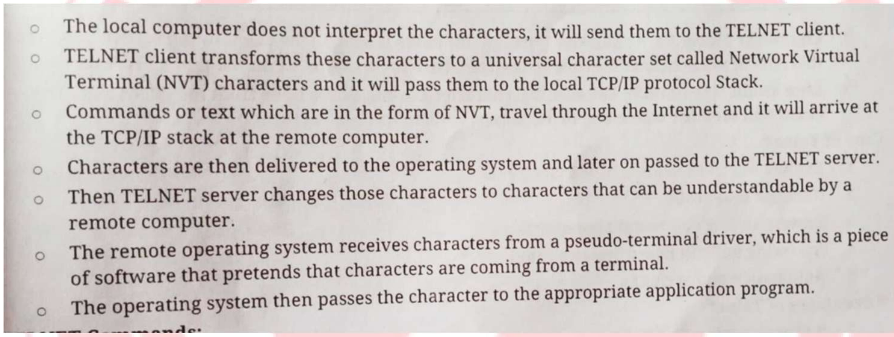


Fig. 4.41: Remote Login-in Logging

- **The Procedure of Remote Login**

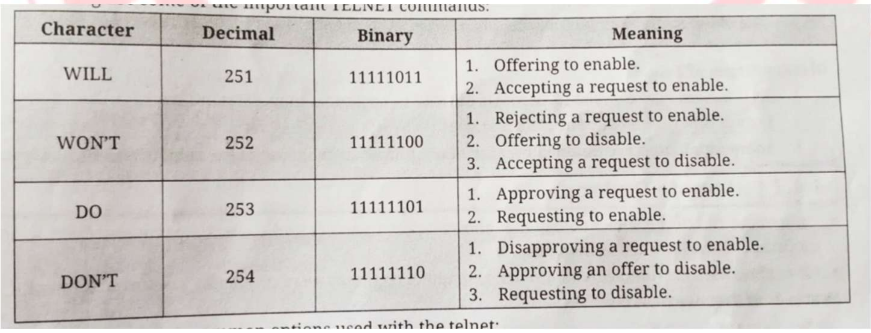


- When the user types something on the local computer, the local operating system accepts the character.

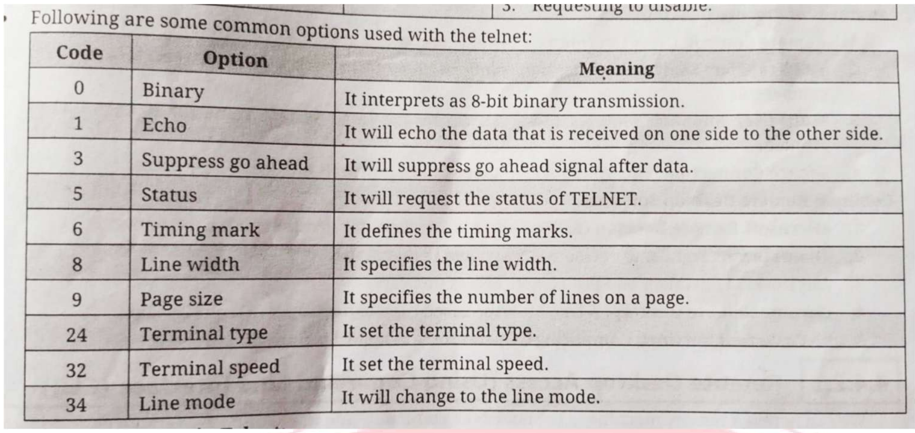


- The local computer does not interpret the characters, it will send them to the TELNET client.
- TELNET client transforms these characters to a universal character set called Network Virtual Terminal (NVT) characters and it will pass them to the local TCP/IP protocol Stack.
- Commands or text which are in the form of NVT, travel through the Internet and it will arrive at the TCP/IP stack at the remote computer.
- Characters are then delivered to the operating system and later on passed to the TELNET server.
- Then TELNET server changes those characters to characters that can be understandable by a remote computer.
- The remote operating system receives characters from a pseudo-terminal driver, which is a piece of software that pretends that characters are coming from a terminal.
- The operating system then passes the character to the appropriate application program.

TELNET Commands



Character	Decimal	Binary	Meaning
WILL	251	11111011	1. Offering to enable. 2. Accepting a request to enable.
WON'T	252	11111100	1. Rejecting a request to enable. 2. Offering to disable. 3. Accepting a request to disable.
DO	253	11111101	1. Approving a request to enable. 2. Requesting to enable.
DON'T	254	11111110	1. Disapproving a request to enable. 2. Approving an offer to disable. 3. Requesting to disable.



3. Requesting to disable.

Following are some common options used with the telnet:

Code	Option	Meaning
0	Binary	It interprets as 8-bit binary transmission.
1	Echo	It will echo the data that is received on one side to the other side.
3	Suppress go ahead	It will suppress go ahead signal after data.
5	Status	It will request the status of TELNET.
6	Timing mark	It defines the timing marks.
8	Line width	It specifies the line width.
9	Page size	It specifies the number of lines on a page.
24	Terminal type	It set the terminal type.
32	Terminal speed	It set the terminal speed.
34	Line mode	It will change to the line mode.

Modes of Operations in TELNET

1. Default Mode

- This is the **standard mode** used when Telnet is first started.
- It allows basic communication between client and server.
- The mode may switch automatically based on server configuration.

2. Character Mode

- In this mode, **each character typed** by the user is immediately sent to the remote server.
- Useful for **real-time command execution**.
- Common in interactive sessions like remote shell access.

3. Line Mode

- In this mode, the **entire line of text** is sent only after the user presses Enter.
- Reduces network traffic by sending fewer packets.
- Suitable for commands that require full-line input.

Uses of TELNET

- Remote login to servers and network devices
- Testing open ports and services (e.g., HTTP, FTP, SMTP)
- Accessing routers and switches for configuration
- Educational purposes for learning command-line networking

Advantages of TELNET

- Simple and easy to use
- Provides direct access to remote systems
- Useful for troubleshooting and testing network services
- Available on most operating systems

Disadvantages of TELNET

- No encryption – data is sent in plain text
- Not secure for sensitive information
- Vulnerable to man-in-the-middle attacks
- Replaced by SSH (Secure Shell) in modern systems

Remote Desktop

- It is a technology that allows user to access and control a computer from another location over a network or the internet
- It enables user to interact with a remote computer as if they are physically present in front of it

Features of Remote Desktop

1. **Remote Control:** user can control mouse, keyboard, applications remotely
2. **File Transfer:** between local and remote computers

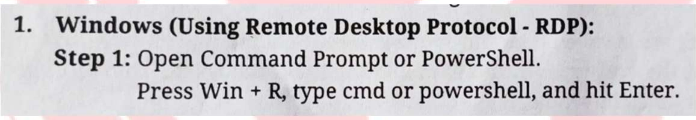
3. **Multi-User Support:** multiple users can access system remotely(if permission granted)
4. **Secure Connection:** encryption and authentication

Common Remote Desktop Software

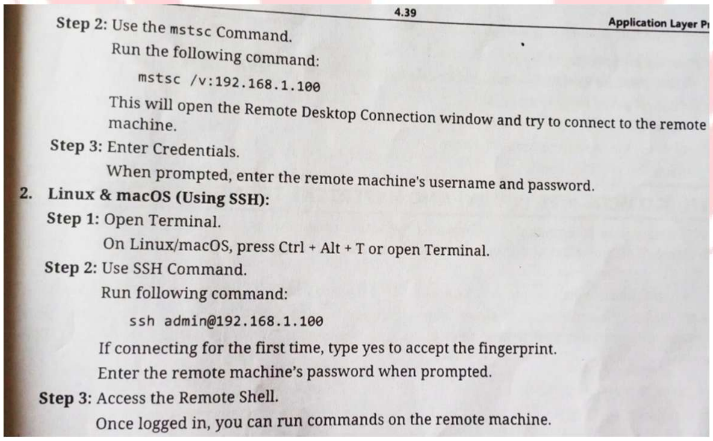
1. **Microsoft Remote Desktop (RDP):** windows
2. **TeamViewer:** personal use
3. **AnyDesk:** lightweight and fast
4. **Chrome Remote Desktop:** free browser based by google
5. **VNC (Virtual Network Computing):** open source

Remote Desktop Access (Using Command Line Interface (CLI))

- Remote desktop can be accessed using command line



1. **Windows (Using Remote Desktop Protocol - RDP):**
   Step 1: Open Command Prompt or PowerShell.
   Press Win + R, type cmd or powershell, and hit Enter.



