DOMAIN NAME SYSTEM (DNS)

The DNS is a distributed database that resides on multiple machines on the internet. It provide e-mail routing information.The DNS protocol runs over UDP and uses port 53.

Figure 5.4.1 shows how TCP/IP uses a DNS client and a DNS server to map a name to an address. A user wants to use a file transfer client to access the corresponding file transfer server running on a remote host. The user knows only the file transfer server name, such as a filesource.com. The TCP/IP suite needs the IP address of the file transfer server to make the connection.

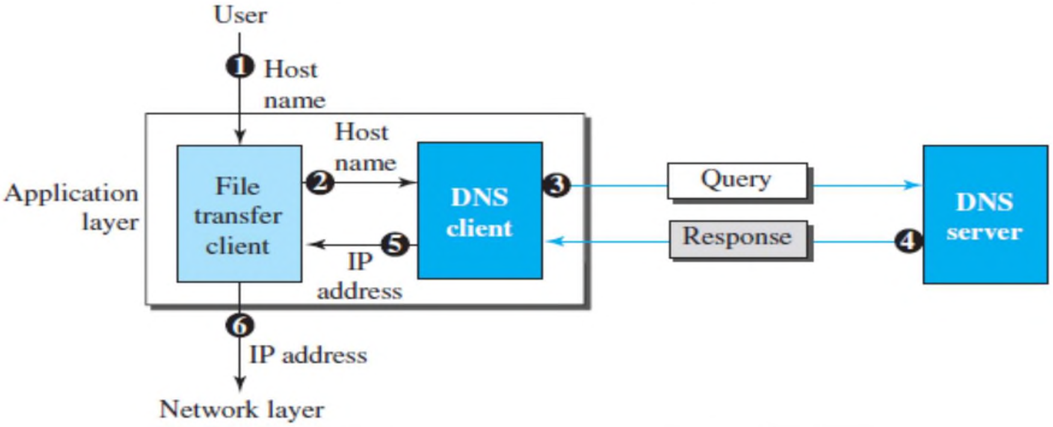


Fig5.4.1: Purpose of DNS

[Source :"Data Communications and Networking" by Behrouz A. Forouzan,Page-910]

The following six steps map the host name to an IP address:

The user passes the host name to the file transfer client.

The file transfer client passes the host name to the DNS client.

Each computer, after being booted, knows the address of one DNS server. The DNS client sends a message to a DNS server with a query that gives the file transfer server name using the known IP address of the DNS server.The DNS server responds with the IP address of the desired file transfer server. The DNS server passes the IP address to the file transfer client. The file transfer client now uses the received IP address to access the file transfer server.

Domain Name Space

To have a hierarchical name space, a domain name space was designed. In this design the names are defined in an inverted-tree structure with the root at the top. The tree has 128 levels: level 0 (root) to level 127 (see Figure 5.4.2).

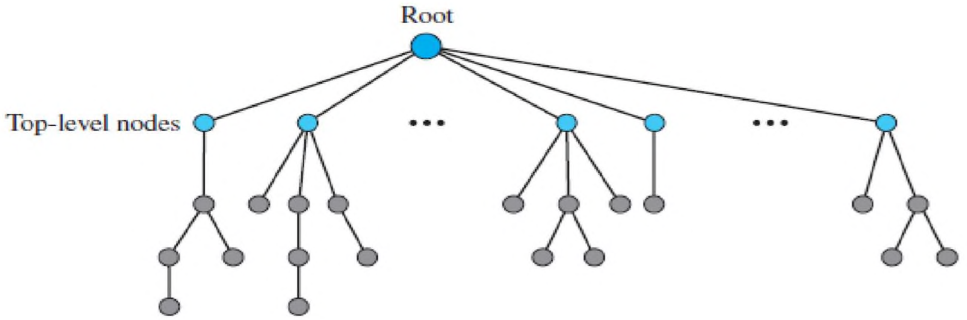


Fig5.4.2: DNS tree.

[Source :"Data Communications and Networking" by Behrouz A. Forouzan,Page-912]

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).

