UNIT V
APPLICATION LAYER

1.ELECTRONIC MAIL (or) SMTP (simple mail transfer protocol)

Electronic Mail

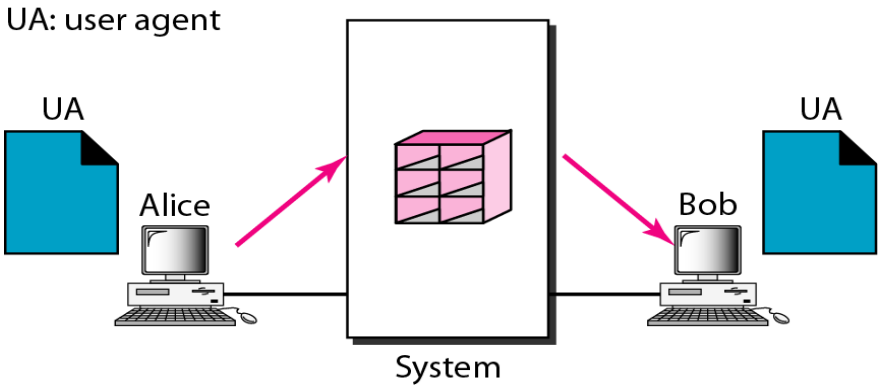
Electronic mail allows a message to include text, audio, and video. It also allows one message to be sent to one or more recipients. e-mail system has three main components:

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

Email Architecture

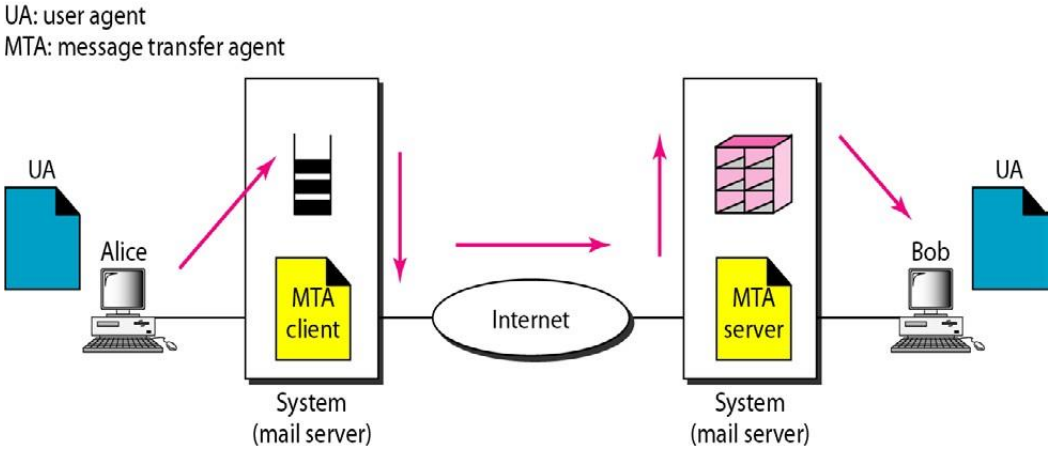
1. First Scenario - sender and the receiver on the same system
2. Second Scenario - sender and the receiver on two different systems.
3. Third Scenario – sender(or Rx) directly connected to his system and receiver(or Tx) separated from system.
4. Fourth common scenario – sender and receiver is connected to mail server by a WAN or a LAN and uses an MAA (message access agents) to retrieve messages.

First scenario in electronic mail



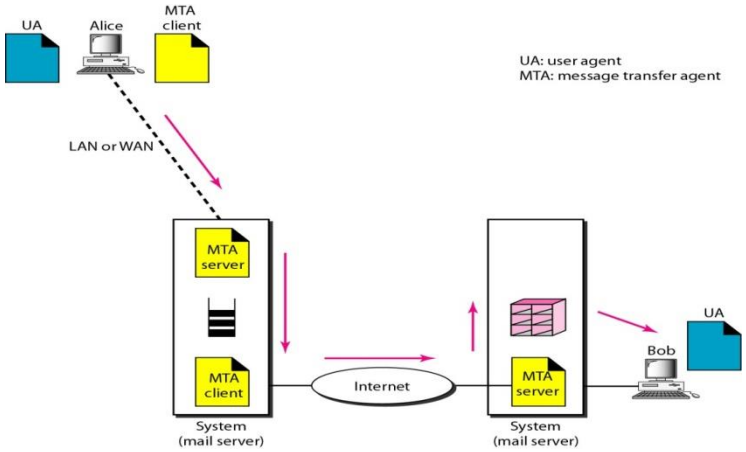
When the sender and the receiver of an e-mail are on the same system, we need only two user agents