4.39 Application Layer Pr

Step 2: Use the mstsc Command.
Run the following command:

```
mstsc /v:192.168.1.100
```

This will open the Remote Desktop Connection window and try to connect to the remote machine.

Step 3: Enter Credentials.
When prompted, enter the remote machine's username and password.

2. **Linux & macOS (Using SSH):**

Step 1: Open Terminal.
On Linux/macOS, press Ctrl + Alt + T or open Terminal.

Step 2: Use SSH Command.
Run following command:

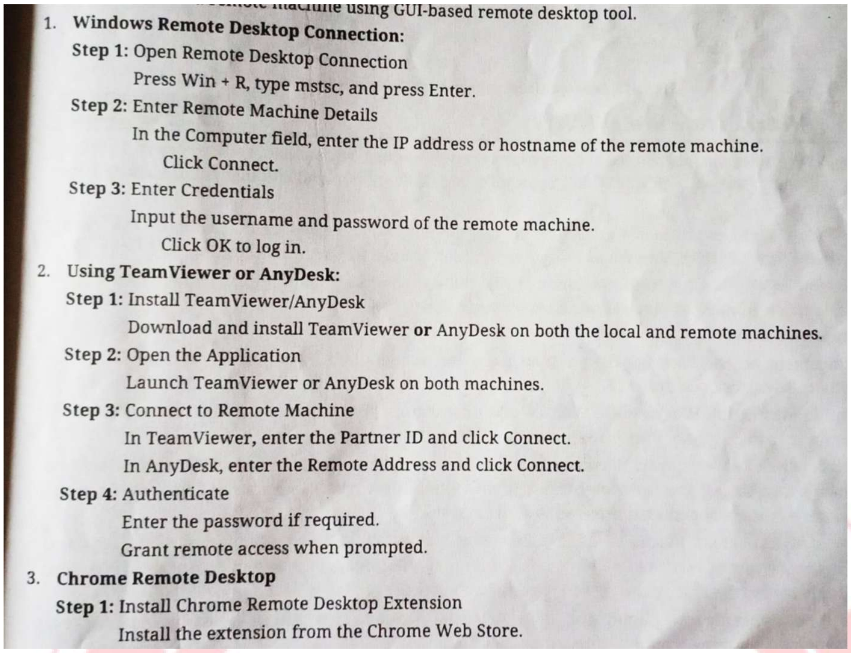
```
ssh admin@192.168.1.100
```

If connecting for the first time, type yes to accept the fingerprint.
Enter the remote machine's password when prompted.

Step 3: Access the Remote Shell.
Once logged in, you can run commands on the remote machine.

Remote Desktop Access (Using GUI-Based Remote Desktop Tool)

- we can access remote desktop using GUI based remote desktop tool



machine using GUI-based remote desktop tool.

1. **Windows Remote Desktop Connection:**

   Step 1: Open Remote Desktop Connection

   Press Win + R, type mstsc, and press Enter.

   Step 2: Enter Remote Machine Details

   In the Computer field, enter the IP address or hostname of the remote machine.

   Click Connect.

   Step 3: Enter Credentials

   Input the username and password of the remote machine.

   Click OK to log in.

2. **Using TeamViewer or AnyDesk:**

   Step 1: Install TeamViewer/AnyDesk

   Download and install TeamViewer or AnyDesk on both the local and remote machines.

   Step 2: Open the Application

   Launch TeamViewer or AnyDesk on both machines.

   Step 3: Connect to Remote Machine

   In TeamViewer, enter the Partner ID and click Connect.

   In AnyDesk, enter the Remote Address and click Connect.

   Step 4: Authenticate

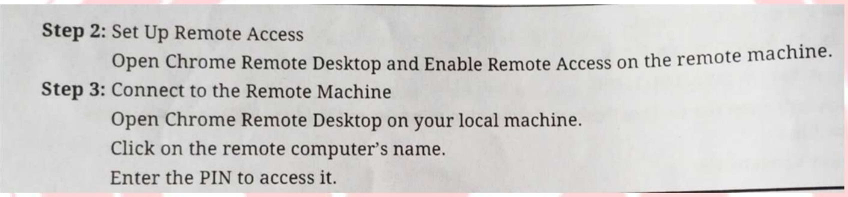
   Enter the password if required.

   Grant remote access when prompted.

3. **Chrome Remote Desktop**

   Step 1: Install Chrome Remote Desktop Extension

   Install the extension from the Chrome Web Store.



   Step 2: Set Up Remote Access

   Open Chrome Remote Desktop and Enable Remote Access on the remote machine.

   Step 3: Connect to the Remote Machine

   Open Chrome Remote Desktop on your local machine.

   Click on the remote computer's name.

   Enter the PIN to access it.

WORLD WIDE WEB (WWW) AND HYPERTEXT TRANSFER PROTOCOL

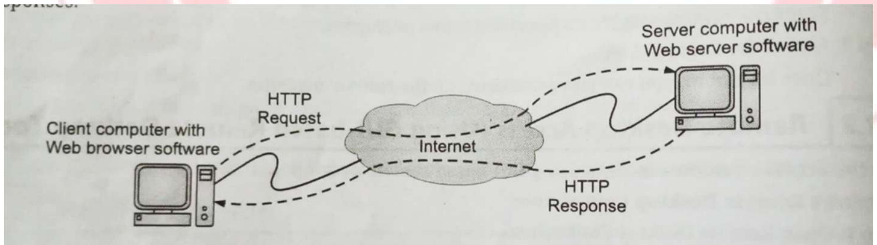


Fig. 4.42: Communication between Client and Server

World Wide Web (WWW)

The World Wide Web (WWW) is a system of interlinked hypertext documents accessed via the internet. It allows users to browse and interact with web pages using a web browser.

Terminologies in WWW

Web Page: Documents written in HTML, linked using hyperlinks, and displayed in browsers.

HTML: Hyper Text Markup Language

Hypertext: is a system of text that contains links (called hyperlinks) to other texts. It allows users to navigate between documents or sections by clicking on linked words or phrases.

Hypermedia: is an extension of hypertext that includes multimedia elements such as text, images, audio, video, and animations linked together

Links: Links or Hyperlinks are clickable elements in a document that connect to another document, section, or resource.

Types of web documents

- **A Static web page**

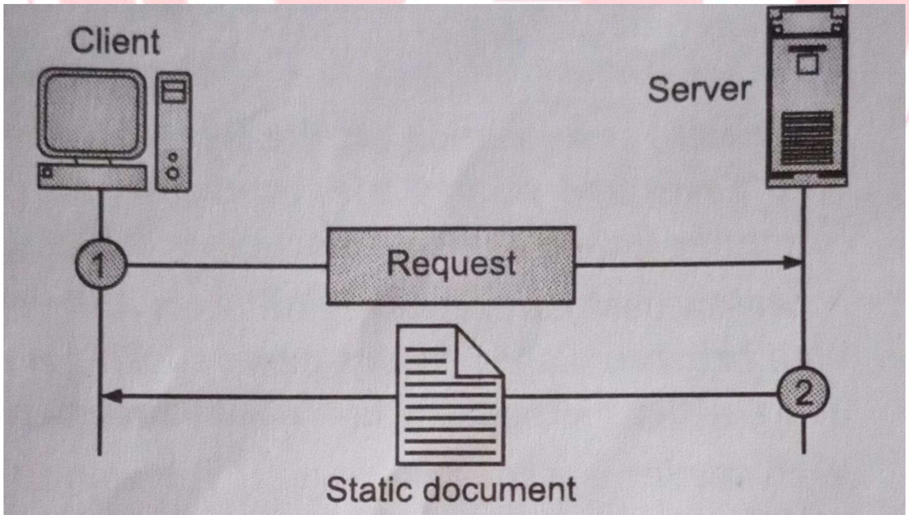


Fig. 4.43: Static Web Page

- A static web page is a fixed-content page that displays the same information to every user. It is created using HTML and does not change unless manually updated by the developer.

