UNIT-V
APPLICATION LAYER

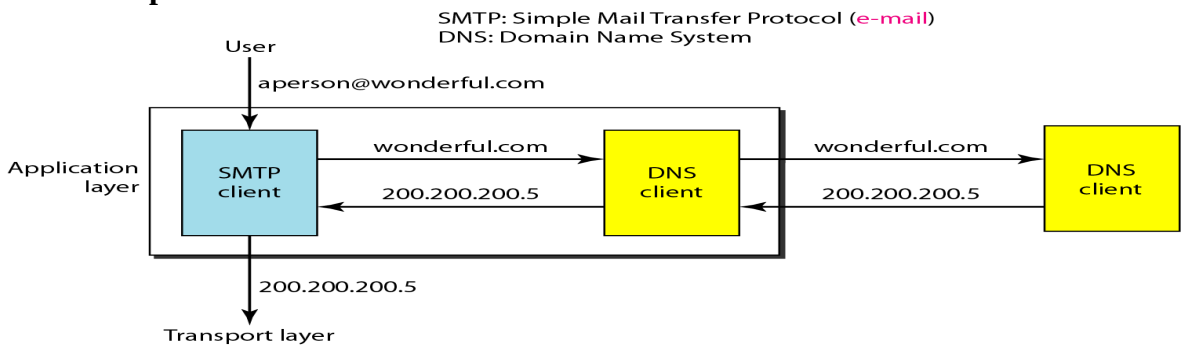
DOMAIN NAME SYSTEM

The client/server programs can be divided into two categories:

1. Programs that can be directly used by user.
2. Programs that support other application programs.

Domain Name System (DNS) is a supporting program that is used by other programs such as E-mail.

Basic concept of DNS



The above figure shows a DNS client/server program can support an E-mail program to find the IP address of an E-mail recipient.

- A user of an E-mail program knows the E-mail address of the recipient but the IP protocol needs the IP address.
- The DNS client program sends a request to a DNS server to map the E-mail address to the corresponding IP address.
- To identify an entity TCP/IP protocols uses the IP address, which uniquely identifies the connection of a host to the Internet.
- People prefer to use names instead of numeric addresses. Hence we need a system that can map a name to an address or an address to a name. DNS is designed for this purpose.

NAME SPACE

The names assigned to machines must be unique because the addresses are unique.

A name space that maps each address to a unique name can be organized in two ways:

- Flat Name Space
- Hierarchical Name Space

Flat Name Space

- In a flat name space, a name is assigned to an address.
- A name in this space is a sequence of characters without structure.

Disadvantage: Flat Name Space cannot be used in a large system such as the Internet because it must be centrally controlled to avoid ambiguity and duplication.

Hierarchical Name Space

- In Hierarchical name space, each name is made of several parts.
- The first part can define the nature of the organization, the second part can define the name of an organization, the third part can define departments in the organization etc.
- In this case, the authority to assign and control the name spaces can be decentralized.

- A central authority can assign the part of the name that defines the nature of the organization and the name of the organization only.
- The responsibility of the rest of the name can be given to the organization itself.
- The organization can add suffixes or prefixes to the name to define its host or resources.
- The management of the organization need not worry that the prefix chosen for a host is taken by another organization because, even if part of an address is the same, the whole address is different.

Example:

Assume three Education institutions named one of their computers **Challenger**.

The three colleges have given names by the central authority such as iitm.ac.in, berkeley.edu and smart.edu.

When these organizations add the name **Challenger** the names will be :

- challenger.iitm.ac.in
- challenger.berkeley.edu
- challenger.smart.edu

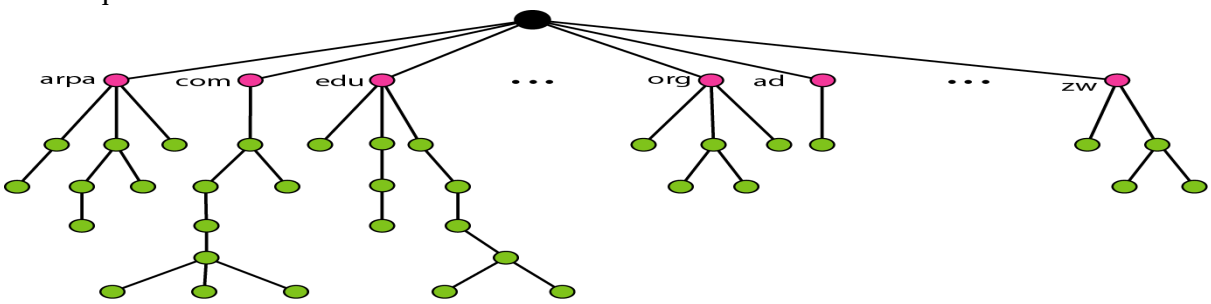
DOMAIN NAME SPACE

A domain name space was designed to have a Hierarchical Name Space.

In this design the names are defined in a tree structure with the root at the top. The tree can have only 128 levels: level 0 (root) to level 127.

Label

- Each node in the tree has a label, which is a string with a maximum of 63 characters. The root label is a null string (empty string).
- DNS requires that children of a node have different labels, which guarantees the uniqueness of the domain names.



Domain Name

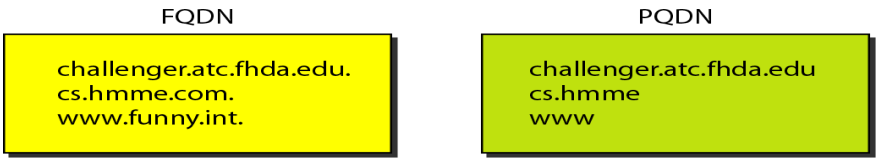
- Each node in the tree has a domain name. A full domain name is a sequence of labels separated by dots (.).
- Domain names are always read from the bottom to top.
- The last label is the label of the root (null). (i.e.) a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing.

Fully Qualified Domain Name (FQDN)

- If a label is terminated by a null string, it is called a fully qualified domain name (FQDN).
- An FQDN is a domain name that contains the full name of a host.
- It contains all labels that uniquely define the name of the host.

- A DNS server can only match an FQDN to an address.

Example: **challenger.atc.fhda.edu.**



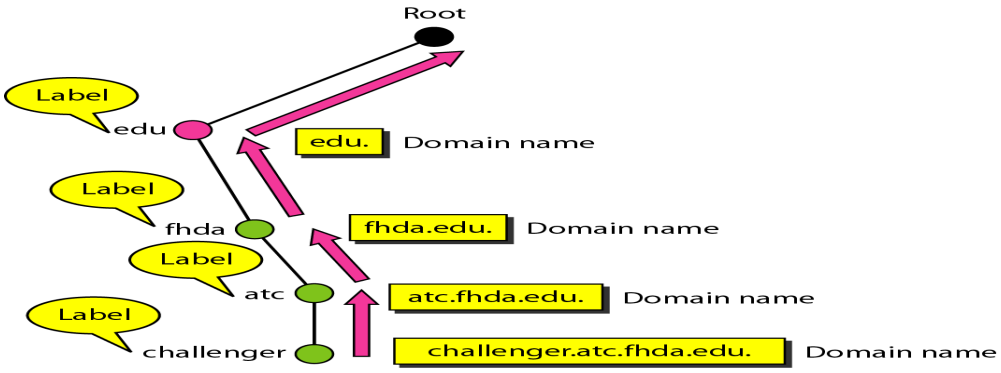
Partially Qualified Domain Name (PQDN)

- If a label is not terminated by a null string, it is called a PQDN.
- A PQDN starts from a node, but it does not reach the root.
- It is used when the name to be resolved belongs to the same site as the client.
- Here the resolver can supply the missing part, called the suffix, to create an FQDN.

Example: **challenger**

If a user at the **"fhda.edu."** site wants to get the IP address of the challenger computer, he or she can define the partial name "**challenger**".

The DNS client adds the suffix **"atc.fhda.edu."** before passing the address to the DNS server.



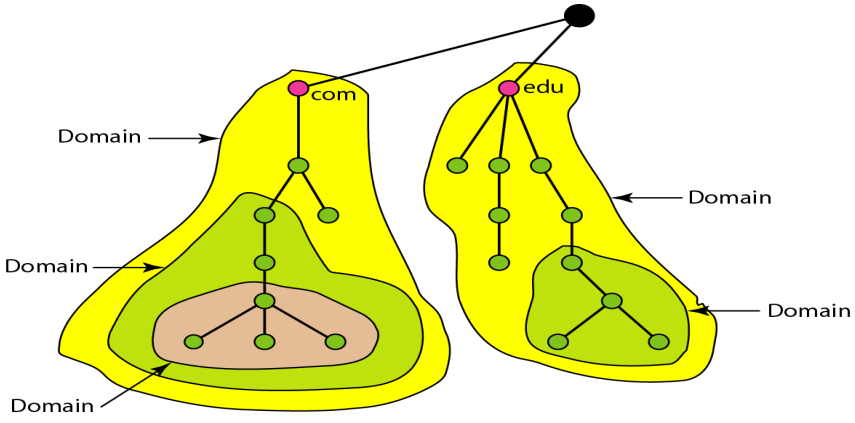
Note: The DNS client normally holds a list of suffixes such as:

- **atc.fhda.edu**
- **fhda.edu**
- **null**

These suffixes were added when the user defines an FQDN.

Domain

A **domain** is a sub-tree of the domain name space. The name of the domain is the domain name of the node at the top of the subtree. A domain may itself be divided into sub-domains.

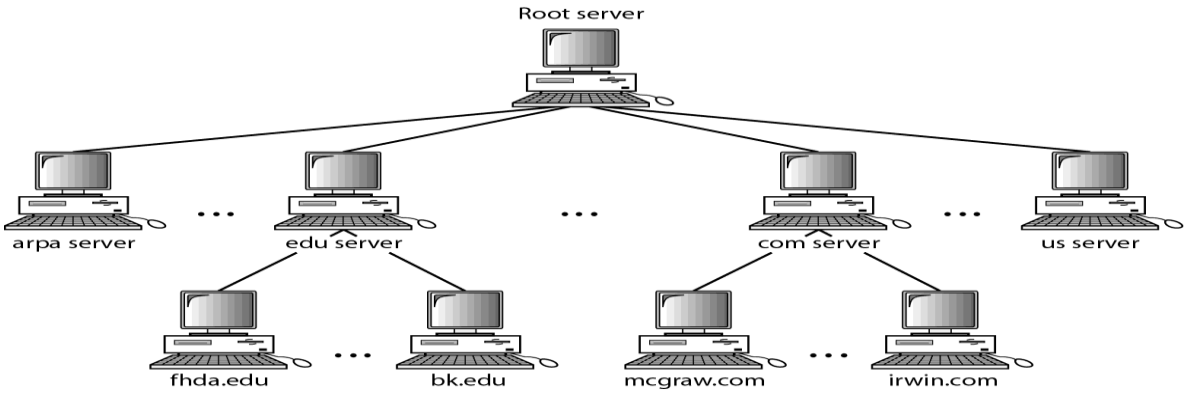


DISTRIBUTION OF NAME SPACE

- The information contained in the domain name space must be stored.
- This information is distributed among different computers and in different places.
- It is very inefficient and also unreliable to have just one computer store such a huge amount of information at one computer in one place.

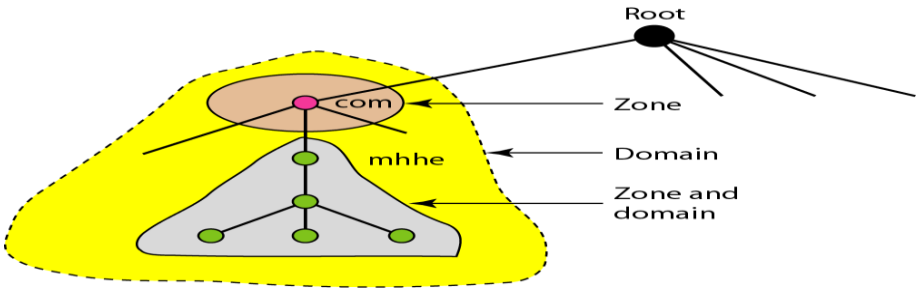
Hierarchy of Name Servers

- Distribution of the information among many computers called DNS servers.
- The whole space is divided into many domains based on the first level. The root stand alone and create as many domains (subtrees).
- A domain created in this way could be very large; DNS allows domains to be divided further into smaller domains (subdomains).
- Each server can be responsible (authoritative) for either a large or a small domain.



Zone

- The complete domain name hierarchy cannot be stored on a single server, it is divided among many servers.
- A zone is a contiguous part of the entire tree and it defines what a server is responsible for or server has authority over.



Case 1: When Domain is same as Zone.
If a server accepts responsibility for a domain and does not divide the domain into smaller domains, the domain and the zone refer to the same thing. The server makes a database called a zone file and keeps all the information for every node under that domain.

Case 2: When Domain and Zone are different.
If a server divides its domain into subdomains and delegates part of its authority to other servers, domain and zone refer to different things. The information about the nodes in the subdomains is stored in the servers at the lower levels, with the original server keeping some sort of reference to these lower-level servers.

Root Server

- A root server is a server whose zone consists of the whole tree.
- A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers.
- The servers are distributed all around the world.

DNS defines two types of servers: Primary and Secondary servers.

Primary Server

- A primary server is a server that 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 the zone file on a local disk.

Secondary Server

- A secondary server is a server that transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk.
- The secondary server neither creates nor updates the zone files.
- If updating is required, it must be done by the primary server, which sends the updated version to the secondary.

Note:

i. These secondary servers are used for crash recovery. These are redundant servers which are created when one server fails the other server can continue serving the clients for their data requests.

ii. A server can be a primary server for a specific zone and a secondary server for another zone.

Zone transfer

A primary server loads all information from the disk file. The secondary server loads all information from the primary server. When the secondary server downloads information from the primary server it is called zone transfer.

DNS IN THE INTERNET

In the Internet the domain name space tree is divided into three different sections:

- Generic Domains,
- Country Domains
- The Inverse Domain

Generic Domains

- The **generic domains** define Registered hosts according to their generic behavior.
- Each node in the tree defines a domain, which is an index to the domain name space database.