Domain Name

Each node in the tree has a domain name. A full domain name is a sequence of labels separated by dots (.). The domain names are always read from the node up to the root as in figure 5.4.3. The last label is the label of the root (null). This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing.

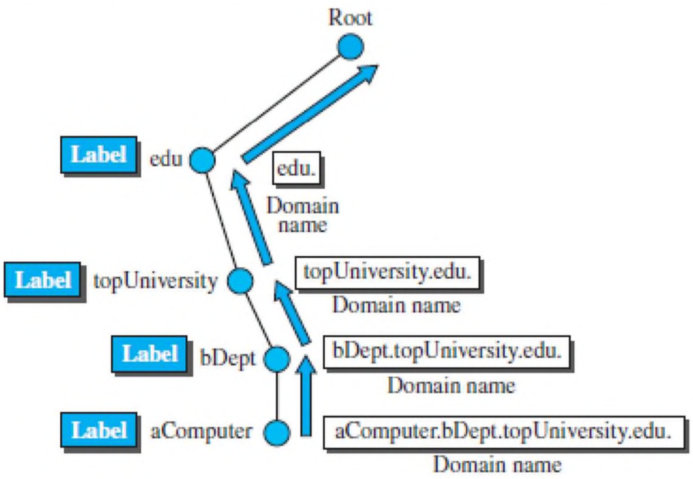


Fig5.4.3: Domain names.

[Source :"Data Communications and Networking" by Behrouz A. Forouzan,Page-913]

If a label is terminated by a null string, it is called a fully qualified domain name (FQDN). The name must end with a null label, but because null means nothing, the label ends with a dot.

If a label is not terminated by a null string, it is called a partially qualified domain name (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.

A domain is a sub tree of the domain name space. The name of the domain is the name of the node at the top of the sub tree. Figure (below) shows some domains. A domain is divided into domains.

Distribution of Name Space

The information contained in the domain name space must be stored. It is inefficient and not reliable to have just one computer store such a huge amount of information. It is inefficient because responding to requests from all over the world places a heavy load on the system.

Hierarchy of Name Servers

To distribute the information among many computers called DNS servers as shown in figure 5.4.4. One way to do this is to divide the whole space into many domains based on the first level. DNS allows domains to be divided further into smaller domains (subdomains). Each server can be responsible (authoritative) for either a large or small domain.

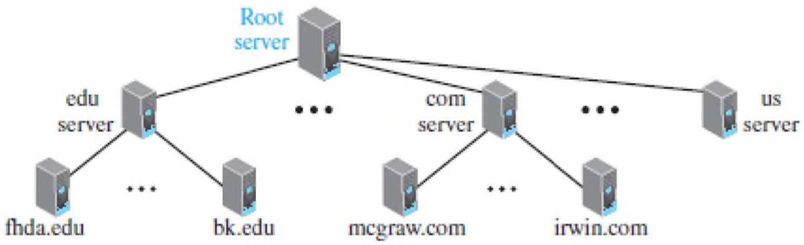


Fig5.4.4: Domain name hierarchy.

[Source : "Data Communications and Networking" by Behrouz A. Forouzan,Page-914]

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 data base called a zone file and keeps all the information for every node under that domain as shown in figure 5.4.5.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.

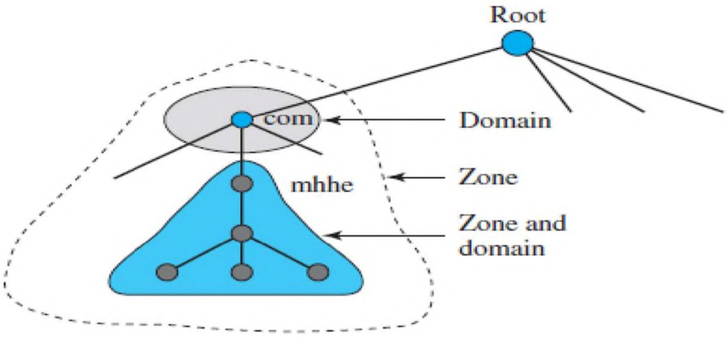


Fig5.4.5: Zone

[Source :"Data Communications and Networking" by Behrouz A. Forouzan,Page-914]

Root Server

If zone consists of the full tree then that zone server is called root server. A root server does not store any information about domains. 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.