- Content remains constant
- Fast loading speed
- Easy to develop and host
- No interaction with databases

Example:

An “About Us” page showing company details.

Technologies Used:

- HTML
- CSS
- **A dynamic web page**

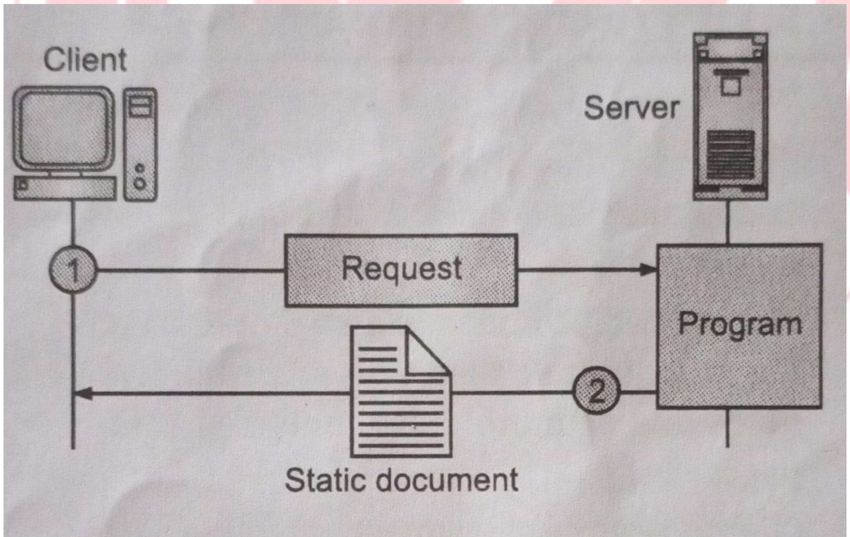


Fig. 4.44: Dynamic Document using CGI

- A dynamic web page displays content that can change based on user interaction, time, or data from a server. It is generated in real-time using server-side or client-side scripting.

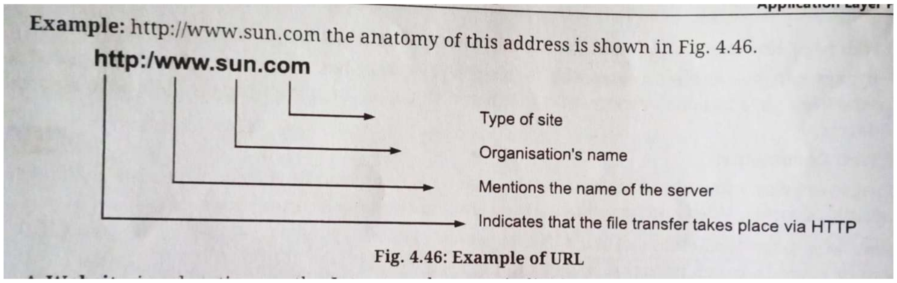
- Content changes dynamically
- Interactive and personalized
- Connects to databases
- Requires scripting languages
- Example:
  - A login page or shopping cart that updates based on user input.
  - Technologies Used:
  - HTML + CSS
  - JavaScript (Client-side)
  - PHP, ASP.NET, Node.js (Server-side)
  - Database (e.g., MySQL, MongoDB)

URL: A URL is the address used to access resources on the internet. It specifies the location of a web page or file.

https://www.msbte.org.in/exam/results.html

Components of a URL:

- Protocol: https
- Host: www.msbte.org.in
- Port Number: Default is 80 for HTTP, 443 for HTTPS
- Path: /exam/results.html

- 

Example: http://www.sun.com the anatomy of this address is shown in Fig. 4.46.

http:/www.sun.com
Type of site
Organisation's name
Mentions the name of the server
Indicates that the file transfer takes place via HTTP

Fig. 4.46: Example of URL

PROTOCOL

A protocol is a set of rules that defines how data is transmitted over the internet.

- Common Protocols:
  - HTTP – HyperText Transfer Protocol
  - HTTPS – Secure version of HTTP
  - FTP – File Transfer Protocol

Host

The host is the domain name or IP address of the server where the website is stored.

- Example:

In https://www.msbte.org.in, the host is www.msbte.org.in.

Port Number

A port number identifies a specific process or service on a server.

- Common Ports:
  - 80 → HTTP
  - 443 → HTTPS

- 21 → FTP

Usually hidden in URLs unless a custom port is used.

Path: The path specifies the location of a file or resource on the server.

the location of the resource in the directory structure of the server.

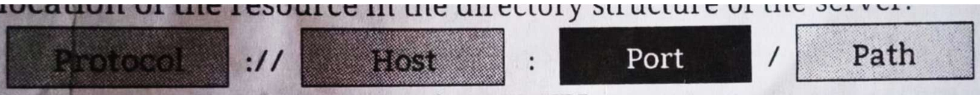


Fig. 4.45: URL

Website

A website is a collection of related web pages hosted on a server and accessible via a domain name.

Features:

- Contains text, images, videos, and links
- Can be static or dynamic
- Accessed using a browser

Web Server

A web server is a computer system that stores, processes, and delivers web pages to users.

- Functions:
  - Handles HTTP requests
  - Serves HTML, CSS, JS files
  - Manages client-server communication

- Examples:

Apache, Nginx, Microsoft IIS

Web Publishing or Online publishing

Web publishing is the process of uploading content (text, images, videos) to a website so it can be accessed online.

- Steps Involved:

1. Create content using HTML/CSS
2.
3. Host it on a web server
4. Make it accessible via a domain name

- Tools Used:
  - CMS (Content Management Systems) like WordPress
  - Web hosting platforms
  - FTP clients