Generic domain labels are listed as:

Label	Description
com	Commercial organizations
org	Nonprofit organizations
net	Network support centers
edu	Educational institutions
gov	Government institutions

Country Domains

- The country domains section uses two-character country abbreviations.
  Ex: in for India, us for USA.
- Second labels can be organizational, or they can be more specific, national designations.
  Ex: .ac.in for nptel.ac.in, .gov.in for tspsc.gov.in etc.

Inverse Domain

The inverse domain is used to map an Address to a Name. It uses inverse query or pointer query.

Inverse Query

When a server has received a request from a client to do a task, the server has a file that contains a list of authorized clients and their received IP addresses are listed. The server asks its resolver to send a query to the DNS server to map an address to a name to determine if the client is on the authorized list. This type of query is called an Inverse Query or Pointer Query.

Example: An IP address such as 132.34.45.121 is read as 121.45.34.132.in-addr. arpa.

- The servers that handle the inverse domain are also hierarchical.
- To handle a pointer query, the inverse domain is added to the domain name space with the first-level node called **arpa**.
- The second level is also one single node named **in-addr** for inverse address.
- The rest of the domain defines IP addresses.

RESOLUTION

Mapping a name to an address or an address to a name is called Name-Address Resolution.

Resolver

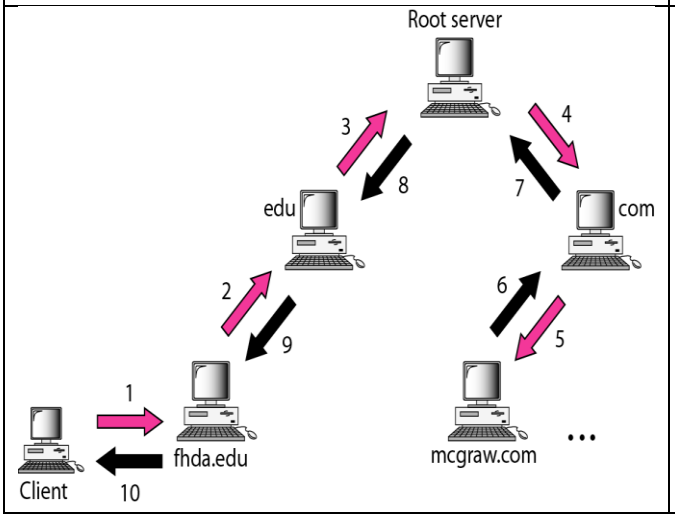
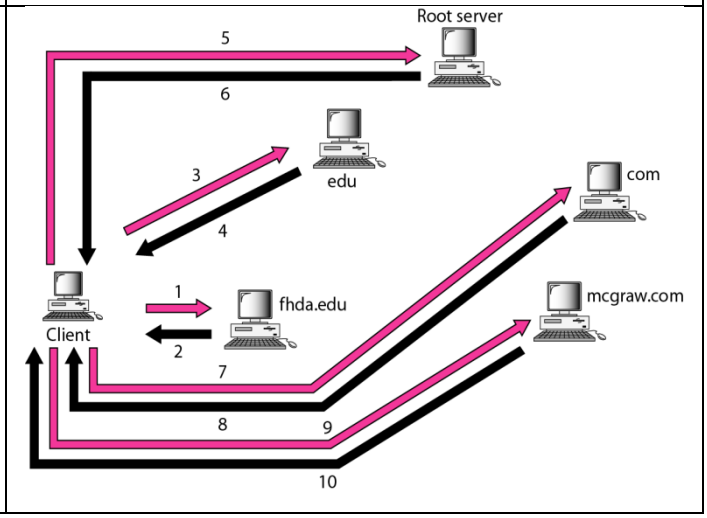
- DNS is designed as a client/server application. A host that needs to map an address to a name or a name to an address calls a DNS client called a **Resolver**.
- The resolver accesses the closest DNS server with a mapping request.
- If the server has the information, it sends the information to resolver.
- If server does not have the information it either refers the resolver to other servers or asks other servers to provide the information.
- After the resolver receives the mapping, it interprets the response to see if it is a real resolution or an error and finally delivers the result to the requested host process.

Recursive Resolution

- The client (resolver) can ask for a recursive answer from a name server. This means that the resolver expects the server to supply the final answer.
- If the server is the authority for the domain name, it checks its database and responds.
- If the server is not the authority, it sends the request to another server (the parent) and waits for the response.
- If the parent is the authority, it responds; otherwise, it sends the query to yet another server.
- When the query is finally resolved, the response travels back until it finally reaches the requesting client.
- This process is called **Recursive Resolution**.

Iterative Resolution

- The client repeats the same query to multiple servers.
- If the client does not ask for a recursive answer, the mapping can be done iteratively.
- If the server is an authority for the name, it sends the answer.
- If the server is not an authority it returns the IP address of the server to the client, that the server thinks can resolve the query.
- The client is responsible for repeating the query to this second server.
- If the newly addressed server can resolve the problem, it answers the query with the IP address; otherwise, it returns the IP address of a new server to the client.
- Now the client must send the query to the third server. This process is called Iterative Resolution.

Recursive Resolution	Iterative Resolution
	

Caching

- When a server asks for a mapping from another server and receives the response, it stores this information in its cache memory before sending it to the client.
- If the client asks for the same mapping, it can check its cache memory and returns the result.
- To inform the client that the response is coming from the cache memory and not from an authoritative source, the server marks the response as Unauthoritative.

DNS MESSAGES

- DNS has two types of messages: Query and Response. Both types have the same format.
- The Query Message consists of a **Header** and **Question Records**;
- The response message consists of a **Header**, **Question Records**, **Answer records**, **Authoritative Records**, and **Additional Records**.

Header (12 bytes)

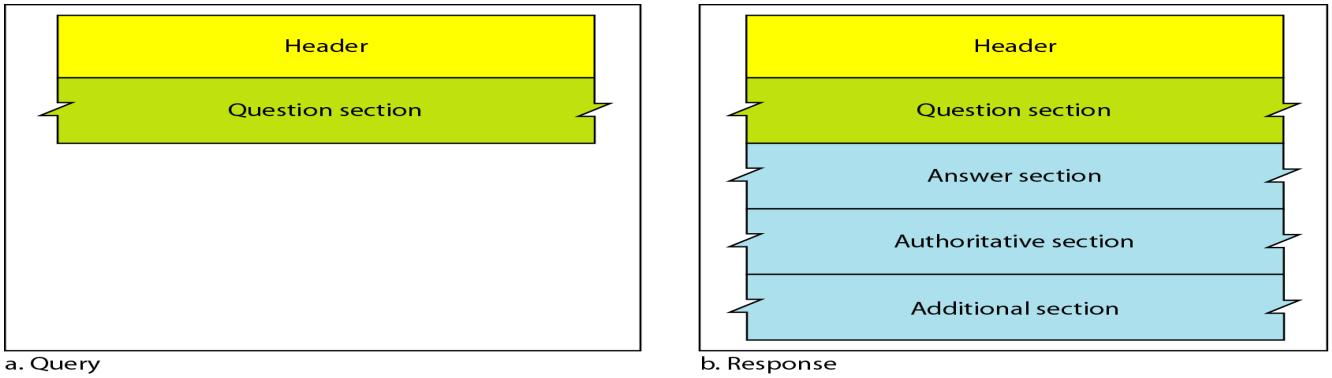
Both query and response messages have the same header format with some fields set to zero for the query messages. Header format and fields are given below:

Identification	Flags
Number of question records	Number of answer records (all 0's in query message)

Number of authoritative records (all 0's in query message)	Number of additional records (all 0's in query message)

- **Identification** subfield is used by the client to match the response with the query. The client uses a different identification number each time it sends a query.
- **Flags** subfield is a collection of subfields that define the type of the message, the type of answer requested, the type of desired resolution (recursive or iterative), and so on.
- **Number of Question Records** subfield contains the number of queries in the question section of the message.
- **Number of Answer Records** subfield contains the number of answer records in the answer section of the response message. Its value is zero in the query message.
- **Number of Authoritative Records** subfield contains the number of authoritative records in the authoritative section of a response message. Its value is zero in the query message.
- **Number of Additional Records** subfield contains the number additional records in the additional section of a response message. Its value is zero in the query message.

Query and Response Messages



a. Query b. Response

- **Question Section** consisting of one or more question records. It is present on both query and response messages.
- **Answer Section** includes the answer from the server to the client (resolver). It is present only on response messages.
- **Authoritative Section** gives information (domain name) about one or more authoritative servers for the query. It is present only on response messages.
- **Additional Information Section** provides additional information that helps the resolver.

Example: A server may give the domain name of an authoritative server to the resolver in the authoritative section, and include the IP address of the same authoritative server in the additional information section.

TYPES OF RECORDS

Two types of records are used in DNS.

- **Question Record**
  It is used by the client to get information from a server. This contains the domain name.
- **Resource Record**
  Each domain name (each node on the tree) is associated with a record called the resource record. The server database consists of resource records. Servers sends resource records to client.

REGISTRARS

Registrar adds new domains to DNS. A registrar first verifies that the requested domain name is unique and then enters it into the DNS database. The registrars and their names, addresses found at: http://www.intenic.net

Example: Domain name: WS.wonderful.com (ws is a server name)
IP address: 200.200.200.5 (new IP address).

DYNAMIC DOMAIN NAME SYSTEM (DDNS)

Need for DDNS?

- In DNS, when there is a change, such as adding a new host, removing a host, or changing an IP address, the change must be made to the DNS master file.
- These types of changes involve a lot of manual updating. Today's internet is not suitable for manual updations.
- Hence we have to update the master file dynamically.

DDNS process

- In DDNS, when a binding between a name and an address is determined the information is sent usually by DHCP to a primary DNS server.
- The primary server updates the zone.
- The secondary servers are notified either actively or passively.
- In active notification, the primary server sends a message to the secondary servers about the change in the zone, whereas in passive notification the secondary servers periodically check for any changes.
- In either case, after being notified about the change, the secondary requests information about the entire zone (zone transfer).
- To provide security and prevent unauthorized changes in the DNS records, DDNS can use an authentication mechanism.

REMOTE LOGGING

In the Internet, users may want to run application programs at a remote site and create results that can be transferred to their local site.

Example: Students may want to connect to their university computer lab from their home to access application programs for doing homework assignments or projects.

A General purpose client/server program that allows a user to log-on to a remote computer to access any application program on that remote computer.

After logging on, a user can use the available services on the remote computer and transfer the results back to the local computer.

TELNET (TErminaL NETwork)

TELNET is a client/server application program. It is the standard TCP/IP protocol for virtual terminal service as proposed by the International Organization for Standards (ISO).

TELNET enables the establishment of a connection to a remote system in such a way that the local terminal appears to be a terminal at the remote system.

Timesharing Environment

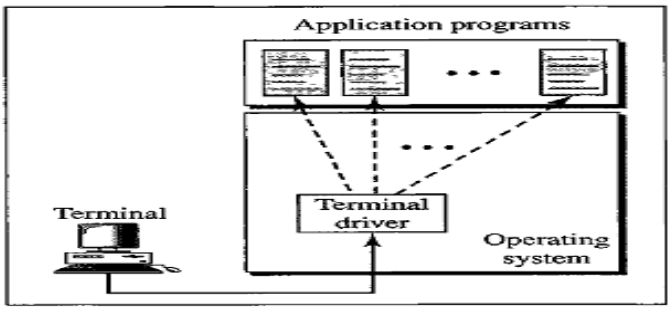
TELNET works in Time Sharing Environment. The interaction between a user and the computer occurs through a terminal.

Logging

- In a timesharing environment, users are part of the system with some right to access resources. Each authorized user has Identification (UserID) and a password.
- To access the system the user logs into the system with a user id or log-in name.
- The system also includes password checking to prevent an unauthorized user from accessing the resources.

Local log-in

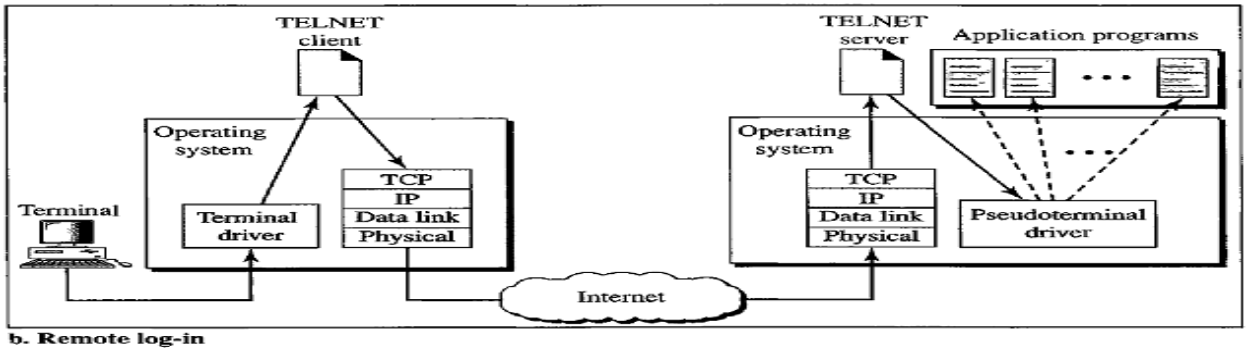
- When a user logs into a local timesharing system it is called local log-in.
- As a user types at a terminal the keystrokes are accepted by the terminal driver. The terminal driver passes the characters to the operating system.
- The operating system interprets the combination of characters and invokes the desired application program or utility.



a. Local log-in

Remote Log-in

- When a user wants to access an application program or utility located on a remote machine, the user performs remote log-in. TELNET uses client and server programs.
- The user sends the keystrokes to the terminal driver, where the local operating system accepts the characters but does not interpret them.
- The characters are sent to the TELNET client, which transforms the characters to a universal character set called Network Virtual Terminal (NVT) characters and delivers them to the local TCP/IP protocol stack.
- The commands or text in NVT form travel through the Internet and arrive at the TCP/IP stack at the remote machine.



b. Remote log-in

- The characters are delivered to the operating system and passed to the TELNET server, which changes the characters to the corresponding characters understandable by the remote computer.
- The characters cannot be passed directly to the operating system because the remote operating system is not designed to receive characters from a TELNET server. It is designed to receive characters from a terminal driver.
- Hence the characters can be passed to **Pseudo-terminal driver** which passes the characters to operating system.
- The operating system then passes the characters to the appropriate application program.