A secondary server is a server that loads all information from the primary server. Secondary server cannot perform any operation on zone file.

DNS in the Internet

In the Internet, the domain name space (tree) was divided into three sections: generic domains, country domains, and the inverse domains.

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 as shown in figure 5.4.6. Looking at the tree, the first level in the generic domains section allows 14 possible labels.

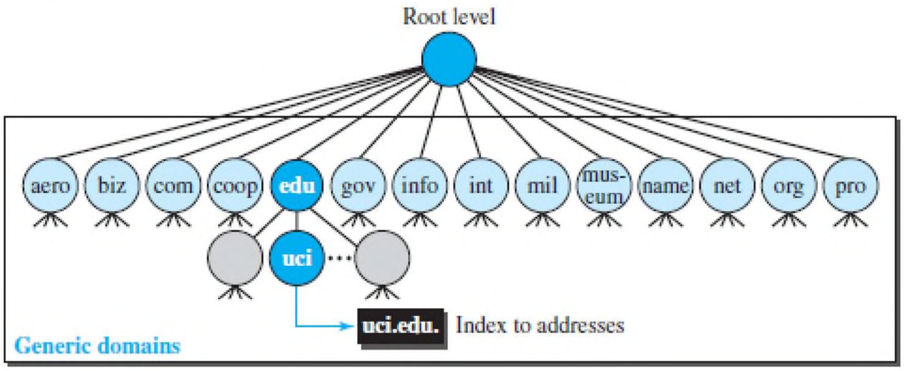


Fig5.4.6: Generic domain.

[Source :"Data Communications and Networking" by Behrouz A. Forouzan,Page-915]

Country Domains

The country domains section uses two-character country abbreviations (e.g., us for United States) as shown in figure 5.4.7. Second labels can be organizational, or they can be more specific national designations.

Example for the country domains section. The address uci.ca.us. can be translated to University of California, Irvine, in the state of California in the United States.

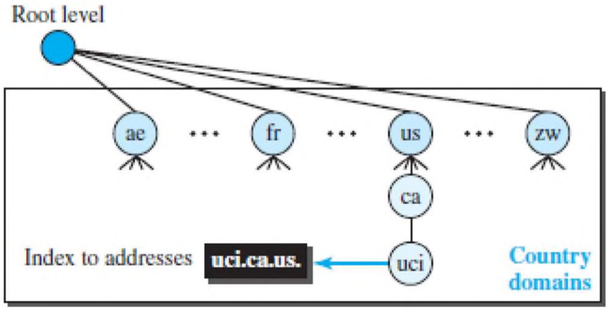


Fig5.4.7: Country domain

[Source : "Data Communications and Networking" by Behrouz A. Forouzan,Page-916]

Inverse domain

Inverse domain is used to find the name of a host when given the IP address.

Resolution: Mapping a name to an address is called name-address resolution.

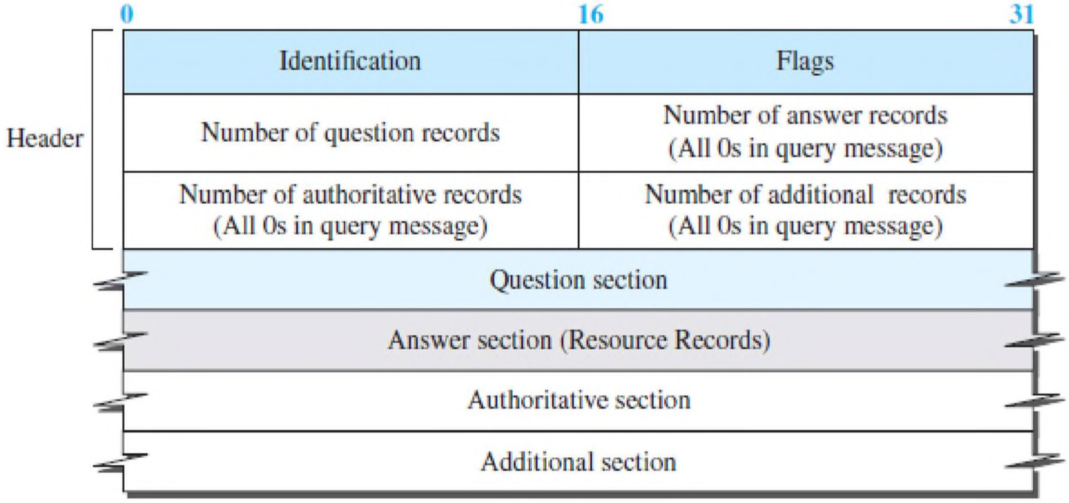
DNS is designed as a client-server application. A host that needs to map an address to a name or a name to an address calls a DNS client called a resolver.The resolver accesses the closest DNS server with a mapping request. If the server has the information, it satisfies the resolver; otherwise, it either refers the resolver to other servers or asks other servers to provide the information.