Architecture of WWW

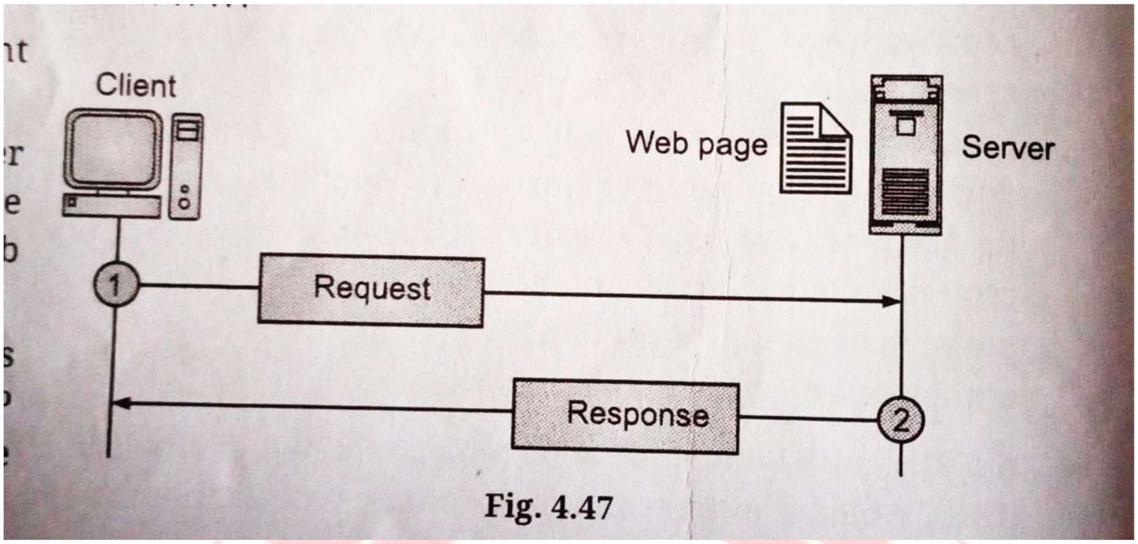


Fig. 4.47

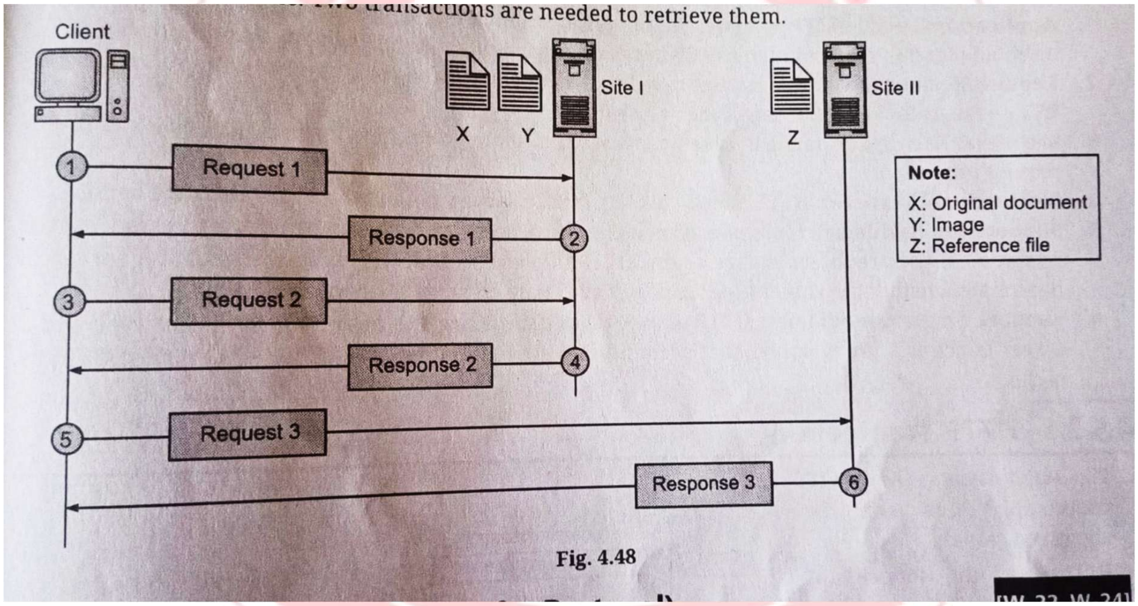


Fig. 4.48

The World Wide Web (WWW) is a system of interlinked hypertext documents accessed via the internet. Its architecture defines how users, browsers, servers, and protocols interact to deliver web content.

Components of WWW Architecture:

Client (Web Browser)

- Software used by users to access web pages.

- Sends requests to web servers using HTTP/HTTPS.
- Examples: Google Chrome, Mozilla Firefox, Microsoft Edge

Web Server

- Stores and serves web pages to clients.
- Processes HTTP requests and sends back responses.
- Examples: Apache, Nginx, Microsoft IIS

Web Page

- A document written in HTML, CSS, and JavaScript.
- Can be static or dynamic.
- Delivered by the server and displayed by the browser.

URL (Uniform Resource Locator)

- The address used to access a specific web page.
- Example: https://www.msbte.org.in/exam/results.html
- HTTP/HTTPS Protocol
- Defines rules for communication between client and server.
- HTTP: HyperText Transfer Protocol
- HTTPS: Secure version of HTTP using encryption

Web Application / Website

- A collection of web pages hosted on a server.
- Can include multimedia, forms, and interactive content.

Database (Optional)

- Used in dynamic websites to store and retrieve data.

- Connected to the server via server-side scripts (e.g., PHP, Node.js)

Working of WWW Architecture:

1. User enters a URL in the browser.
2. Browser sends an HTTP request to the web server.
3. Server processes the request and fetches the web page.
4. Server sends the response back to the browser.
5. Browser renders the page for the user.

HTTP (HyperText Transfer Protocol)

HTTP (HyperText Transfer Protocol) is a protocol used for communication between web browsers and web servers. It defines how requests and responses are formatted and transmitted over the internet**.**

1. Simple Request

- A basic HTTP request sent by a client (browser) to a server.
- Contains only essential information like:
- Request method (GET, POST)
- URL
- HTTP version

GET /index.html HTTP/1.1

3. Full request

A complete HTTP request includes:

- Request line
- Headers (e.g., Host, User-Agent)
- Optional body (for POST/PUT)

```
POST /submit-form HTTP/1.1
Host: www.example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 27
name=Amit&email=amit@mail.com
```

Characteristics of HTTP

1. Application level

- HTTP operates at the **application layer** of the OSI model.
- It enables communication between client and server applications.

2. Request/ Response

- Follows a **client-server model**.
- Client sends a **request**, server sends a **response**.

3. Stateless

- Each HTTP request is **independent**.
- Server does **not retain** information about previous requests.

4. Bidirectional Transfer

Supports two-way communication:

- Client → Server (request)
- Server → Client (response)

5. Support for Caching

- HTTP allows **caching** of responses to improve performance.
- Headers like `Cache-Control` and `Expires` manage caching.

6. Support for Intermediaries

- HTTP supports **intermediate systems** like:
  - Proxies
  - Gateways
  - Load balancers
- These help in routing, filtering, and improving performance.

HTTP Transaction

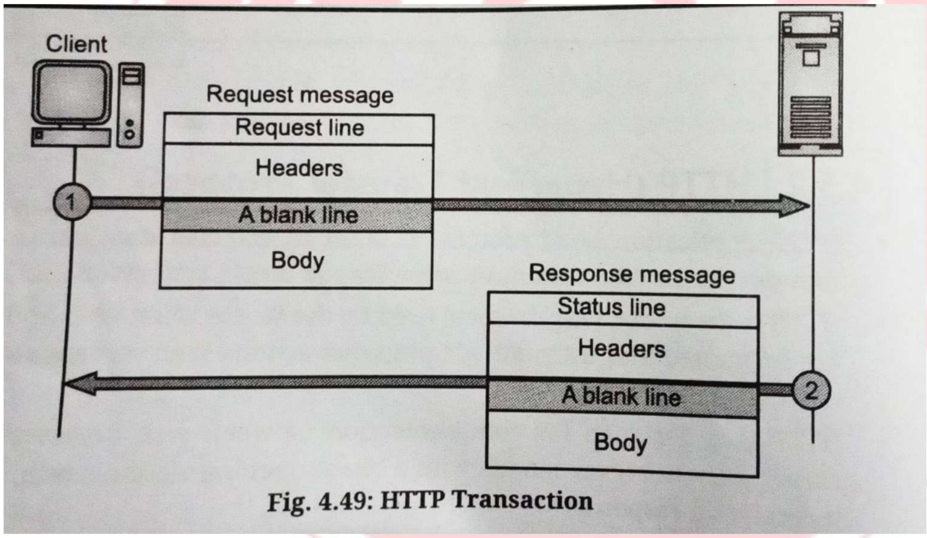


Fig. 4.49: HTTP Transaction

- **HTTP uses services of TCP**
- It is stateless protocol
- Does not keep information about client
- Client initializes message and server replies with response
- **Request Message:** it consist of request line , a header, sometimes body