Network Virtual Terminal (NVT)

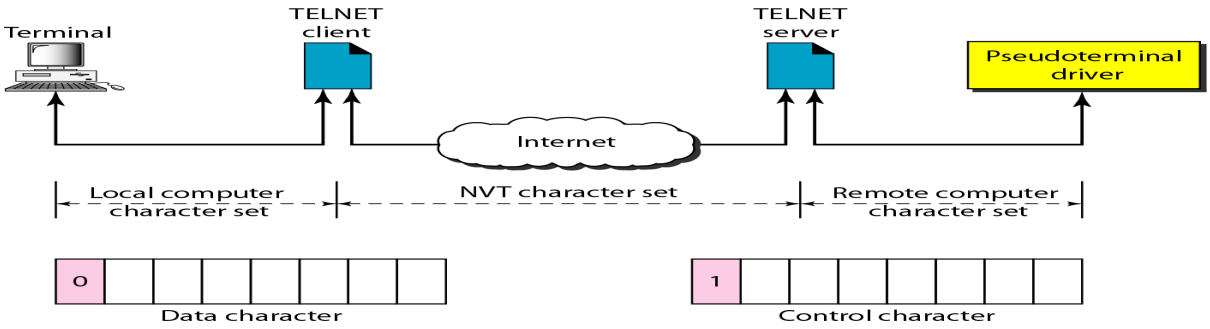
- TELNET defines a universal interface called the Network Virtual Terminal character set. Via this interface the client TELNET translates characters (data or commands) that come from the local terminal into NVT form and delivers them to the network.
- The server TELNET translates data and commands from NVT form into the form acceptable by the remote computer.

NVT Character Set

NVT uses two sets of characters, one for data and the other for control. Both are 8-bit.

- For data, NVT is an 8-bit character set in which the highest-order bit is 0 and the 7 lowest-order bits are the same as ASCII.
- To send control characters between computers (i.e.) from client to server or server to client NVT uses an 8-bit character set in which the highest-order bit is set to l.

Examples of NVT control characters such as **EOF** – End of File, **EOR**- End of Record, **NOP**- No operation etc.



Embedding

- TELNET uses only one TCP connection. The same connection is used for sending both data and control characters. TELNET accomplishes this by embedding the control characters in the data stream.
- To distinguish data from control characters, each sequence of control characters is preceded by a special control character called **Interpret As Control** (IAC).
- For TELNET, server uses the well-known port 23 and the client uses an ephemeral port.

Example: a user wants a server to display a file (filel) on a remote server. User can type

Cat file1

Suppose the name of the file has been mistyped (filea instead of file1). The user uses the backspace key to correct this situation.

Cat filea <backspace>1

In the default implementation of TELNET the user cannot edit locally. The editing is done at the remote server.
The backspace character is translated into two remote characters (lAC EC), which are embedded in the data and sent to the remote server.

C	**A**	**t**		**f**	**i**	**l**	**e**	**a**	**IAC**	**EC**	**1**

Options
Options are extra features available to a user with a more sophisticated terminal. TELNET allows the client and server negotiate options before or during the use of the service.

Option Negotiation
Option negotiation is done between the client and the server to use any of the options. Four control characters are used for this purpose: WILL, WONT, DO, DON'T.

A party can offer to enable or disable an option if it has the right to do so. The offering can be approved or disapproved by the other party.

- To offer enabling, the offering party sends the **WILL** command, which means "Will I enable the option?"
- The other party sends either the **DO** command, which means "Please do," or the **DONT** command, which means "Please don't."
- To offer disabling, the offering party sends the **WONT** command, which means "I won't use this option any more."
- The answer must be the DONT command, which means "Don't use it anymore."

Suboption Negotiation
Some options require additional information such as defining type of terminal or speed of the terminal etc.
There are two suboption characters are defined: SB-Suboption begin, SE- Suboption End

Mode of Operation
Most TELNET implementations operate in one of three modes: **Default mode**, **Character mode** or **Line mode**.

Default Mode

- The default mode is used if no other modes are invoked through option negotiation.
- In this mode the echoing is done by the client.
- The user types a character and the client echoes the character on the screen (or printer) but does not send it until a whole line is completed.

Character Mode

- In the character mode each character typed is sent by the client to the server.
- The server normally echoes the character back to be displayed on the client screen.
- In this mode the echoing of the character can be delayed if the transmission time is long such as in a satellite connection.
- It also creates overhead (traffic) for the network because three TCP segments must be sent for each character of data.

Line Mode

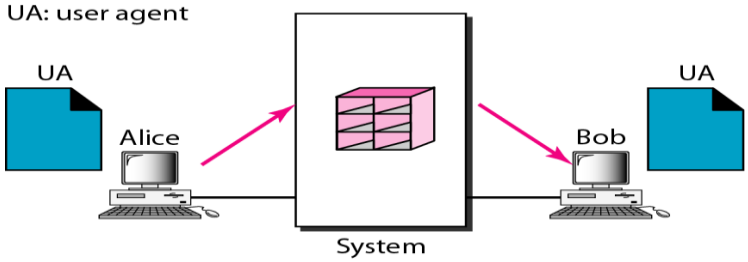
- Line mode has been proposed to compensate for the deficiencies of the default mode and the character mode.
- In line mode, line editing (echoing, character erasing, line erasing, and so on) is done by the client. The client then sends the whole line to the server.

ELECTRONIC MAIL

Electronic mail (E-mail) is one of the most popular Internet services. E-mail allows a message to include text, audio, and video. There are Four Scenarios of E-mail:

First Scenario

- In the first scenario, the sender and the receiver of the E-mail are user application programs on the same system. They are directly connected to a shared system.
- The administrator has created one mailbox for each user where the received messages are stored.
- A mailbox is part of a local hard drive and it a special file with permission restrictions. Only the owner of the mailbox has access to it.

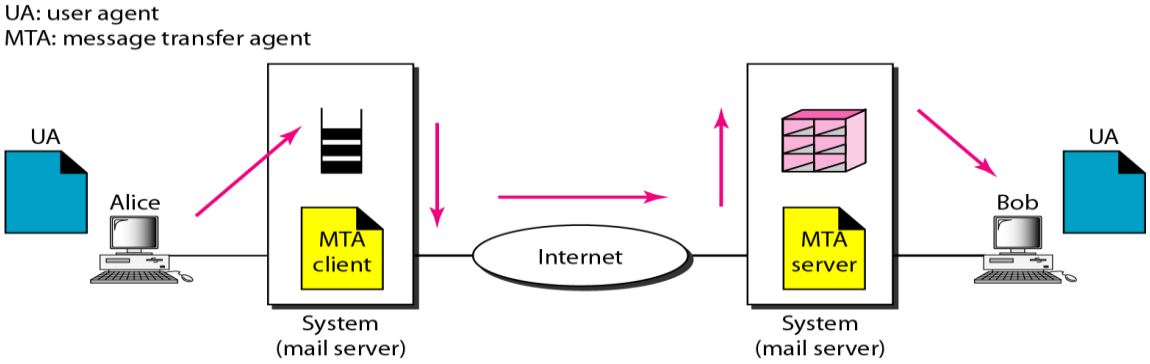


Example: Consider the above figure, Alice and Bob are two users of mail server.

- When a user Alice needs to send a message to Bob, Alice runs a User Agent (UA) program to prepare the message and store it in Bob's mailbox.
- The message has the sender and recipient mailbox addresses (names of files).
- Bob can retrieve and read the contents of his mailbox using a User Agent.

Second Scenario

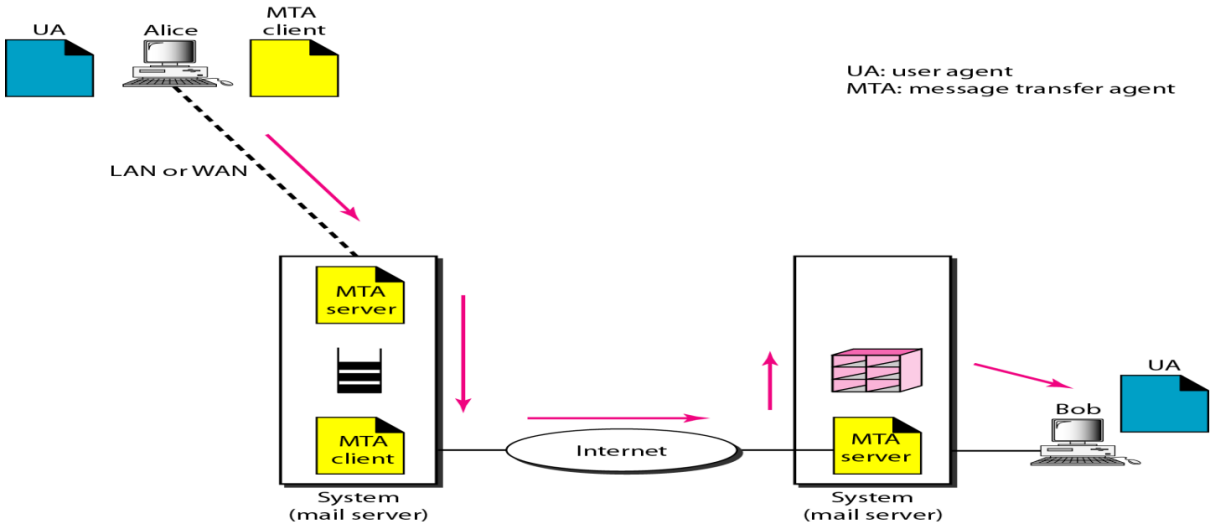
- In the second scenario, the sender and the receiver of the E-mail are user application programs on two different systems. The message needs to be sent over the Internet.
- We need User Agents (UAs) and Message Transfer Agents (MTA's).



- Alice needs to use a user agent program to send her message to the mail server at her own site.
- The mail server at Alice site uses a queue to store messages waiting to be sent.

- Bob also needs a user agent program to retrieve messages stored in the mailbox of the system at his site.
- The message needs to be sent through the Internet from Alice's site to Bob's site. Here two Message Transfer Agents are needed: one for client and one for server.
- Most client/server programs on the Internet, the server needs to run all the time because it does not know when a client will ask for a connection.
- The client can be alerted by the system when there is a message in the queue to be sent.

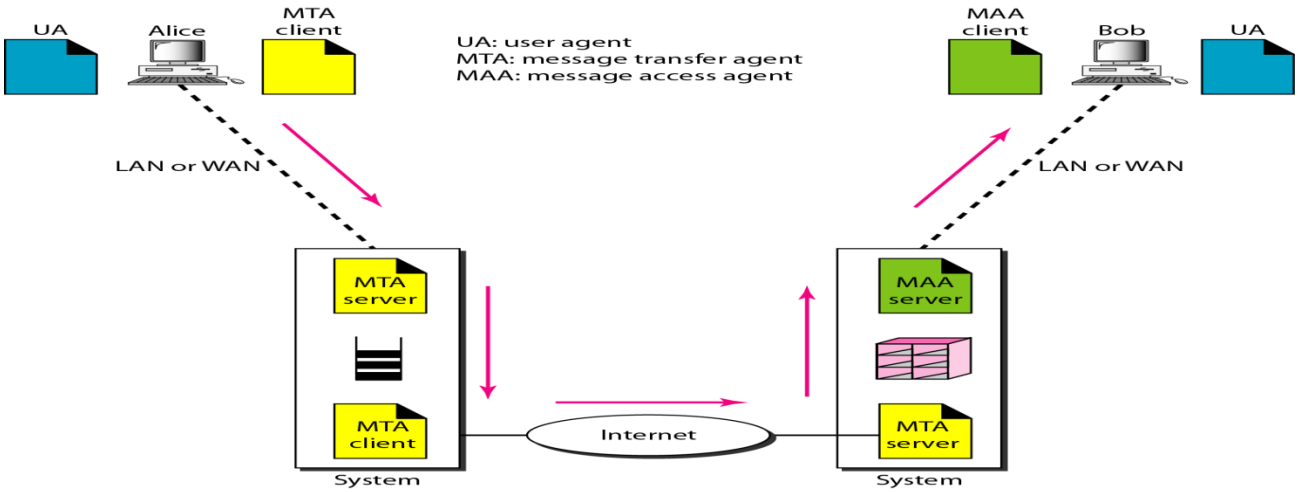
Third Scenario



- In the third scenario, Bob is directly connected to his system (i.e. Mail Server). Alice is separated from her system.
- Alice is connected to the mail server via WAN or LAN.
- In an organization that uses one mail server for handling E-mails, all users need to send their messages to this mail server.
- User agent of Alice prepares message and sends the message through the LAN or WAN.
- Whenever Alice has a message to send, Alice calls the user agent and user agent calls the MTA client.
- The MTA client establishes a connection with the MTA server on the system.
- The system at Alice's site queues all messages received. It then uses an MTA client to send the messages to the system at Bob's site. The system receives the message and stores it in Bob's mailbox.
- Bob uses his user agent to retrieve the message and reads it.
- It needs two MTA client and two MTA server programs.

Fourth Scenario

- It is the most common scenario, Alice and Bob both are connected to their mail server by a WAN or a LAN.
- After the message has arrived at Bob's mail server, Bob needs to retrieve it. Now Bob needs another set of client/server agents called Message Access Agents (MAA). Bob uses an MAA client to retrieve his messages.
- The client sends a request to the MAA server and requests the transfer of the messages.