Recursive Resolution

A client request complete translation. If the server is authority for the domain name, it checks its database and responds. If the server is not authority, it sends the request to another server and waits for the response.When the query is resolved ,the response travel back until it finally reaches the requesting client. This is called recursive resolution.

DNS Messages

To get information about hosts, DNS uses two types of messages: query and response. Both messages have the same format as shown in Figure 5.4.8.


0
16
31
Header
Identification
Flags
Number of question records
Number of answer records (All 0s in query message)
Number of authoritative records (All 0s in query message)
Number of additional records (All 0s in query message)
Question section
Answer section (Resource Records)
Authoritative section
Additional section

Note:
The query message contains only the question section.
The response message includes the question section,
the answer section, and possibly two other sections.

Fig5.4.8: DNS messages.

[Source :"Data Communications and Networking" by Behrouz A. Forouzan,Page-919]

The identification field is used by the client to match the response with the query. The flag field defines whether the message is a query or response. It also includes status of error. The next four fields in the header define the number of each record type in the message. The question section consists of one or more question records. It is present in both query and response messages. The answer section consists of one or more resource records. It is present only in response messages. The authoritative section gives information (domain name) about one or more authoritative servers for the query. The additional information section provides additional information that may help the resolver.

Encapsulation

DNS can use either UDP or TCP. The port used by the server is port 53. UDP is used when the size of the response message is less than 512 bytes because most UDP packages have a 512-byte packet size limit. If the size of the response message is more than 512 bytes, a TCP connection is used.

Security of DNS

Applications such as Web access or e-mail are dependent on the proper operation of DNS. DNS can be attacked in several ways .

1.The attacker may read the response of a DNS server to find the nature or names of sites the user mostly accesses. This type of information can be used to find the user's profile. To prevent this attack, DNS messages need to be confidential .

2. The attacker may intercept the response of a DNS server and change it or create a totally new bogus response to direct the user to the site or domain the attacker wishes the user to access. This type of attack can be prevented using message origin authentication and message integrity.

3**.** The attacker may flood the DNS server to overwhelm it or eventually crash it. This type of attack can be prevented using the provision against denial-of-service attack.

To protect DNS, IETF has devised a technology named DNS Security (DNSSEC) that provides message origin authentication and message integrity using a security service called digital signature.

5.5 ELECTRONIC MAIL

Electronic mail (or e-mail) allows users to exchange messages. Electronic mail is used for sending a single message like text, voice,video or grapics to one or more recipients. Email is fast,easy to distribute and inexpensive. The sender and receiver use three different agents: a user agent (UA), a message transfer agent(MTA), and a message access agent (MAA).

The mail server uses a queue (spool) to store messages waiting to be sent. The message, needs to be sent through the Internet from client to server site using an MTA. Here two message transfer agents are needed: one client and one server.

User Agent

The first component of an electronic mail system is the user agent (UA).It provides service to the user to make the process of sending and receiving a message easier. A user agent is a software package (program) that composes, reads, replies to, and forwards messages. It also handles local mailboxes on the user computers.There are two types of user agents: command-driven and GUI-based. A command-driven user agent normally accepts a one character command from the keyboard to perform its task.