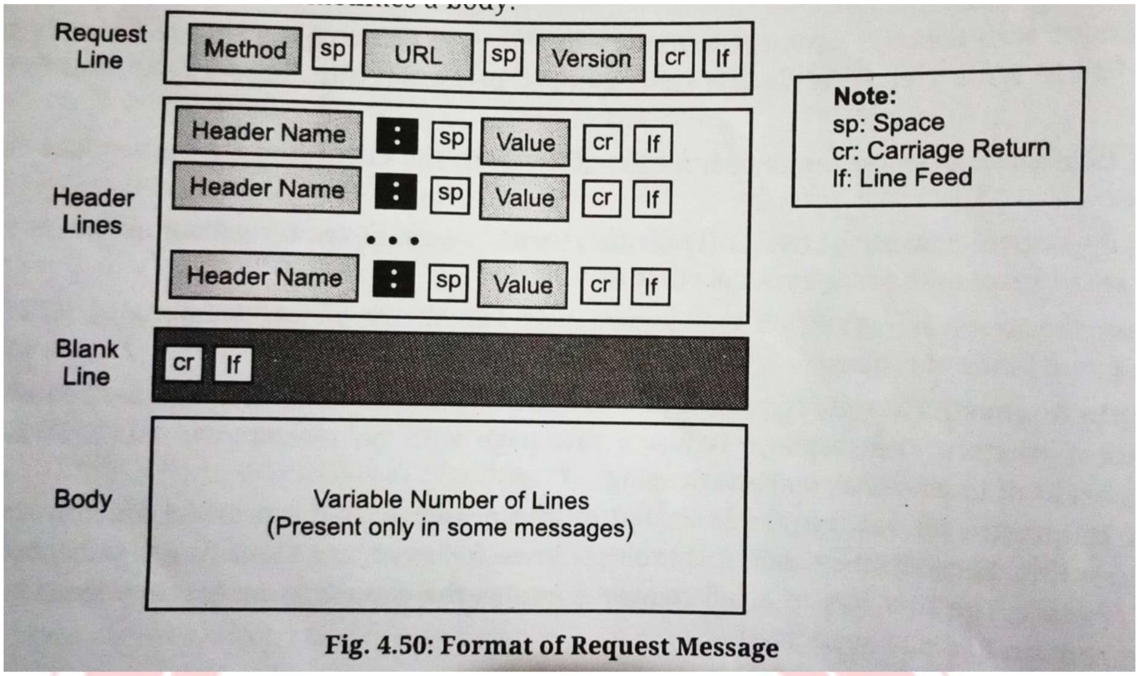
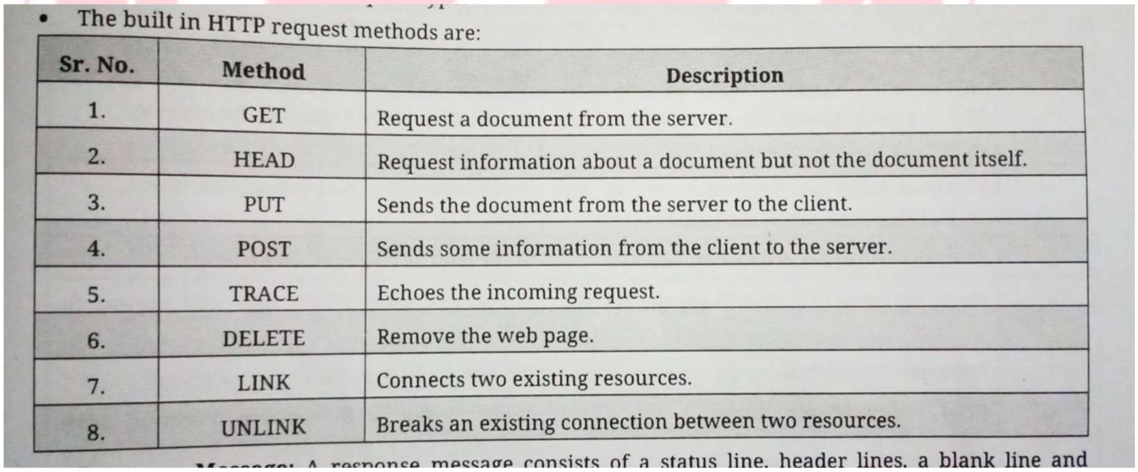


Fig. 4.50: Format of Request Message

- **Request Line:** it's the first line in request message→ 3 fields→ separated by character delimiter→ methods, URL, version→ method field defines request type



- The built in HTTP request methods are:

Sr. No.	Method	Description
1.	GET	Request a document from the server.
2.	HEAD	Request information about a document but not the document itself.
3.	PUT	Sends the document from the server to the client.
4.	POST	Sends some information from the client to the server.
5.	TRACE	Echoes the incoming request.
6.	DELETE	Remove the web page.
7.	LINK	Connects two existing resources.
8.	UNLINK	Breaks an existing connection between two resources.

- **Response Message:** consists of status line, header line, blank line and sometimes body

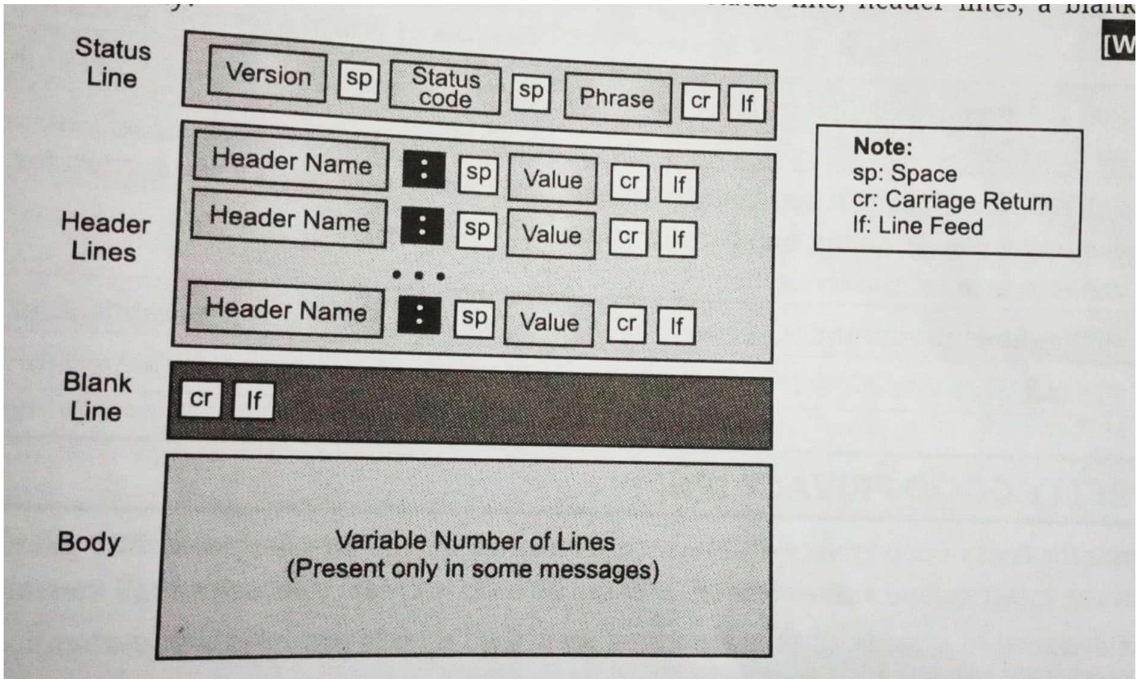
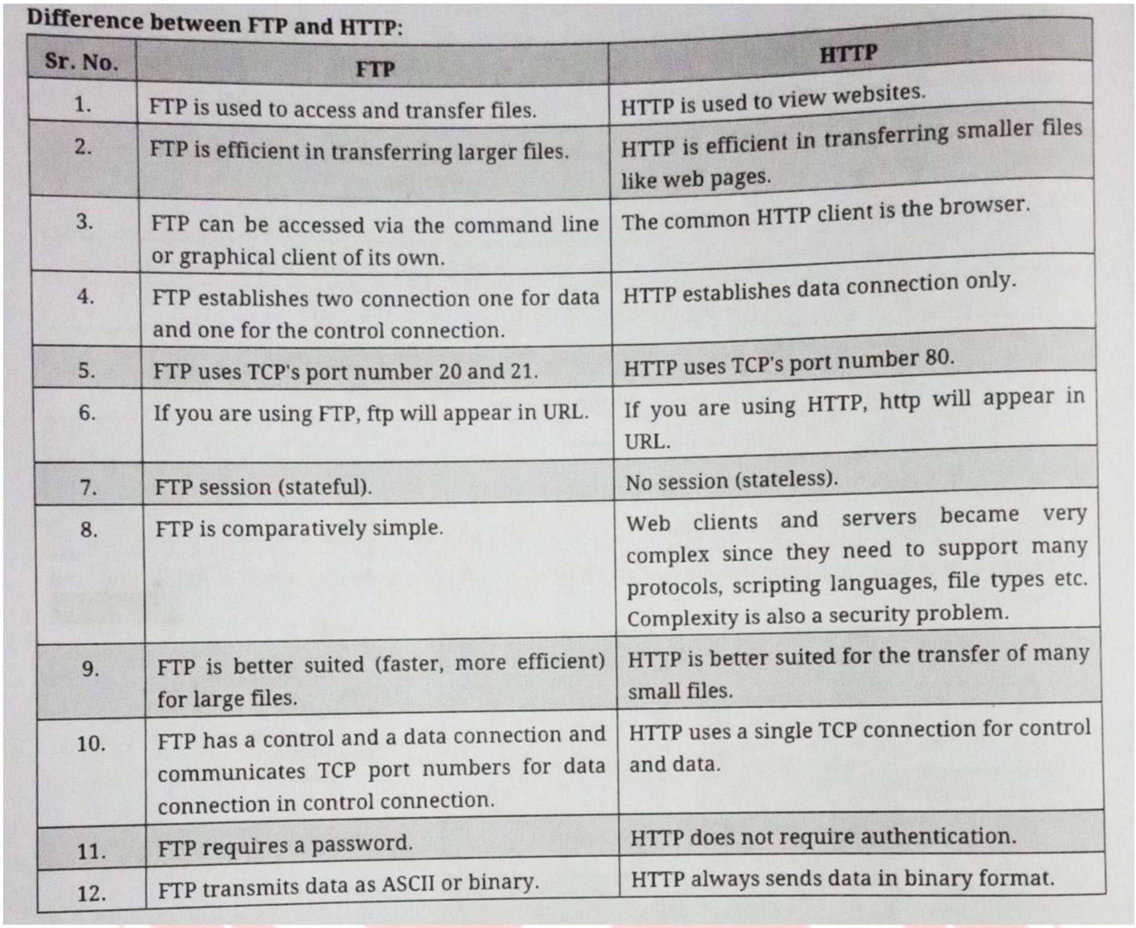