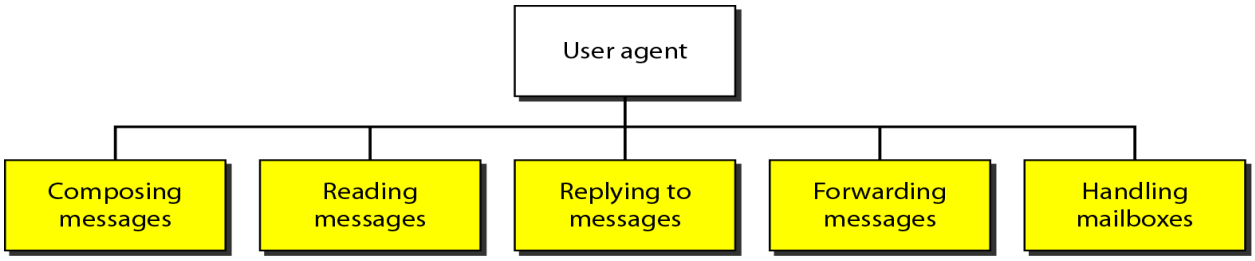
Architecture of E-mail

There are three major components in the architecture of E-mail:

1. User Agent
2. Message Transfer Agent
3. Message Access Agent

User Agent

User Agent provides services to the user to make the process of sending and receiving a message easier. Services provided by User agent are:



Composing Messages

A user agent helps the user to compose the E-mail message to be sent out. Most user agents provide a template on the screen to be filled in by the user.

Reading Messages

The user agent reads the incoming messages. When a user invokes a user agent, it first checks the mail in the incoming mailbox. Each E-mail contains the following fields:

1. A number field.
2. A flag field that shows the status of the mail such as new, already read but not replied to, or read and replied to.
3. The size of the message.
4. The sender.
5. The optional subject field.

Replying to Messages

After reading a message, a user can use the user agent to reply to a message. A user agent allows the user to reply to the original sender or to reply to all recipients of the message.

Forwarding Messages

It means sending a message to a third party. A user agent allows receiver to forward message.

Handling Mailboxes

- A user agent normally creates two mailboxes: **Inbox** and **Outbox**.
- Inbox and Outbox is a file with a special format that can be handled by user agent.
- **Inbox** keeps all the received E-mails until they are deleted by the user.
- **Outbox** keeps all the sent E-mails until the user deletes them.

User Agent Types

There are two types of user agents: **Command-driven** and **GUI-based**.

Command-Driven User Agent

- Command Driven user agents present as the underlying user agents in servers.
- It normally accepts a one-character command from the keyboard to perform its task.

Examples: mail, pine, and elm.

GUI-Based User Agents

- Modern user agents are GUI-based. They contain Graphical-User Interface (GUI) components that allow the user to interact with the software by using both the keyboard and the mouse.
- They have graphical components such as icons, menu bars, and windows that make the services easy to access.

Example: Eudora, Microsoft's Outlook, and Netscape.

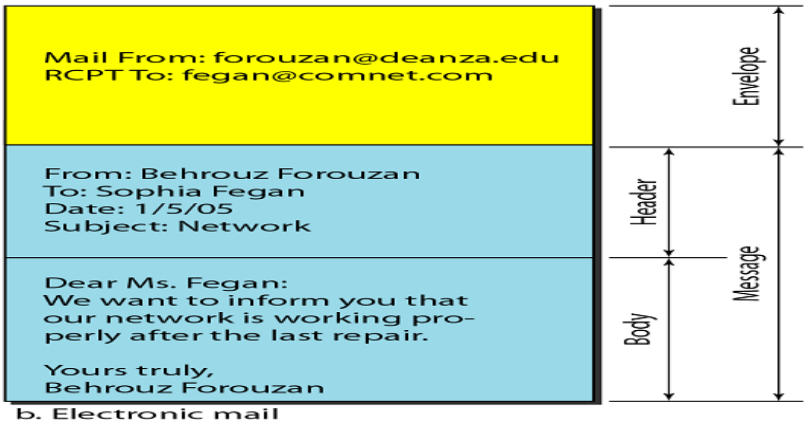
Sending Mail

A user E-mail has an Envelope and a Message.

Envelope usually contains the sender and the receiver addresses.

Message

- Message contains the Header and Body.
- Header of the message defines the sender, the receiver, the subject of the message, and some other information such as encoding type.
- Body of the message contains the actual information to be read by the recipient.



b. Electronic mail

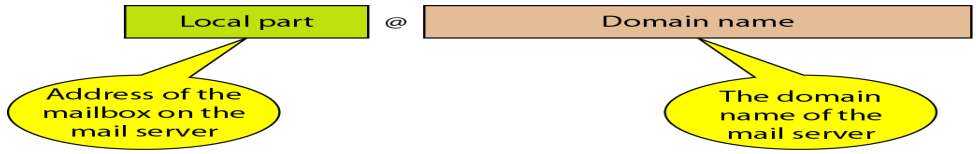
Receiving Mail

The user agent is triggered by the user or a timer. If a user has mail, the User Agent informs the user with a notice.

If the user is ready to read the mail, a list is displayed in which each line contains a summary of the information about a particular message in the mailbox.

Addresses

In the Internet, the address consists of two parts: a local part and a domain name separated by @ symbol.



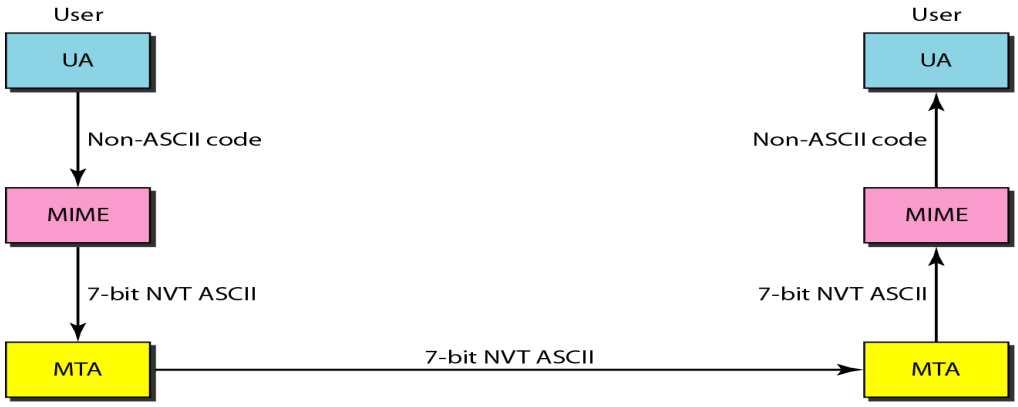
- **Local part** defines the address of the mailbox on the mail server. All the mail received for a user is stored for retrieval by the message access agent.
- **Domain Name** The domain name assigned to each mail server either comes from the DNS database or is a logical name (i.e.) the name of the organization.

Mailing List

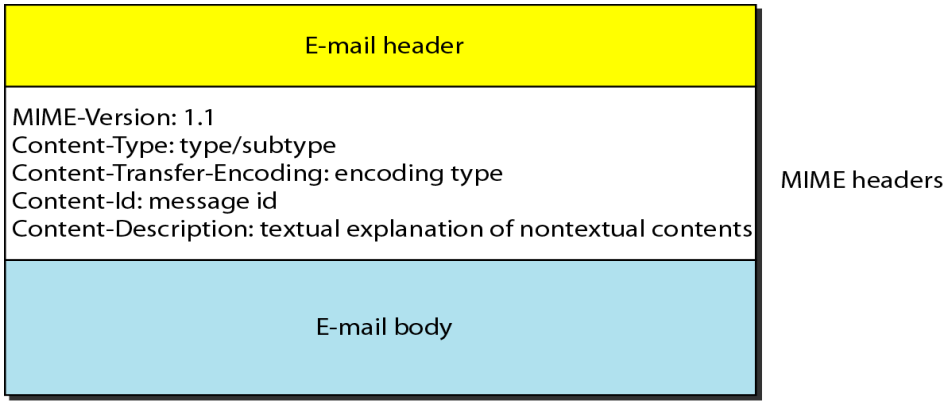
- Electronic mail allows one name (an alias) to represent several different E-mail addresses is called a mailing list.
- Every time a message is to be sent, the system checks the recipient's name against the alias database.

MIME (Multipurpose Internet Mail Extensions)

- MIME is a supplementary protocol that allows non-ASCII data to be sent through E-mail.
- French, German, Hebrew, Russian, Chinese, and Japanese are non-ASCII characters.
- MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers them to the client MTA to be sent through the Internet.
- The message at the receiving side is transformed back to the original data.



MIME defines five headers that can be added to the original E-mail header section to define the transformation parameters:



- **MIME-Version** header defines the version of MIME used. The current version is 1.1.
- **Content-Type** header defines the type of data used in the body of the message. Depending on the subtype the header may contain other parameters.
  MIME allows seven different types of data: Text, Multipart, Message, Image, Video, Audio and Application.
- **Content-Transfer-Encoding** header defines the method used to encode the messages into 0s and 1s for transport.
- **Content-Id** header uniquely identifies the whole message in a multiple-message environment.
- **Content-Description** header defines whether the body is image, audio or video.

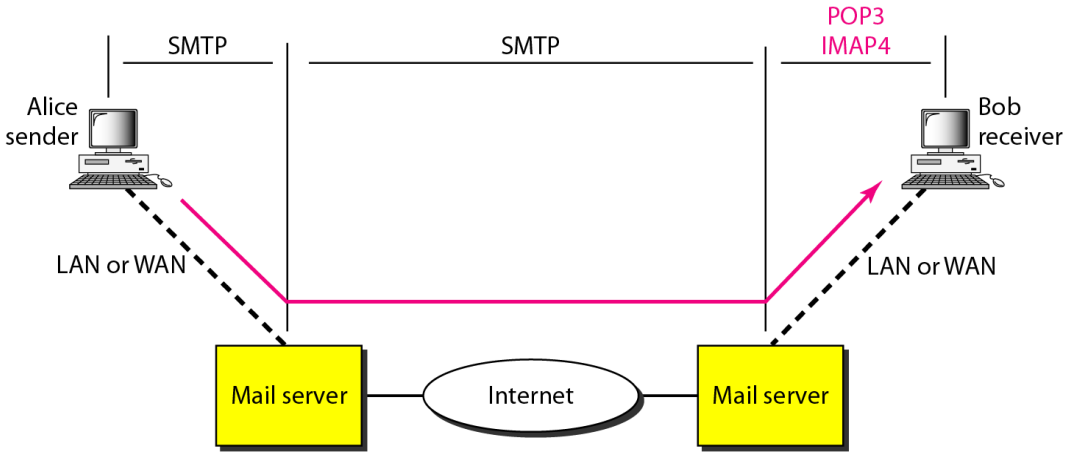
SIMPLE MAIL TRANSFER PROTOCOL (SMTP)

Message Transfer Agent: SMTP

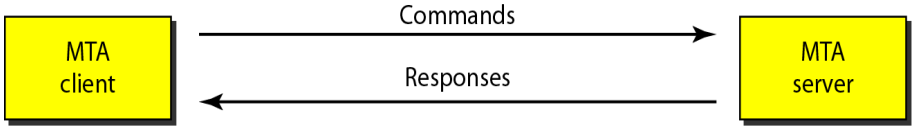
- The actual mail transfer is done through message transfer agents (MTA).
- Simple Mail Transfer Protocol defines the MTA Client and MTA server in the internet.
- MTA Client is used to send mail and MTA Server is used to receive a mail.

SMTP is used two times:

1. Between sender and sender mail server.
2. Between sender mail server and receiver mail server.



SMTP uses commands and responses to transfer messages between an MTA client and an MTA server. Each command or reply is terminated by a two-character end-of-line token.



Commands

Commands are sent from the client to the server. It consists of a keyword followed by zero or more arguments.

There are five mandatory commands are used. Every implementation must support these five commands: HELO (Sender's Host name), MAIL FROM, RCPT TO, DATA, QUIT.

Responses

Responses are sent from the server to the client. A response is a three digit code that may be followed by additional textual information.

Responses are divided into four categories: The leftmost digit of the code 2, 3, 4, and 5 defines the category.

- 2-Positive Completion Reply
- 3-Positive Intermediate Reply
- 4-Transient Negative Completion Reply
- 5-Permanent Negative Completion Reply

Mail Transfer Phases

The process of transferring a mail message occurs in three phases:

- Connection Establishment
- Mail Transfer
- Connection Termination

Example or SMTP

Consider the below example that directly uses SMTP to send an E-mail and simulate the commands and responses. Well known port number for SMTP is 25.

- It uses TELNET to log into port 25.
- We then use the commands directly to send an E-mail.
- In this example, forouzanb@adelphia.net is sending an E-mail to himself.
- The first few lines show TELNET trying to connect to the Adelphia mail server.
- After connection, we can type the SMTP commands and then receive the responses.