Sending Mail

To send mail, the user, through the UA, creates mail that looks very similar to postal mail. It has an envelope and a message.The message contains the header and the body. The header of the message defines the sender, the receiver, the subject of the message, and some other information. The body of the message contains the actual information to be read by the recipient.

Receiving Mail

The user agent is triggered by the user (or a timer). If a user has mail, the UA 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

To deliver mail, a mail handling system must use an addressing system with unique addresses.

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

Simple Mail Transfer Protocol (SMTP)

Simple Mail Transfer Protocol (SMTP) act as message transfer agent(MTA).

Commands and Responses

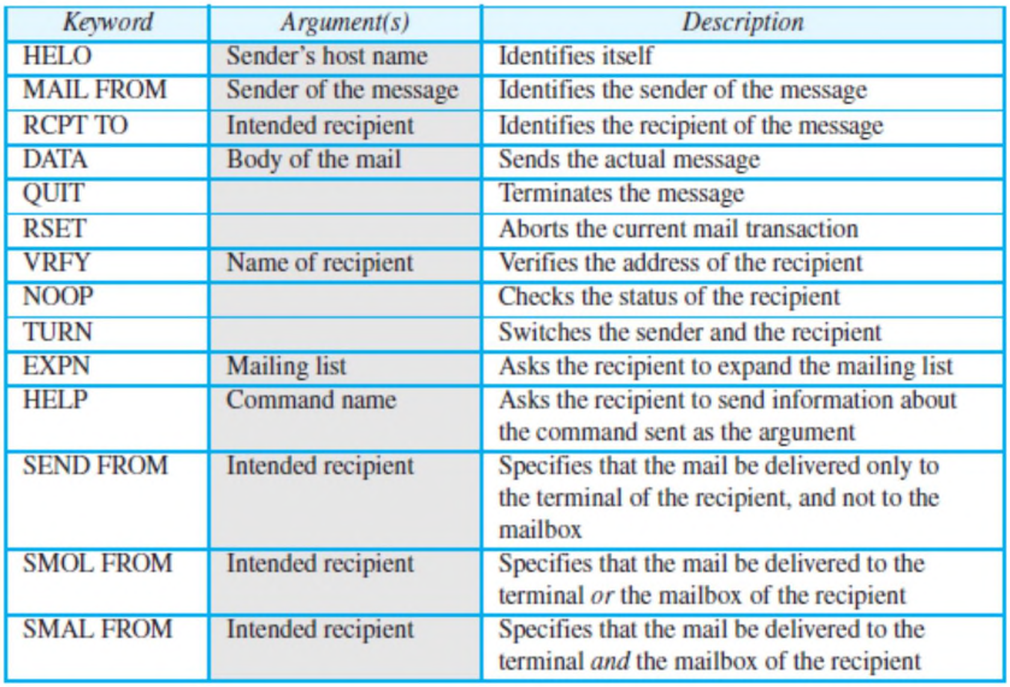
SMTP uses commands and responses to transfer messages between an MTA client and an MTA server. The command is from an MTA client to an MTA server; the response is from an MTA server to the MTA client.

Commands are sent from the client to the server. Keyword: argument(s)

SMTP defines 14 commands as shown in table 5.5.1.

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

Table5.5.1:SMTP Commands



Keyword	*Argument(s)*	*Description*
HELO	Sender's host name	Identifies itself
MAIL FROM	Sender of the message	Identifies the sender of the message
RCPT TO	Intended recipient	Identifies the recipient of the message
DATA	Body of the mail	Sends the actual message
QUIT		Terminates the message
RSET		Aborts the current mail transaction
VRFY	Name of recipient	Verifies the address of the recipient
NOOP		Checks the status of the recipient
TURN		Switches the sender and the recipient
EXPN	Mailing list	Asks the recipient to expand the mailing list
HELP	Command name	Asks the recipient to send information about the command sent as the argument
SEND FROM	Intended recipient	Specifies that the mail be delivered only to the terminal of the recipient, and not to the mailbox
SMOL FROM	Intended recipient	Specifies that the mail be delivered to the terminal *or* the mailbox of the recipient
SMAL FROM	Intended recipient	Specifies that the mail be delivered to the terminal *and* the mailbox of the recipient

Mail Transfer Phases

The process of transferring a mail message occurs in three phases: connection establishment, mail transfer, and connection termination.