Fig. 4.51: Format of the Response Message

Status Line: first line of response message→ 3 fields→separated by spaces, carriage return and line feed→version, status of the request (3 digit)

Body: contains documents to be sent→ from client to server

Difference between FTP & HTTP



Difference between FTP and HTTP:

Sr. No.	FTP	HTTP
1.	FTP is used to access and transfer files.	HTTP is used to view websites.
2.	FTP is efficient in transferring larger files.	HTTP is efficient in transferring smaller files like web pages.
3.	FTP can be accessed via the command line or graphical client of its own.	The common HTTP client is the browser.
4.	FTP establishes two connection one for data and one for the control connection.	HTTP establishes data connection only.
5.	FTP uses TCP's port number 20 and 21.	HTTP uses TCP's port number 80.
6.	If you are using FTP, ftp will appear in URL.	If you are using HTTP, http will appear in URL.
7.	FTP session (stateful).	No session (stateless).
8.	FTP is comparatively simple.	Web clients and servers became very complex since they need to support many protocols, scripting languages, file types etc. Complexity is also a security problem.
9.	FTP is better suited (faster, more efficient) for large files.	HTTP is better suited for the transfer of many small files.
10.	FTP has a control and a data connection and communicates TCP port numbers for data connection in control connection.	HTTP uses a single TCP connection for control and data.
11.	FTP requires a password.	HTTP does not require authentication.
12.	FTP transmits data as ASCII or binary.	HTTP always sends data in binary format.

Pretty Good Privacy (PGP)

- Invented by Phil Zimmermann
- PGP provides email with → privacy, integrity, authentication, non-repudiation
- It is used to create secure email message
- It uses secret key encryption
- It is open source
- Provides compression by ZIP algorithm

Services of PGP

1. **Authentication→** using SHA-1 algorithm
2. **Confidentiality→** using 3-DES

3. **Confidentiality and Authentication**→ for same message
4. **Compression**→ before encryption to reduce size
5. **E-mail Compatibility**
6. **Segmentation**→ max length of e-mail is 50000 octets→ if more broken down into smaller segments and mailed independently

Disadvantages of PGP Encryption

1. **Difficult to Administration**
2. **Compatibility Issues**→between sender and receiver
3. **Complexity**
4. **No Recovery** → faces problem of losing password→ does not retrieve forgotten password