$ telnet mail.adelphia.net 25
Trying 68.168.78.100 . . .
Connected to mail.adelphia.net (68.168.78.100).

```
================  Connection Establishment  ================
  220 mta13.adelphia.net SMTP server ready Fri, 6 Aug 2004 . . .
HELO mail.adelphia.net
  250 mta13.adelphia.net
```

```
==================       Mail Transfer       ================
MAIL FROM: forouzanb@adelphia.net
  250 Sender <forouzanb@adelphia.net> Ok
RCPT TO: forouzanb@adelphia.net
  250 Recipient <forouzanb@adelphia.net> Ok
DATA
  354 Ok Send data ending with <CRLF>.<CRLF>
From: Forouzan
TO: Forouzan

This is a test message
to show SMTP in action.
.
```

```
==================  Connection Termination  ==============
    250 Message received: adelphia.net@mail.adelphia.net
QUIT
    221 mta13.adelphia.net SMTP server closing connection
Connection closed by foreign host.
```

Message Access Agent: POP and IMAP

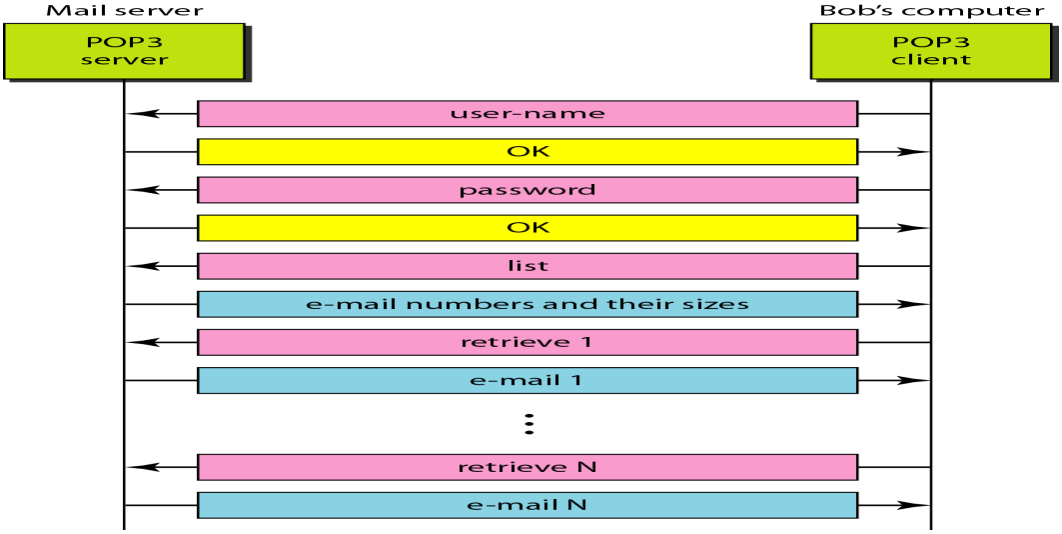
- The first and the second stages of mail delivery use SMTP.
- SMTP is not involved in the third stage because SMTP is a push protocol. It pushes the message from the client to the server.
- The third stage needs a pull protocol. The client must pull messages from the server. The direction of the bulk data is from the server to the client. The third stage uses a Message Access Agent.

There are two Message Access Protocols are available: POP3 and IMAP4

POP3 (Post office Protocol)

Post Office Protocol version 3 (POP3) is simple and limited in functionality.

- The client POP3 software is installed on the recipient computer.
- The server POP3 software is installed on the mail server.
- Mail access starts with the client when the user needs to download E-mail from the mailbox on the mail server.
- The client opens a connection to the server on TCP port 110. It then sends its user name and password to access the mailbox.
- The user can list and retrieve the mail messages one by one.



POP3 has two modes:

- **Delete Mode** The mail is deleted from the mailbox after each retrieval.
- **Keep Mode** The mail remains in the mailbox after retrieval.

Deficiencies of POP3

- POP3 does not allow the user to organize their mail on the server.
- The user cannot have different folders on the server.
- POP3 does not allow user to partially check the contents of the mail before downloading.

IMAP4 (Internet Mail Access Protocol-version 4)

IMAP4 is more powerful and more complex than POP3. It is implemented to overcome the deficiencies of POP3.

IMAP4 provides the following extra functions:

- A user can check the E-mail header prior to downloading.

- A user can search the contents of the E-mail for a specific string of characters prior to downloading.
- A user can partially download E-mail. This is especially useful if bandwidth is limited and the E-mail contains multimedia with high bandwidth requirements.
- A user can create, delete, or rename mailboxes on the mail server.
- A user can create a hierarchy of mailboxes in a folder for E-mail storage.

FILE TRANSFER PROTOCOL (FTP)

File Transfer Protocol (FTP) is the standard mechanism provided by TCP/IP for copying a file from one host to another. FTP uses the services of TCP. FTP implemented to solve below problems.

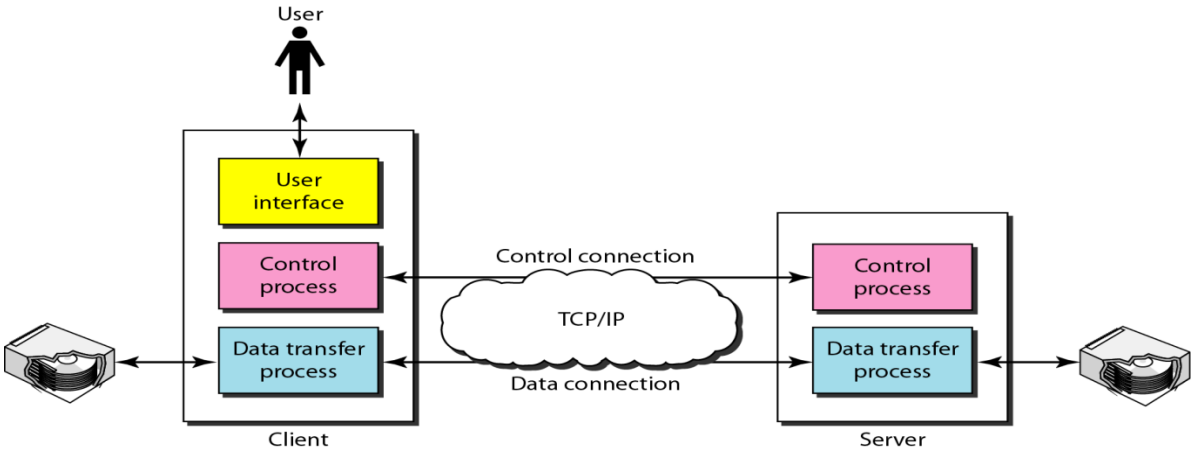
Problems with File transfer

- Two systems may use different file name conventions.
- Two systems may have different ways to represent text and data.
- Two systems may have different directory structures.

FTP differs from other client/server applications in that it establishes two connections between the hosts. FTP uses two well-known ports these connections.

1. Data transfer connection (Port 20 is used)
2. Control connection (Port 21 is used)

Basic Model of FTP

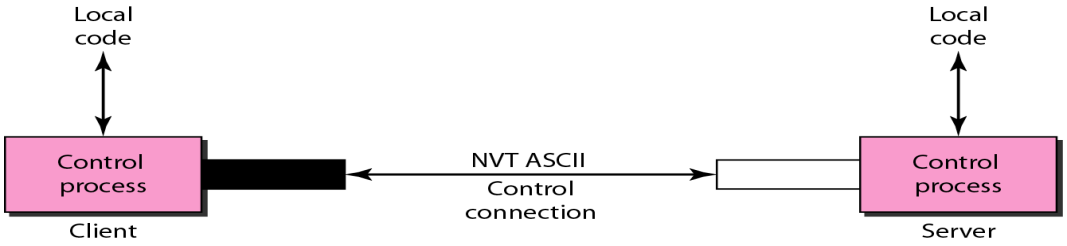


Consider the above figure that shows basic model of FTP that contains client and server components.

- Client has three components: User Interface, Client Control Process, and Client Data Transfer Process.
- Server has two components: Server Control Process and Server Data Transfer Process.
- The control connection is made between the control processes.
- The data connection is made between the data transfer processes.
- The control connection remains connected during the entire interactive FTP session.
- The data connection is opened and then closed for each file transferred.
- When a user starts an FTP session, the control connection opens. While the control connection is open, the data connection can be opened and closed multiple times if several files are transferred.

Communication over Control Connection

- FTP uses the same approach as SMTP to communicate across the control connection. FTP uses the 7-bit ASCII character set.
- Communication is achieved through commands and responses. FTP sends one command or response at a time.
- Each command or response is only one short line. Each line is terminated with a two-character end-of-line token. (One character for carriage return and other for line feed).



Communication over Data Connection

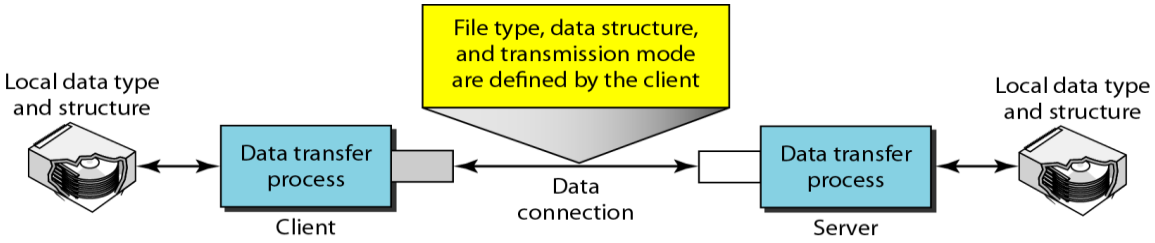
In data connection transferring of files can be done. File transfer occurs over the data connection under the control of the commands sent over the control connection.

File transfer in FTP means one of the three things: **Retrieve Store and List.**

1. A file is to be copied from the server to the client. This is called **Retrieving a file**. It is done under the supervision of the **RETR** command,
2. A file is to be copied from the client to the server. This is called **Storing a file**. It is done under the supervision of the **STOR** command.
3. A list of directory or file names is to be sent from the server to the client. This is done under the supervision of the **LIST** command.

The heterogeneity problem is resolved by defining three attributes of communication:

- File Type
- Data Structure
- Transmission Mode



File Type

FTP can transfer one of the following file types across the data connection: an ASCII file, image file or EBCDIC file (the file format used by IBM).

Data Structure

FTP can transfer a file across the data connection by using three types of data structure.

- **File Structure:** The file is a continuous stream of bytes.
- **Record Structure:** The file is divided into records. This can be used only with text files.
- **Page Structure:** The file is divided into pages and each page having a page number and a page header.

Transmission Mode

FTP uses three transmission modes: **Stream Mode, Block Mode** and **Compressed Mode.**

- **Stream mode** is the default mode. Data are delivered from FTP to TCP as a continuous stream of bytes.
- **Block Mode:** Data can be delivered from FTP to TCP in blocks. Each block is preceded by a 3-byte header. They are one byte **Block Descriptor**, **Size of the block** in 2 bytes.
- **Compressed Mode** In this mode consecutive appearances of a data unit are replaced by one occurrence and the number of repetitions.

Example of FTP

ftp> ls reports

The client sends the list command (***ls*** reports) to find the list of files on the directory named report.

Anonymous FTP

- To use FTP, a user needs an account (user name) and a password on the remote server but some sites have a set of files available for **public access**, to enable **Anonymous FTP**.
- To access these files, a user does not need to have an account or password. Instead, the user can use **anonymous** as the user name and **guest** as the password.
- User access to the system is very limited. Some sites allow anonymous users only a subset of commands.

Example: Most sites allow the user to copy some files, but do not allow navigation through the directories.

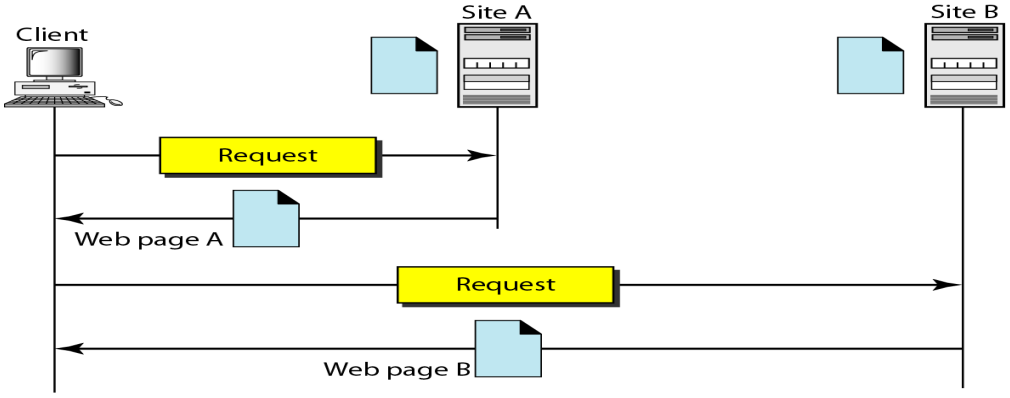
WORLD WIDE WEB (WWW)

World Wide Web (WWW) is a repository of information linked together from locations all over the world. WWW has a unique combination of flexibility, portability, and user-friendly features that distinguish it from other services provided by the Internet.

Architecture

The WWW is a distributed client/server service, in which a client using a browser can access a service using a server.

- The service provided is distributed over many locations called site.
- Each site holds one or more documents, referred to as Web pages.
- Each Web page can contain a link to other pages in the same site or at other sites is called Hyperlink.
- The pages can be retrieved and viewed by using browsers.



Architecture of WWW contains four parts: **1. Client 2. Server 3. URL 4. Cookies**

Client (Browser)

A Client is a browser that interprets and displays a Web document.

Each browser consists of three parts: **Controller**, **Client protocol**, and **Interpreters.**

- The controller receives input from the keyboard or the mouse and uses the client programs to access the document.
- After the document has been accessed, the controller uses one of the interpreters to display the document on the screen.
- The interpreter can be HTML, Java, or JavaScript depending on the type of document.
- The client protocol can be FTP or HTTP.

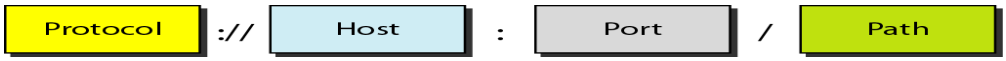
Server

- The Web page is stored at the server. Each time a client request arrives, the corresponding document is sent to the client.
- To improve efficiency, servers normally store requested files in a cache in memory.
- A server uses multi-threading or multi-processing for answering more than one request at a time to increase the efficiency.

Uniform Resource Locator (URL)

The Uniform Resource Locator (URL) is a standard for specifying any kind of information on the Internet. A client that wants to access a Web page needs the address. To facilitate the access of documents distributed throughout the world, HTTP uses locators.

URL defines four things: Protocol, Host computer, Port, and Path.



- **Protocol:** It is the client/server program used to retrieve the document. Ex:FTP or HTTP.
- **Host:** The host is the computer on which the information is located. Web pages are usually stored in computers and computers are given **alias names** that usually begin with the characters "www". This is not mandatory.
- **Port:** The URL can optionally contain the port number of the server.
- **Path:** It is the pathname of the file where the information is located.

Note: The path can itself contain slashes that separate the directories from the subdirectories and files.

Cookies

Cookies are used to devise the following functionalities:

- Some websites need to allow access to registered clients only.
- Websites are being used as electronic stores (such as Flipkart or Amazon) that allow users to browse through the store, select wanted items, put them in an electronic cart, and pay at the end with a credit card.
- Some websites are used as portals: the user selects the Web pages he wants to see.
- Some websites are just advertising.