Connection Establishment: After a client has made a TCP connection to the well known port 25, the SMTP server starts the connection phase. This phase involves the following three steps:

1. The server sends code 220 (service ready) to tell the client that it is ready to receive mail. If the server is not ready, it sends code 421 (service not available).

2. The client sends the HELO message to identify itself, using its domain name address. This step is necessary to inform the server of the domain name of the client.

3. The server responds with code 250 (request command completed) or some other code depending on the situation.

Message Transfer: After connection has been established between the SMTP client and server, a single message between a sender and one or more recipients can be exchanged. This phase involves eight steps. Steps 3 and 4 are repeated if there is more than one recipient.

1. The client sends the MAIL FROM message to introduce the sender of the message.It includes the mail address of the sender (mailbox and the domain name). This step is needed to give the server the return mail address for returning errors and reporting messages.
2. The server responds with code 250 or some other appropriate code.
3. The client sends the RCPT TO (recipient) message, which includes the mail address of the recipient.
4. The server responds with code 250 or some other appropriate code.
5. The client sends the DATA message to initialize the message transfer.
6. The server responds with code 354 (start mail input) or some other appropriate message.
7. The client sends the contents of the message in consecutive lines. Each line is terminated by a two-character end-of-line token (carriage return and line feed). The message is terminated by a line containing just one period.
8. The server responds with code 250 (OK) or some other appropriate code.

Connection Termination: After the message is transferred successfully, the client terminates the connection.

Two steps.

. The client sends the QUIT command.

The server responds with code 221 or some other appropriate code.

POP3 (Post Office Protocol, version 3)

Post Office Protocol, version 3 (POP3) is simple in its function.

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 its 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 then list and retrieve the mail messages, one by one. Figure 5.5.1 shows an example of downloading using POP3.

POP3 has two modes: the delete mode and the keep mode.

In the delete mode, the mail is deleted from the mailbox after each retrieval. In the keep mode, the mail remains in the mailbox after retrieval.The delete mode is normally used when the user is working at her permanent computer and can save and organize the received mail after reading or replying.The keep mode is normally used when the user accesses her mail away from her primary computer (for example, from a laptop).

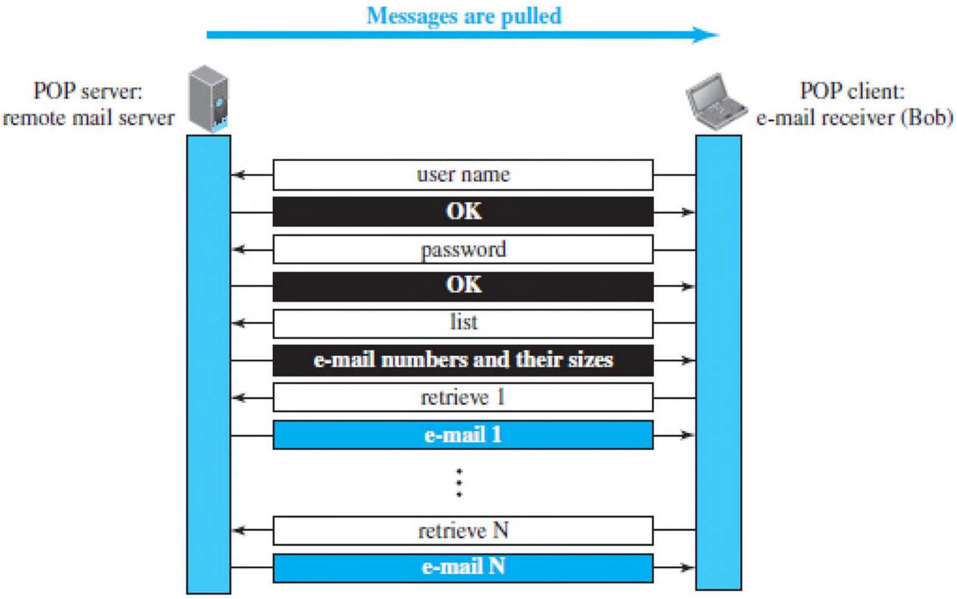


Fig5.5.1: POP3 format.
[Source : "Data Communications and Networking" by Behrouz A. Forouzan,Page-899]

POP3 drawbacks.