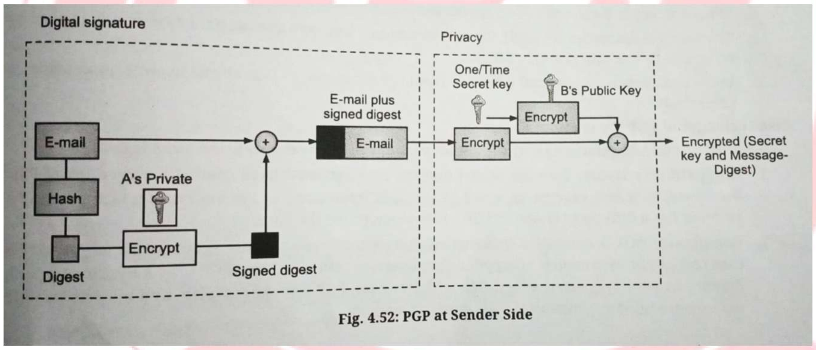


Fig. 4.52: PGP at Sender Side

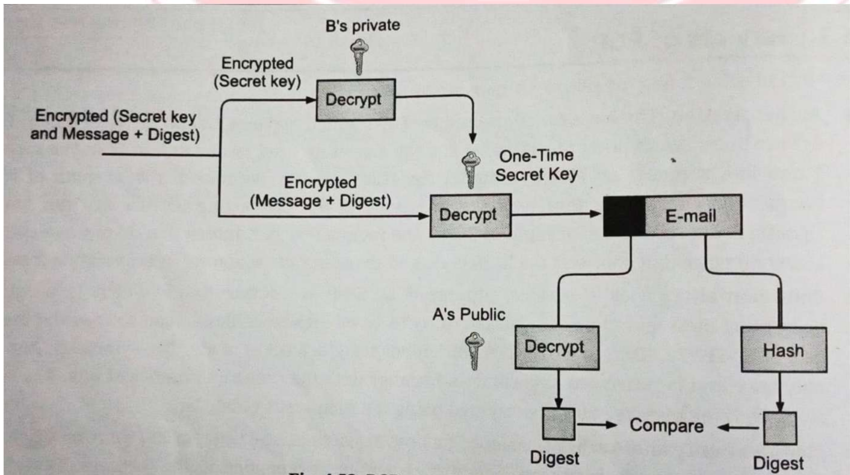
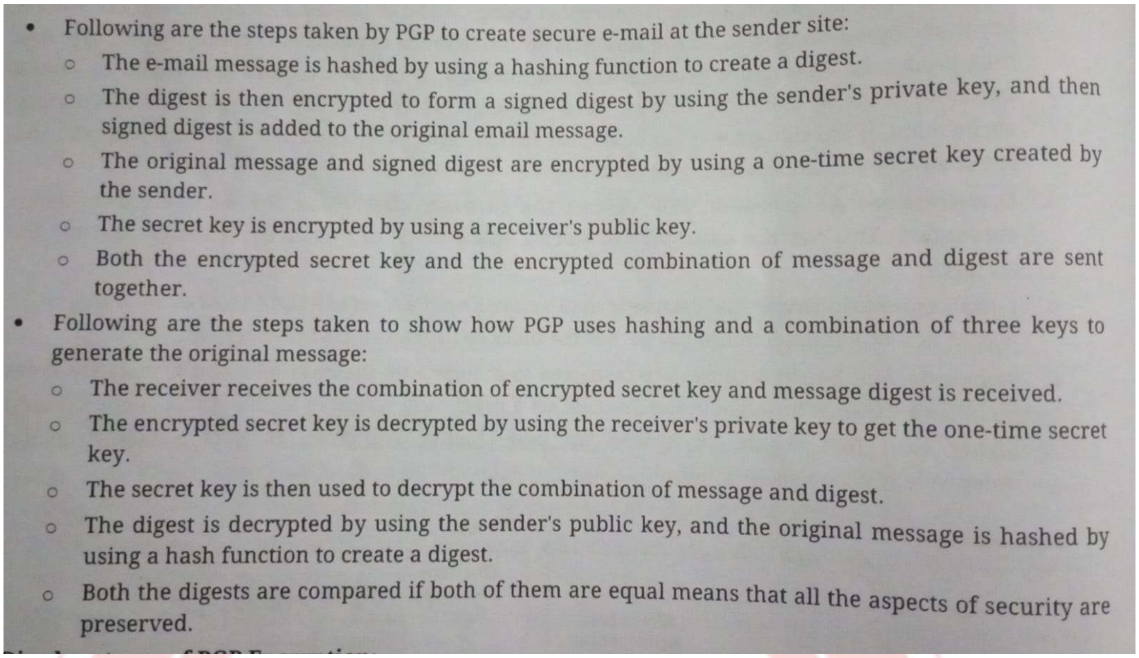
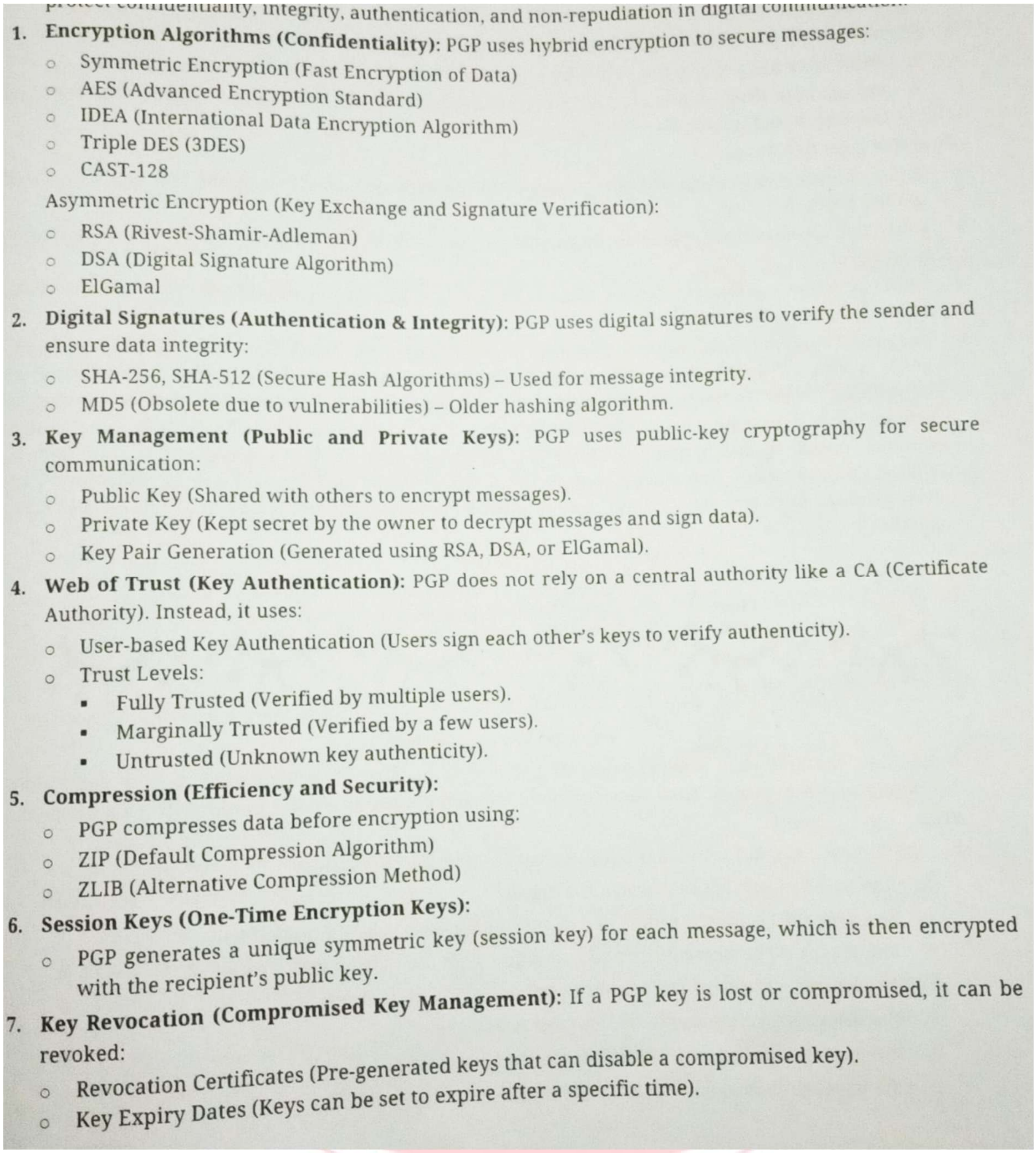


Fig. 4.53: PGP at Receiver Site



- Following are the steps taken by PGP to create secure e-mail at the sender site:
  - The e-mail message is hashed by using a hashing function to create a digest.
  - The digest is then encrypted to form a signed digest by using the sender's private key, and then signed digest is added to the original email message.
  - The original message and signed digest are encrypted by using a one-time secret key created by the sender.
  - The secret key is encrypted by using a receiver's public key.
  - Both the encrypted secret key and the encrypted combination of message and digest are sent together.
- Following are the steps taken to show how PGP uses hashing and a combination of three keys to generate the original message:
  - The receiver receives the combination of encrypted secret key and message digest is received.
  - The encrypted secret key is decrypted by using the receiver's private key to get the one-time secret key.
  - The secret key is then used to decrypt the combination of message and digest.
  - The digest is decrypted by using the sender's public key, and the original message is hashed by using a hash function to create a digest.
  - Both the digests are compared if both of them are equal means that all the aspects of security are preserved.

Security Parameters Services



protect confidentiality, integrity, authentication, and non-repudiation in digital communication.