Creation and Storage of Cookies

The creation and storage of cookies depend on the implementation:

1. When a server receives a request from a client, it stores information about the client in a file or a string.
   The information may include the domain name of the client, a timestamp, the contents of the cookie such as client name, client registration number and other information depending on the implementation.
2. The server includes the cookie in the response that it sends to the client.
3. When the client receives the response, the browser stores the cookie in the cookie directory, which is sorted by the domain server name.

Using Cookies

A cookie is used for following purposes:

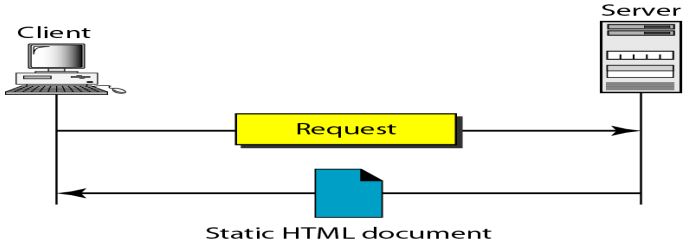
1. The site that restricts access to registered clients only sends a cookie to the client when the client registers for the first time. For any repeated access, only those clients that send the appropriate cookie are allowed.
2. An electronic store such as Flipkart or Amazon can use a cookie for its client shoppers. When a client selects an item and inserts it into a cart, a cookie that contains information about the item, such as its number and unit price, is sent to the browser.
   If the client selects a second item the cookie is updated with the new selection information.
   When the client finishes shopping and wants to check out, the last cookie is retrieved and the total charge is calculated.
3. A Web portal uses the cookie, when a user selects her favorite pages, a cookie is made and sent. If the site is accessed again, the cookie is sent to the server to show what the client is looking for.

WEB DOCUMENTS

Documents in the WWW can be grouped into three categories: **Static, Dynamic** and **Active.**

Static Documents

- Static documents are fixed-content documents that are created and stored in a server.
- The client can get only a copy of the document (i.e.) the contents of the file are determined when the file is created, not when it is used.
- The contents in the server can be changed but the user cannot change them.
- When a client accesses the document, a copy of the document is sent and the user can then use a browsing program to display the document.



Hypertext Markup Language (HTML)

- Hypertext Markup Language (HTML) is a language for creating Web pages.
- Data for a Web page are formatted for interpretation by a browser.

- HTML allows us to embed formatting instructions in the file itself. The instructions are included with the text.

A Web page is made up of two parts: **Head** and **Body.**
Head: The head is the first part of a Web page. The head contains the title of the page and other parameters that the browser will use.

Body: The actual contents of a page are in the body, which includes the text and the tags. The text is the actual information contained in a page. The tags define the appearance of the document.

HTML Tags

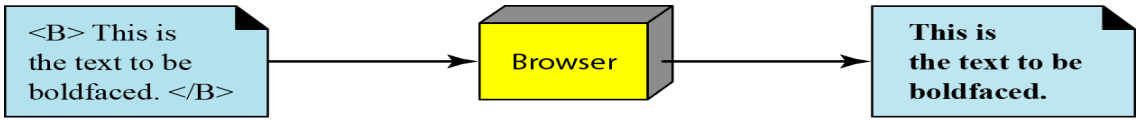
- Every HTML tag is a name followed by an optional list of attributes enclosed between less-than and greater-than symbols (< and >).
- An attribute is followed by an equals sign and the value of the attribute.
- The browser makes a decision about the structure of the text based on the tags, which are embedded into the text.

a. Beginning tag

b. Ending tag

The common tags used in HTML are : **Bold, Italic, Underline** the text.

- The two **Bold** face tags **<B>** and **</B>** are instructions for the browser.
- The two **Italic** tags **<I>** and **</I>** make the text italic
- The two **Underline** tags **<U>** and **</U>** put underline below the text.



There are two other tags used in HTML are: **Image** and **Hyperlink** tags.

Image tag

- Non-textual information such as digitized photos or graphic images is not a physical part of an HTML document.
- The image tag defines the address (URL) of the image to be retrieved. An image tag is used to point to the file of a photo or image.
- It also specifies how the image can be inserted after retrieval. Image tag contains attributes such as: SRC , ALIGN.
- SRC (source) defines the source address
- ALIGN defines the alignment of the image

<IMG SRC=”/bin/images/image1.gif” ALIGN=MIDDLE>

Hyperlink tag

- Hyperlink tag is needed to link documents together. Any item such as word, phrase, paragraph or image can refer to another document through a mechanism called an **anchor**.

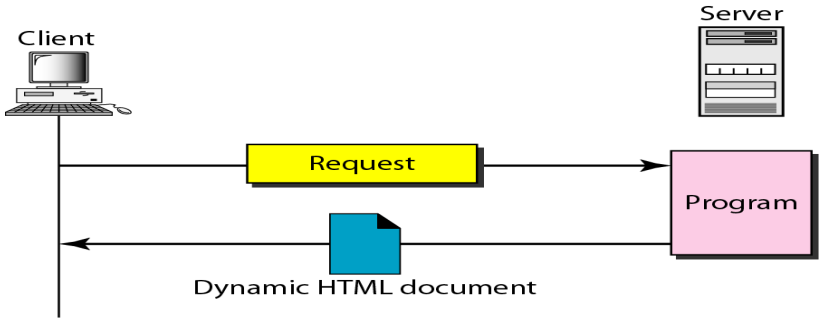
- The anchor is defined by <A ... > and </A> tags and the anchored item uses the URL to refer to another document.
- When the document is displayed, the anchored item is underlined, blinking, or boldfaced.
- The user can click on the anchored item to go to another document.
- The reference phrase is embedded between the beginning and ending tags.
- The beginning tag can have an attributes called HREF (Hyperlink Reference) defines the address (URL) of the linked document.

<A HREF= http://www/deanza.edu/forouzan> Author </A>

Dynamic Documents

- A **dynamic document** is created by a Web server whenever a browser requests the document.
- When a request arrives, the Web server runs an application program that creates the dynamic document.
- The server returns the output of the program as a response to the browser.
- Because a fresh document is created for each request, the contents of a dynamic document can vary from one request to another.

Example: the retrieval of the time and date from a server is a dynamic document.



Common Gateway Interface (CGI)

- CGI is a technology that creates and handles dynamic documents.
- CGI is a set of standards that defines how a dynamic document is written, how data are input to the program and how the output result is used
- The CGI also defines a set of rules and terms that the programmer must follow.
- CGI allows programmers to use any of several languages such as C, C++, Bourne Shell, Korn Shell, C Shell, Tcl, or Perl.
- CGI program can be used to access resources such as databases, graphical packages etc.

Input

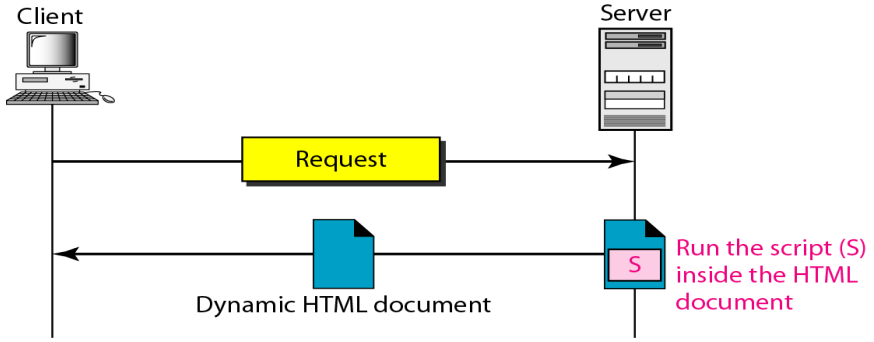
- The input from a browser to a server is sent by using a Form. If the information in a form is small (such as a word), it can be appended to the URL after a question mark.
  Example: The following URL is carrying Form information (23, a value):
  http://www.deanza/cgi-bin/prog.pl?23
- If the input from a browser is too long to fit in the query string, the browser can ask the server to send a form. The browser can then fill the form with the input data and send it to the server.
- The information in the form can be used as the input to the CGI program.

Output

- CGI executes a CGI program at the server site and send the output to the client (browser).
- The output can be a plain text, graphics or binary data, a status code, instructions to the browser to cache the result, or instructions to the server to send an existing document instead of the actual output.
- The output of the CGI program always consists of two parts: **Header** and **Body**.
- The Header is separated by a blank line from the body.

Scripting Technologies for Dynamic Documents

- The problem with CGI technology is the inefficiency that results, if part of the dynamic document that is to be created is fixed and not changing from request to request.
- The solution is to create a file containing the fixed part of the document using HTML and embed a script, a source code that can be run by the server to provide the dynamic part.
- PHP, JSP, ASP, ColdFusion are the technologies have been involved in creating dynamic documents using scripts.



Active Documents

- Applications need a program or a script to be run at the client site. These are called active documents.
- When a browser requests an active document, the server sends a copy of the document or a script. The document is then run at the client site (browser).

Example: Suppose we want to run a program that creates animated graphics on the screen. The program definitely needs to be run at the client site where the animation takes place.

Java Applets

- By using java applets we can create an active document.
- Java is a combination of a high-level programming language, a run-time environment, and a class library that allows a programmer to write an active document (an applet) and a browser to run it.
- Java can also be a stand-alone program that doesn't use a browser.
- An applet is a program written in Java on the server. It is compiled and ready to be run.
- The document is in byte-code (binary) format.
- The client process (browser) creates an instance of this applet and runs it.

JavaScript

- Java script in dynamic documents can also be used for active documents.
- If the active part of the document is small, it can be written in a scripting language; then it can be interpreted and run by the client at the same time.

- The script is in source code (text) and not in binary form.
- JavaScript is a very high level scripting language developed for this purpose.

HYPERTEXT TRANSFER PROTOCOL (HTTP)

The Hypertext Transfer Protocol (HTTP) is a protocol used mainly to access data on the World Wide Web. HTTP uses the services of TCP on well-known port 80.

HTTP functions as a combination of FTP and SMTP.

Similarity between HTTP and FTP

- HTTP is similar to FTP because it transfers files and uses the services of TCP.
- HTTP uses only TCP data connection to transfer the data between client and server and there is no control connection.

Similarity between HTTP and SMTP

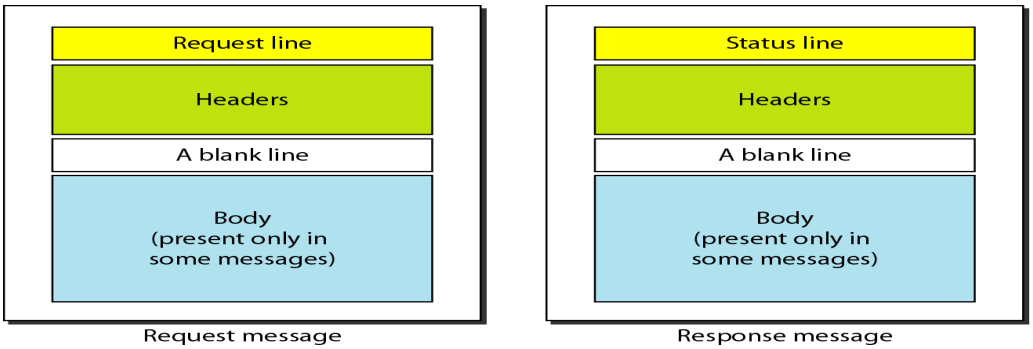
- In HTTP, the data transferred between the client and the server look like SMTP messages.
- The format of the messages is controlled by MIME-like headers.
- HTTP messages are read and interpreted by the HTTP server and HTTP client (browser).
- SMTP messages are stored & forwarded, but HTTP messages are delivered immediately.
- The commands from the client to the server are embedded in a request message.
- The contents of the requested file or information are embedded in a response message.

HTTP Transaction

HTTP is a stateless protocol even though it uses TCP services. The client initializes the transaction by sending a request message. The server replies by sending a response.

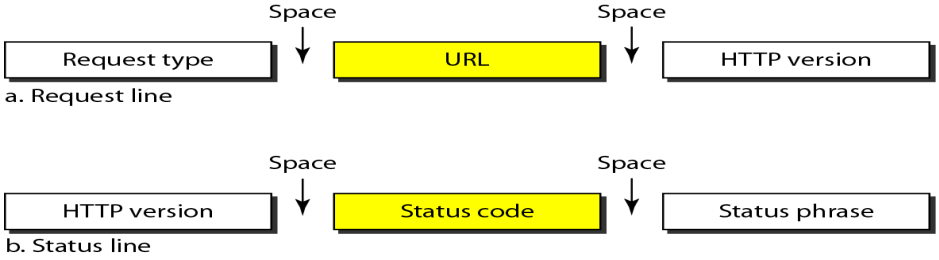
HTTP uses two types of messages: Request, Response

- A request message consists of a request line, a header, and optional body.
- A response message consists of a status line, a header, and optional body.



Request and Status Lines

- The first line in a request message is called a request line.
- The first line in the response message is called the status line.



Request type

This field is used in the request message. In version 1.1 of HTTP defines several request types. The request type is categorized into methods.

Methods	Action
GET	Requests a document from the server
POST	Sends some information from the client to the server
PUT	Sends a document from the server to the client

URL: By using URL, clients can access the webpage.

Version The most current version of HTTP is 1.1.

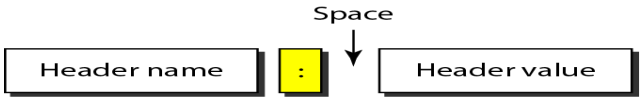
Status code is used in the response message. It consists of 3-digits. 100, 200, 300, 400, 500.

Status phrase: It explains the status code in text form, it is used in the response message.

Status Code	Status phrase	Description
100	Continue, switch	Informational
200	OK, CREATED, ACCEPTED	Successful request
300	Moved Permanently or temporarily, Not modified.	Redirect Client to another URL
400	400- Bad request, 404- Not found,	Error at client side
500	500- Internal server error 503-Service unavailable	Error at server side

Header

The header exchanges additional information between the client and the server. The header can consist of one or more header lines.