It does not allow the user to organize her mail on the server; the user cannot have different folders on the server. In addition, POP3 does not allow the user to partially check the contents of the mail before downloading.

IMAP4

IMAP4 is Internet Mail Access Protocol, version 4 (IMAP4). IMAP4 is similar to POP3, but it has more features; IMAP4 is more powerful and more complex.

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 very 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.

MIME

Multipurpose Internet Mail Extensions (MIME) is a supplementary protocol that allows non-ASCII data to be sent through e-mail. MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers it to the client MTA to be sent through the Internet.The message at the receiving site is transformed back to the original data. MIME means a set of software functions that transforms non-ASCII data to ASCII data as shown in figure 5.5.2.

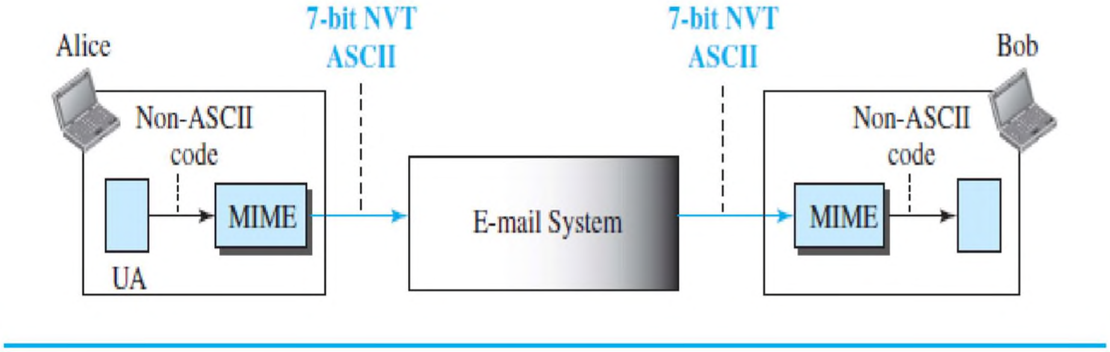


Fig5.5.2: MIME.

[Source : "Data Communications and Networking" by Behrouz A. Forouzan,Page-900]

MIME Headers

MIME defines five headers, as shown in Figure5.5.3 , which can be added to the original e-mail header section to define the transformation parameters.

MIME-Version The current version is 1.1.

Content-Type This header defines the type of data used in the body of the message. The content type and the content subtype are separated by a slash. Depending on the subtype, the header may contain other parameters.