1. **Encryption Algorithms (Confidentiality):** PGP uses hybrid encryption to secure messages:
   - Symmetric Encryption (Fast Encryption of Data)
   - AES (Advanced Encryption Standard)
   - IDEA (International Data Encryption Algorithm)
   - Triple DES (3DES)
   - CAST-128

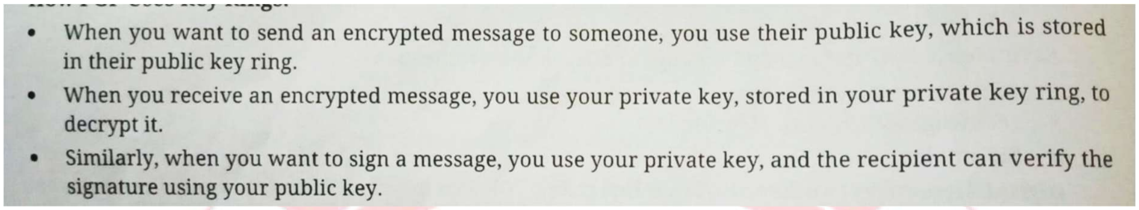
   Asymmetric Encryption (Key Exchange and Signature Verification):
   - RSA (Rivest-Shamir-Adleman)
   - DSA (Digital Signature Algorithm)
   - ElGamal
2. **Digital Signatures (Authentication & Integrity):** PGP uses digital signatures to verify the sender and ensure data integrity:
   - SHA-256, SHA-512 (Secure Hash Algorithms) – Used for message integrity.
   - MD5 (Obsolete due to vulnerabilities) – Older hashing algorithm.
3. **Key Management (Public and Private Keys):** PGP uses public-key cryptography for secure communication:
   - Public Key (Shared with others to encrypt messages).
   - Private Key (Kept secret by the owner to decrypt messages and sign data).
   - Key Pair Generation (Generated using RSA, DSA, or ElGamal).
4. **Web of Trust (Key Authentication):** PGP does not rely on a central authority like a CA (Certificate Authority). Instead, it uses:
   - User-based Key Authentication (Users sign each other's keys to verify authenticity).
   - Trust Levels:
     - Fully Trusted (Verified by multiple users).
     - Marginally Trusted (Verified by a few users).
     - Untrusted (Unknown key authenticity).
5. **Compression (Efficiency and Security):**
   - PGP compresses data before encryption using:
   - ZIP (Default Compression Algorithm)
   - ZLIB (Alternative Compression Method)
6. **Session Keys (One-Time Encryption Keys):**
   - PGP generates a unique symmetric key (session key) for each message, which is then encrypted with the recipient's public key.
7. **Key Revocation (Compromised Key Management):** If a PGP key is lost or compromised, it can be revoked:
   - Revocation Certificates (Pre-generated keys that can disable a compromised key).
   - Key Expiry Dates (Keys can be set to expire after a specific time).

PGP Key Rings

- **Key rings:** database that store cryptographic keys used for→ encryption, decryption, authentication
- It is a file that stores public or private keys
- Mainly 2 types of key rings:

1. **Public Key Ring:** stores public keys of other users and its own as well
2. **Private Key Ring:** stores users private key→ used for decryption and creating digital signature

How PGP Uses Key Rings



- When you want to send an encrypted message to someone, you use their public key, which is stored in their public key ring.
- When you receive an encrypted message, you use your private key, stored in your private key ring, to decrypt it.
- Similarly, when you want to sign a message, you use your private key, and the recipient can verify the signature using your public key.

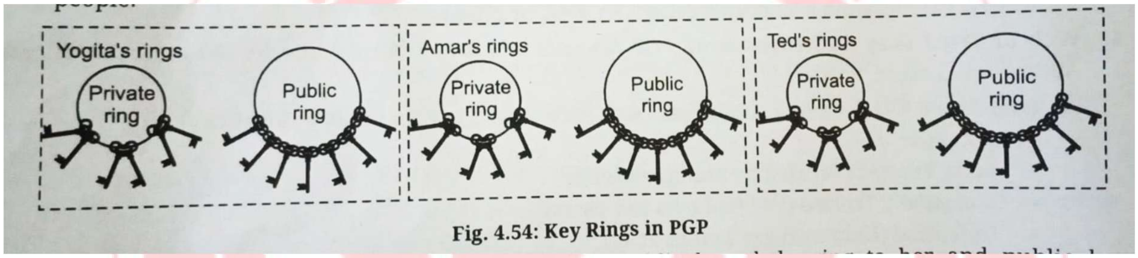
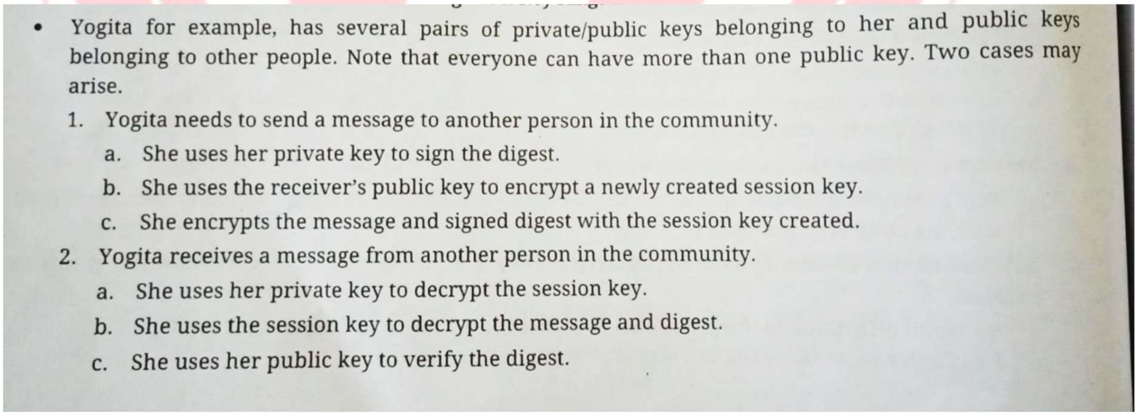


Fig. 4.54: Key Rings in PGP

- Each user has two sets of key rings: public and private
- They can use different combinations with different users



- Yogita for example, has several pairs of private/public keys belonging to her and public keys belonging to other people. Note that everyone can have more than one public key. Two cases may arise.
  1. Yogita needs to send a message to another person in the community.
     a. She uses her private key to sign the digest.
     b. She uses the receiver's public key to encrypt a newly created session key.
     c. She encrypts the message and signed digest with the session key created.
  2. Yogita receives a message from another person in the community.
     a. She uses her private key to decrypt the session key.
     b. She uses the session key to decrypt the message and digest.
     c. She uses her public key to verify the digest.

PGP Algorithms

- PGP defines set of symmetric-key and asymmetric key algorithms, cryptography hash function and compression method
- Key algorithms used in PGP:

1. Public Key Cryptography (e.g., RSA)

- Used for key exchange, encryption, decryption
- Each user has pair of keys→ public key: used by sender for encryption, private key: used by recipient for decryption

2. Symmetric Key Cryptography (e.g., 3DES, AES)

- One session key is established using public- key cryptography
- Symmetric key algorithm is used for encryption
- Efficient and faster

3. Hashing (e.g., SHA-1, SHA-256)

- Used to create fingerprints or message digest of data
- Then data is digitally signed
- Ensures message hasn't been tampered

4. Compression (e.g., ZIP)

- PGP optionally compresses the data before encryption to reduce size

5. Digital Signatures

- It is used to verify authenticity and integrity of message
- Sender signs message digest using their private key
- Receiver verifies signature using sender's public key

PGP Certificates

- It is a digitally signed document that contains a user's public key, identity information and trust indicators
- PGP certificates are used to associate a public key with identity of its owner

A PGP certificate includes:

1. **Public Key:** used for encryption and signature verification
2. **User Identity (Name, Email, etc.):** helps associate key with a real person
3. **Key ID and Fingerprint:** unique identifiers for public key
4. **Key Expiry Date:** optional, defines the key's validity period
5. **Digital Signature:** Key is assigned by key owner or others to verify authenticity
6. **Trust Level:** shows how much the key is trusted in the Web of Trust

How PGP Certificate works?

- **Key Generation:**
  - User generates public-private key pair
  - Public key is packaged into a PGP certificate
- **Certificate distribution**
  - Certificate is shared via email, key servers, personal exchange
- **Verification and Signing:**

  - Other users can sign the certificate to establish trust

- **Encryption and Authentication**
  - Public key certificate is used to encrypt message
  - It helps to verify digital signature

Applications of PGP certificates

1. **Email encryption:**

- Ensure secure communication by encrypting emails using verified public keys

2. **Digital Signatures:**

- Authenticate message and verify their integrity using certificate

3. **File encryption**

- Protect sensitive files by associating the with trusted public keys