A Header line can be divided into 4 categories: 1. **General** 2.**Request** 3.**Response** 4.**Entity**.

1. A request message can contain Request header, General header, Entity header.
2. A response message can contain Response header, General header, Entity header.

General header

General header gives general information about the message such as Date, MIME version.

Request header

Request header specifies the client's configuration and the client's preferred document format.

Example: Accept: Shows the medium format the client can accept
From: Shows the E-mail address of the user
Host: Shows the host and port number of the server
Referrer: Specifies the URL of the linked document
User agent: Identifies the client program.

Response header

This header specifies the server's configuration and special information about the request.

Example: Age: shows the age of the document, public: shows the supported list of methods, server: shows the server name and address.

Entity header

The entity header gives information about the body of the document. some request messages such as POST or PUT methods may contain a body also use this type of header.

Examples: Etag- Gives an entity tag, Content-type- Specifies the medium type, Last-modified- Gives the date and time of the last change etc.

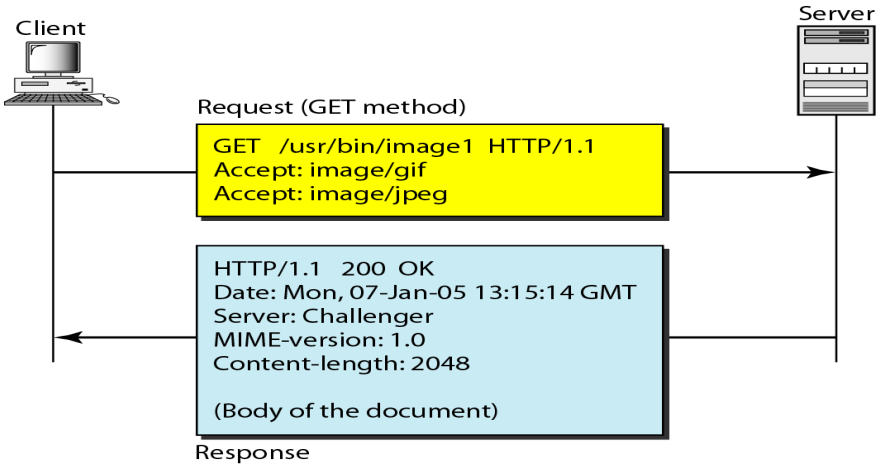
Body

The body can be present in a request or response message. Body contains the document to be sent or received.

Example of HTTP transaction

Consider the below figure that shows how to retrieve a document.

- We use the GET method to retrieve an image with the path /usr/bin/imagel.
- The request line shows the method (GET), the URL, and the HTTP version (1.1).
- Header has two lines that show the client can accept images in the GIF or JPEG format.
- The request does not have a body.
- The response message contains the status line and four lines of header.
- The header lines define the date, server, MIME version, and length of the document.
- The body of the document follows the header.



HTTP Connections

Non-persistent connection

Versions before 1.1 use Non-persistent method as the default connection. In this connection, one TCP connection is made for each request/response.

The steps involved in this strategy:

1. The client opens a TCP connection and sends a request.
2. The server sends the response and closes the connection.
3. The client reads the data until it encounters an end-of-file marker after that the client closes the connection.

In this strategy, for N different pictures in different files, the connection must be opened and closed N times.

The Non-persistent strategy imposes high overhead on the server because the server needs N different buffers and requires a slow start procedure each time a connection is opened.

Persistent Connection

- Persistent connection is the default in HTTP version 1.1.
- In this connection, the server leaves the connection open for more requests after sending a response.
- Server can close the connection at the request of a client or if a time-out has been reached.

- The sender sends the length of the data with each response. If the sender does not know the length of the data then document is created dynamically or actively.
- The server informs the client that the length is not known and closes the connection after sending the data so the client knows that the end of the data has been reached.

Proxy Server

HTTP supports Proxy Servers. A Proxy server is a computer that keeps copies of responses to recent requests.

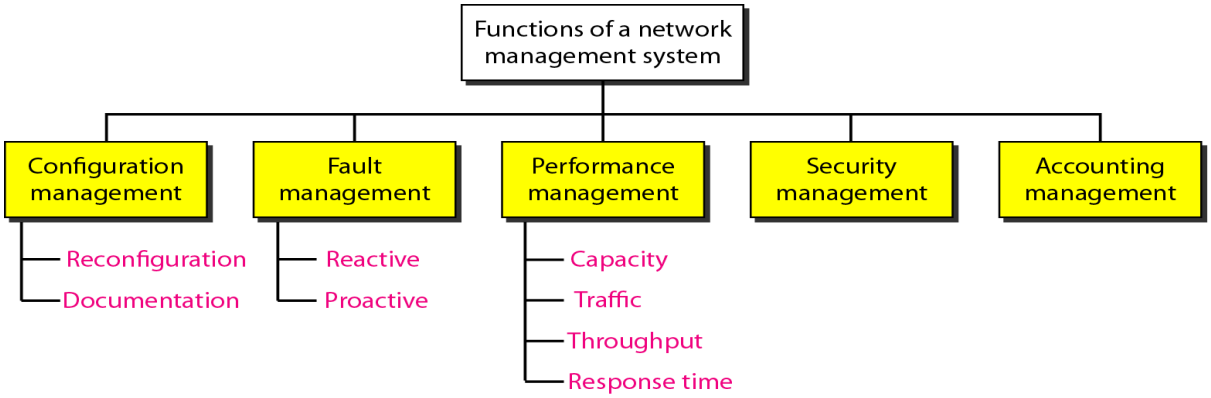
- The HTTP client sends a request to the proxy server. The proxy server checks its cache.
- If the response is not stored in the cache, the proxy server sends the request to the corresponding server.
- Incoming responses are sent to the proxy server and stored for future requests from other clients.
- Proxy server reduces the load on the original server, decreases traffic and improves latency.
- To use the proxy server, the client must be configured to access the proxy instead of the target server.

NETWORK MANAGEMENT: SNMP

Network Management can be defined as monitoring, testing, configuring and troubleshooting network components to meet a set of requirements defined by an organization.

Functions of Network Management System

Network Management System can be divided into five broad categories:



Configuration Management

- A large network is usually made up of hundreds of entities that are physically or logically connected to one another.
- These entities have an initial configuration when the network is set up, but can change with time.
- The configuration management system must know the status of each entity and its relation to other entities.

Configuration management can be divided into two subsystems: Reconfiguration and Documentation.

Reconfiguration

Reconfiguration means in a large network, the network components and features are adjusted daily. There are three types of reconfiguration:

i. **Hardware reconfiguration:** It covers all changes to the hardware. These are handled manually. Example: A subnetwork (Router) may be added or removed from the network.
ii. **Software reconfiguration:** It covers all changes to the software. Most of the software reconfiguration can be automated. Example: Updating Operating system.
iii. **User-account reconfiguration:** It covers adding and deleting the users on a system and it also considers user's individual privileges and Group privileges.
Example: A user may have read and write permission with regard to some files, but only read permission with regard to other files.

Documentation

The network configuration and each subsequent change in hardware, software and user accounts must be documented.

Hardware documentation

- It involves two sets of documents: Maps and Specifications.
- **Maps** track each piece of hardware and its connection to the network.
- General maps that shows the logical relationship as well as physical relationship between each subnetwork.
- For each sub-network, there are one or more maps that show all pieces of equipment.
- **Specification** information such as hardware type, serial number, vendor address and phone number, time of purchase and warranty information must be included for each piece of hardware connected to the network.

Software Documentation It includes information such as the software type, the version, the time installed etc.

User documentation Operating system utilities allows the documentation of user accounts and their privileges. The information in these files are updated and secured.

Fault Management

Fault management is the area of network management that handles the issues in network components. Example: A fault may be a damaged communication medium.

Fault management system has two subsystems: Reactive and Proactive.

Reactive Fault Management

The responsibilities of reactive fault management can be divided into 4 steps:

i. **Detect the fault:** Fault management system must have to detect the exact location of the fault.
ii. **Isolate the fault:** If a fault is isolated that affects only a few users. After isolation, the affected users are immediately notified and given an estimated time of correction.
iii. **Correct the fault:** This may involve replacing or repairing the faulty components.
iv. **Record the fault:** After the fault is corrected, it must be recorded (i.e. documented). The record should show the exact location of the fault, the possible cause, the action or actions taken to correct the fault, the cost and time it took for each step.

Proactive Fault Management

Proactive fault management tries to prevent faults from occurring. Some failures can be predicted and prevented.

Performance Management

Performance management tries to monitor and control the network to ensure that it is running as efficiently. It can be measured by the following concepts: Capacity, Traffic, Throughput, Response time.

- **Capacity of the Network** Every network has a limited capacity, and the performance management system must ensure that it is not used above this capacity.
  Example: If a LAN is designed for 100 stations at an average data rate of2 Mbps, it will not operate properly if 200 stations are connected to the network.
- **Traffic** Traffic can be measured in two ways: Internally and Externally.
  Internal traffic is measured by the number of packets (or bytes) traveling inside the network.
  External traffic is measured by the exchange of packets (or bytes) outside the network.
- **Throughput** It can be measured by an individual device such as a router or a part of the network. Throughput makes sure that, the device is not reduced to unacceptable levels.
- **Response Time** It is measured from the time a user requests a service to the time the service is granted.

Security Management

Security management is responsible for controlling access to the network based on the predefined policy.

Accounting Management

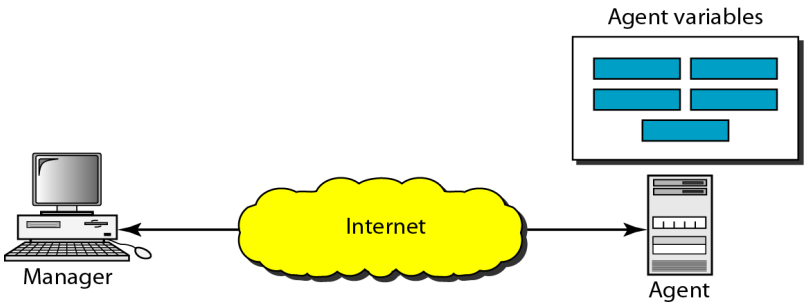
- Accounting management is the control of users' access to network resources through charges.
- Under accounting management, individual users, departments, divisions, or even projects are charged for the services they receive from the network.

SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP)

The Simple Network Management Protocol (SNMP) is a framework for managing devices in an internet using the TCP/IP protocol suite. It provides a set of fundamental operations for monitoring and maintaining an internet.

Concept of SNMP

- SNMP is an application level protocol that uses the concept of Manager and Agent.
- A manager controls and monitors a set of agents.
- A manager may be a host and an Agent may be a router.
- SNMP can monitor devices made by different manufacturers and installed on different physical networks.
- SNMP can be used in LANs and WANs connected by routers.



Managers and Agents

- A Manager or a management station is a host that runs the SNMP client program.
- An Agent or a Managed station is a router or a host that runs the SNMP server program.

Management is achieved through simple interaction between a manager and an agent.

- Agent keeps performance information in a database.
- Manager has access to the values in the database.

Example: A router can store in variables such as the number of packets received and forwarded. The manager can fetch and compare the values of these two variables to see if the router is congested or not.

Management with SNMP is based on three basic ideas:

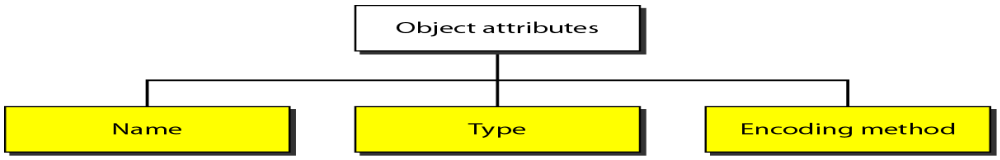
1. A manager checks an agent by requesting information that reflects the behavior of the agent.
2. A manager forces an agent to perform a task by resetting values in the agent database.
3. An agent contributes to the management process by sending warning message to the manager of an unusual situation. The warning message is called the trap.

Management Components

- SNMP uses two other protocols to do management tasks: Structure of Management Information (SMI) and Management Information Base (MIB).

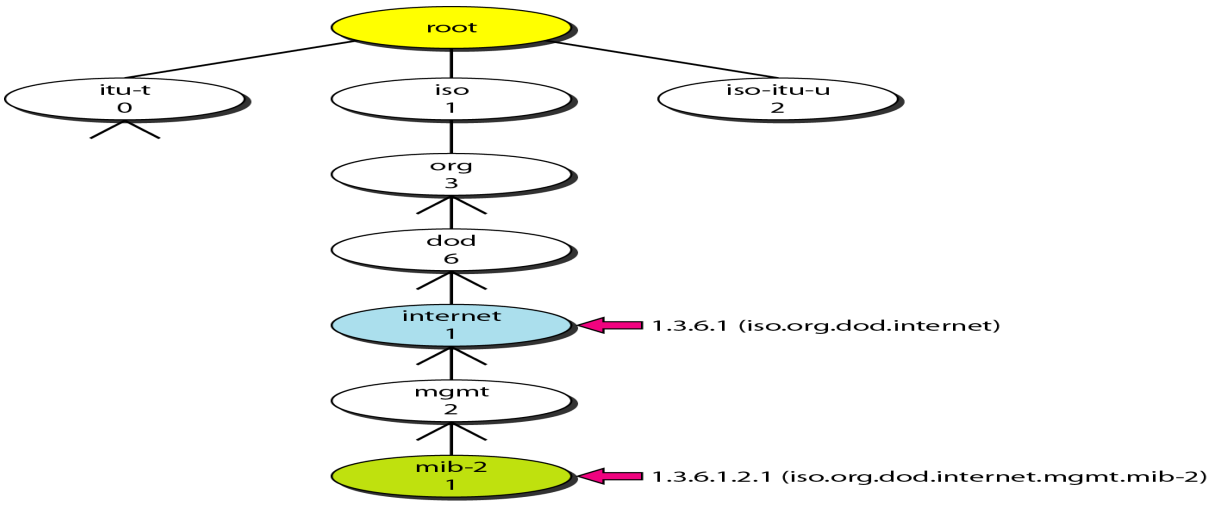
Structure of Management Information (SMI)

The Structure of Management Information version 2 (SMIv2) is a component for network management. Functions of SMI are:



Name

- SMI requires that each managed object (such as a router, a variable in a router, a value) have a unique name.
- To name objects globally SMI uses an object identifier, which is a hierarchical identifier based on a tree structure. The tree structure starts with an unnamed root.