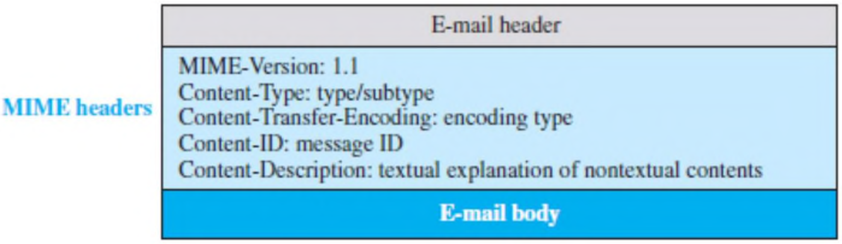
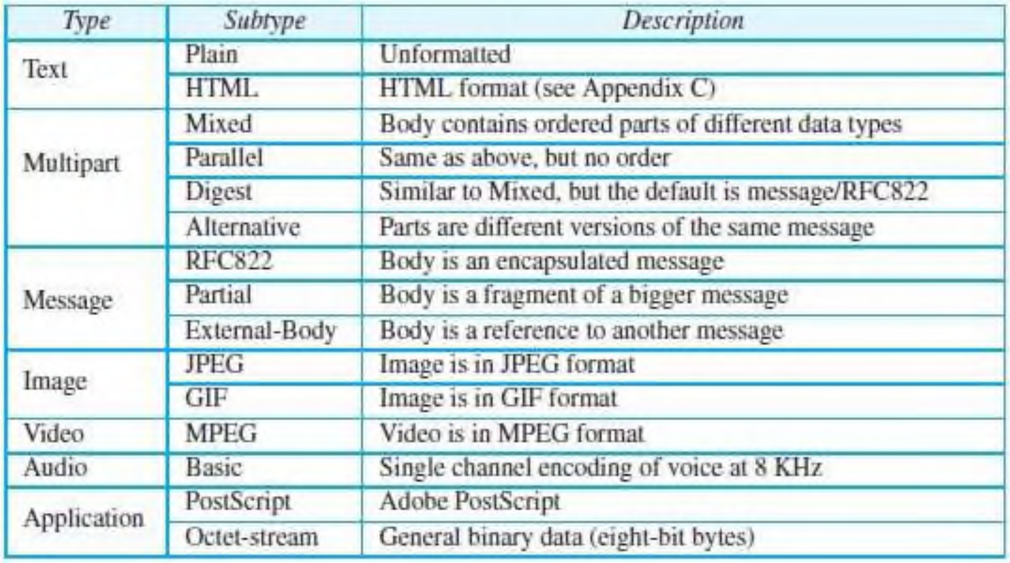

E-mail header
MIME headers
MIME-Version: 1.1
Content-Type: type/subtype
Content-Transfer-Encoding: encoding type
Content-ID: message ID
Content-Description: textual explanation of nontextual contents
E-mail body

Fig5.5.3: MIME header.

[Source :"Data Communications and Networking" by Behrouz A. Forouzan,Page-901]

MIME allows seven different types of data as listed in table 5.5.2.

Table5.5.2: Data types and subtypes in MIME

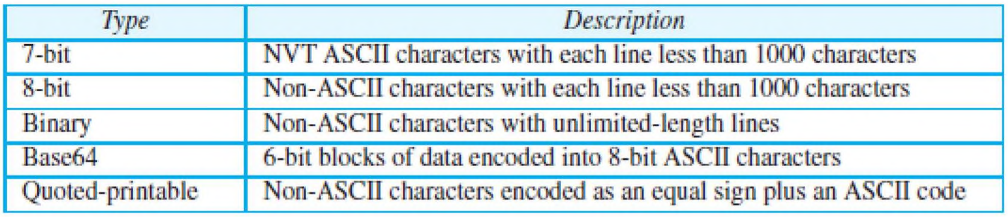


Type	*Subtype*	*Description*
Text	Plain	Unformatted
	HTML	HTML format (see Appendix C)
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but no order
	Digest	Similar to Mixed, but the default is message/RFC822
	Alternative	Parts are different versions of the same message
Message	RFC822	Body is an encapsulated message
	Partial	Body is a fragment of a bigger message
	External-Body	Body is a reference to another message
Image	JPEG	Image is in JPEG format
	GIF	Image is in GIF format
Video	MPEG	Video is in MPEG format
Audio	Basic	Single channel encoding of voice at 8 KHz
Application	PostScript	Adobe PostScript
	Octet-stream	General binary data (eight-bit bytes)

Content-Transfer-Encoding This header defines the method used to encode the messages into 0s and 1s for transport.

The five types of encoding methods are shown in Table 5.5.3.

Table 5.5.3Methods for Content-Transfer-Encoding



Type	*Description*
7-bit	NVT ASCII characters with each line less than 1000 characters
8-bit	Non-ASCII characters with each line less than 1000 characters
Binary	Non-ASCII characters with unlimited-length lines
Base64	6-bit blocks of data encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters encoded as an equal sign plus an ASCII code

In the Base 64 encoding, data, as a string of bits, is first divided into 6-bit chunks. Each 6-bit section is then converted into an ASCII character.