- Each object can be defined by using a sequence of integers separated by dots.
- Tree structure can also define an object by using a sequence of textual names separated by dots.
- The integer-dot representation is used in SNMP. The name-dot notation is used by people.

Example: The following shows the same object in two different notations:

iso.org.dod.internet.mgmt.mib-2 → 1.3.6.1.2.1

Type

- To define the data type, SMI uses fundamental Abstract Syntax Notation 1 (ASN.l).
- SMI has two broad categories of data type: Simple and Structured.

Simple Type

The simple data types are atomic data types sucha as Integer32, Octet String, IP address etc.

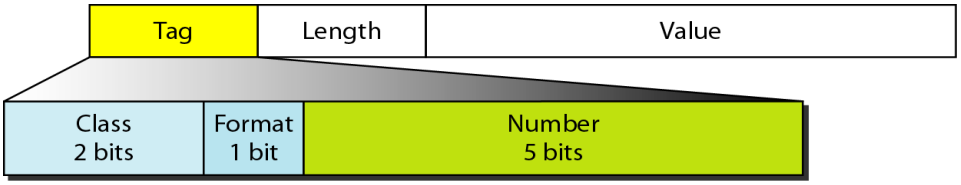
Structured Type

By combining simple and structured data types, we can make new structured data types.
SMI defines two structured data types: Sequence and Sequence of.

- A **Sequence** data type is a combination of simple data types, not necessarily of the same type. It is similar to the concept of a struct used in C programming.
- A **Sequence of** data type is a combination of simple data types all of the same type. It is similar to the concept of an array used in C-programming.

Encoding Method

SMI uses another standard Basic Encoding Rules (BER) to encode data to be transmitted over the network. BER specifies data to be encoded in following format: Tag, Length and Value.



Tag

The tag is a 1-byte field that defines the type of data. It is composed of three subfields:

- **Class (2 bits):** It defines the scope of the data.
  Four classes are defined: 00- universal, 01-Application wide, 10- context specific, 11-private.
- **Format** subfield indicates whether the data are simple (0) or structured (1).
- **Number** subfield further divides simple or structured data into subgroups.
  Example: In the universal class with simple format, INTEGER has a value of 2, OCTET STRING has a value of 4.

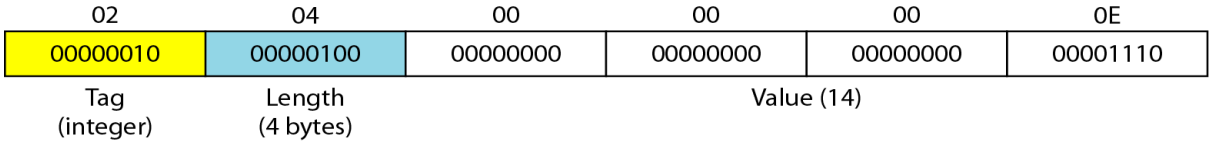
Length

- The length field is 1 or more bytes.
- If it is 1 byte, the most significant bit must be 0. Other 7 bits define the length of the data.
- If it is more than 1 byte, the most significant bit of the first byte must be 1. The other 7 bits of the first byte define the number of bytes needed to define the length.

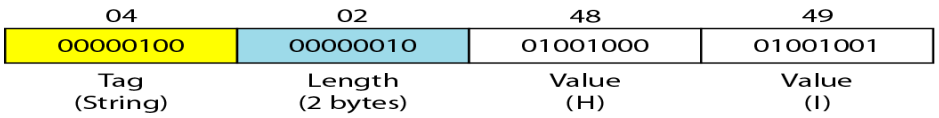
Value field codes the value of the data according to the rules defined in BER.

Example: Show the following in encoding representation:

1. Define INTEGER 14.

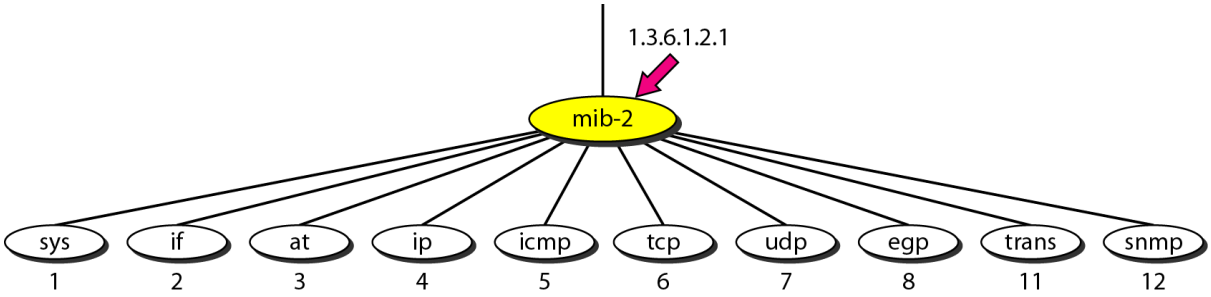


2. Define OCTET STRING "HI"



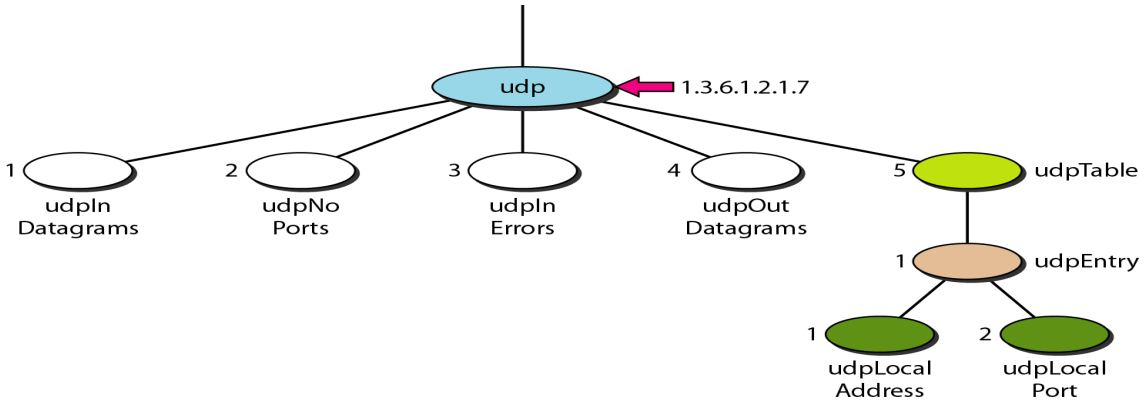
Management Information Base (MIB2)

- MIB version 2 is the second component used in network management.
- MIB creates a collection of named objects, their types and their relationships to each other in an entity to be managed.
- Each agent has its own MIB2, which is a collection of all the objects that the manager can manage.
- The objects in MIB2 are categorized under 10 different groups: system, interface, address translation, ip, icmp, tcp, udp, egp, transmission, and snmp.
- These groups are under the MIB-2 object in the object identifier tree.
- Each group has defined variables and tables.



Accessing MIB Variables

Let us take UDP group to show how to access different variables. To access any of the simple variables, we use the id of the group (1.3.6.1.2.1.7) followed by the id of the variable.



The following shows how to access each variable:

udpInDatagrams	→	1.3.6.1.2.1.7.1
udpNoPorts	→	1.3.6.1.2.1.7.2
udpInErrors	→	1.3.6.1.2.1.7.3
udpOutDatagrams	→	1.3.6.1.2.1.7.4

The object identifiers define the variable not instances (contents). An instance suffix "**0**" should be added to show the instance of each variable.

udpInDatagrams.0 → 1.3.6.1.2.1.7.1.0
udpNoPorts.0 → 1.3.6.1.2.1.7.2.0
udpInErrors.0 → 1.3.6.1.2.1.7.3.0
udpOutDatagrams.0 → 1.3.6.1.2.1.7.4.0

Tables

To identify a table, we first use the table id. To access the table, we have to define the table entries.

udpTable → 1.3.6.1.2.1.7.5
udpEntry → 1.3.6.1.2.1.7.5.1

To access the entry we need to define each entity (field) in the entry.

udpLocalAddress → 1.3.6.1.2.1.7.5.1.1
udpLocalPort → 1.3.6.1.2.1.7.5.1.2

- To access a specific instance (row) of the table, we add the index to the above ids. To access the instance of the local address for the first row, we use the identifier augmented with the instance index:

  udpLocalAddress.181.23.45.14.23 → 1.3.6.1.2.7.5.1.1.181.23.45.14.23

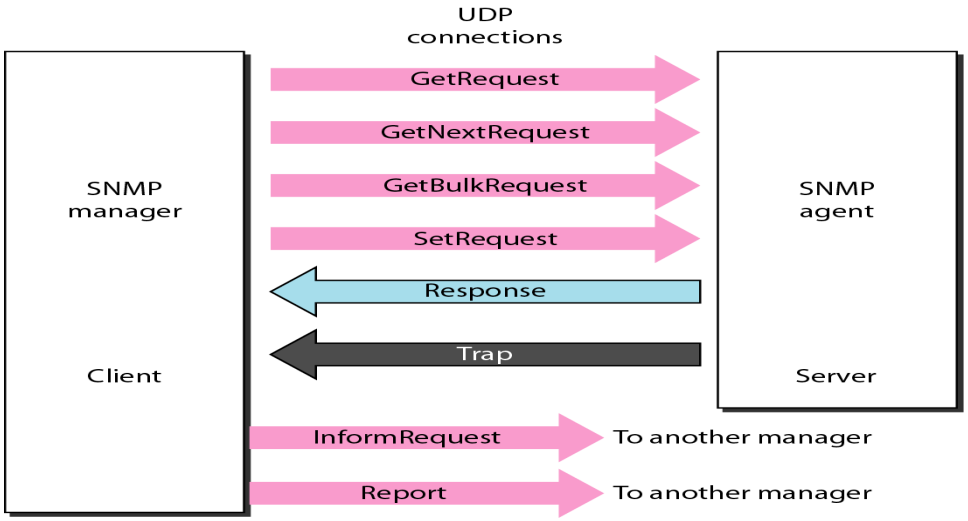
Lexicographic Ordering

- The object identifiers follow in lexicographic order.
- Tables are ordered column by column from the top to the bottom.
- The lexicographic ordering enables a manager to access a set of variables one after another by defining the first variable.

SNMP version3 (SNMPv3)

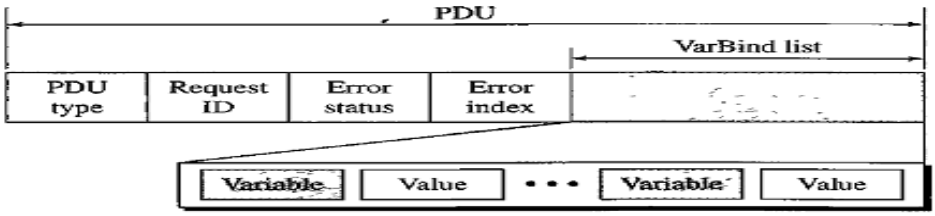
- SNMP defines the format of packets exchanged between a manager and an agent.
- SNMP interprets the result and creates statistics.
- The packets exchanged contain the object (variable) names and their status (values).
- SNMP is responsible for reading and changing these values.
- SNMP uses both SMI and MIB in Internet network management.

SNMPv3 defines eight types of packets or PDUs.



1. **GetRequest** PDU is sent from the manager (client) to the agent (server) to retrieve the value of a variable or a set of variables.
2. **GetNextRequest** PDU is sent from the manager to the agent to retrieve the value of a variable. It is mostly used to retrieve the values of the entries in a table.
3. **GetBulkRequest** PDU is sent from the manager to the agent to retrieve a large amount of data.
4. **SetRequest** PDU is sent from the manager to the agent to set (store) a value in a variable.
5. **Response** PDU is sent from an agent to a manager in response to GetRequest or GetNextRequest.
6. **Trap** PDU is sent from the agent to the manager to report an event. For example, if the agent is rebooted, it informs the manager and reports the time of rebooting.
7. **InformRequest** PDU is sent from one manager to another remote manager to get the value of some variables from agents under the control of the remote manager. The remote manager responds with a Response PDU.
8. **Report** PDU is designed to report some types of errors between managers. It is not yet in use.

Format of PDU



- **PDU type.** This field defines the type of the PDU.
- **Request ID** This field is a sequence number used by the manager in a Request PDU and repeated by the agent in a response. It is used to match a request to a response.
- **Error status**. This is an integer that is used only in Response PDUs to show the types of errors reported by the agent. Its value is 0 in Request PDUs.
- **Error index**. It is an offset that tells the manager which variable caused the error.
- **Non-repeaters**. This field is used only in GetBulkRequest and replaces the error status field, which is empty in Request PDUs.
- **Max-repetition**. This field is also used only in GetBulkRequest and replaces the error index field, which is empty in Request PDUs.
- **VarBind list**. This is a set of variables with the corresponding values the manager wants to retrieve or set. The values are null in GetRequest and GetNextRequest.

Messages

SNMP embeds the PDU in a message. A message in SNMPv3 is made of four elements: Version, Header, Security parameters and Data.

- **Version** defines the current version (3).
- **Header** contains values for message identification, maximum message size, message flag, and a message security model.
- **Security parameter** is used to create a message digest.
- **Data** contain the encoded PDU. The data may or may not be encrypted.

SNMP UDP Ports

- SNMP uses the services of UDP on two well-known ports, 161 and 162.
- The well-known port 161 is used by the server (agent).
- The well-known port 162 is used by the client (manager).

Security

- SNMPv3 provides two types of security: General and Specific.
- SNMPv3 provides message authentication, privacy, and manager authorization.
- SNMPv3 allows a manager to remotely change the security configuration, which means that the manager does not have to be physically present at the manager station.