Second scenario in electronic mail



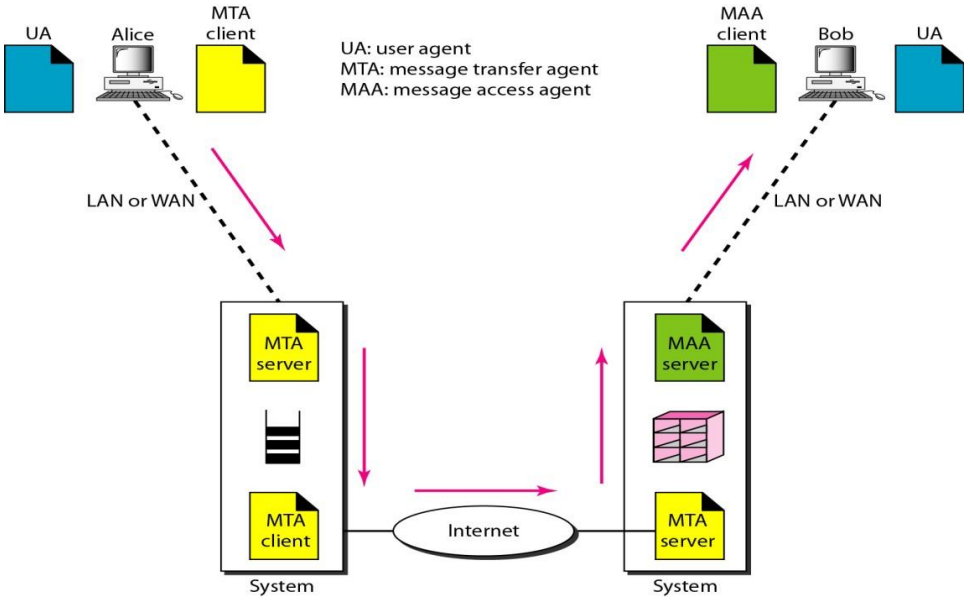
When the sender and the receiver of an e-mail are on different systems, we need two UAs and a pair of MTAs (client and server).

Third scenario in electronic mail



When the sender/ receiver is connected to the mail server via a LAN or a WAN, we need two UAs and two pairs of MTAs (client and server).

Fourth scenario in electronic mail



When both sender and receiver are connected to the mail server via a LAN or a WAN, we need two UAs, two pairs of MTAs and a pair of MAAs.

E – mail:

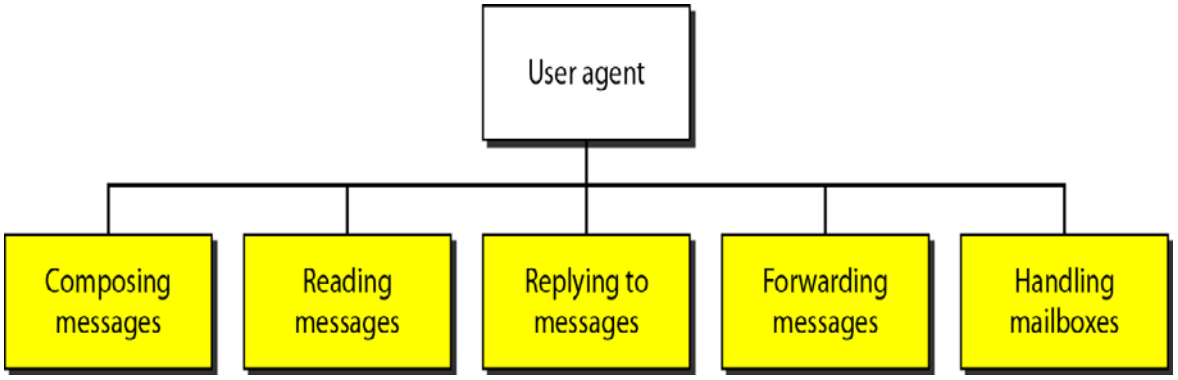
One of the most popular Internet services is electronic mail (e-mail).

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.

Services Provided by a User Agent

- A user agent is a software package that composes, reads, replies to, and forwards messages.
- It also handles mailboxes.
- Figure shows the services of a typical user agent.



Composing Messages

- A user agent helps the user 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 second duty of the user agent is to read the incoming messages.
- When a user invokes a user agent, it first checks the mail in the incoming mailbox.
- Most user agents show a one-line summary of each received mail.
- 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.
- The reply message may contain the original message and the new message.

Forwarding Messages

- *Forwarding* is defined as sending the message to a third party.
- A user agent allows the receiver to forward the message, with or without extra comments, to a third party.

Handling Mailboxes

- A user agent normally creates two mailboxes: an inbox and an outbox.
- The inbox keeps all the received e-mails until they are deleted by the user.
- The 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.

1. **Command-Driven**
   - A command-driven user agent normally accepts a one-character command from the keyboard to perform its task.
   - For example, a user can type the character r, at the command prompt, to reply to the sender of the message, or type the character R to reply to the sender and all recipients.

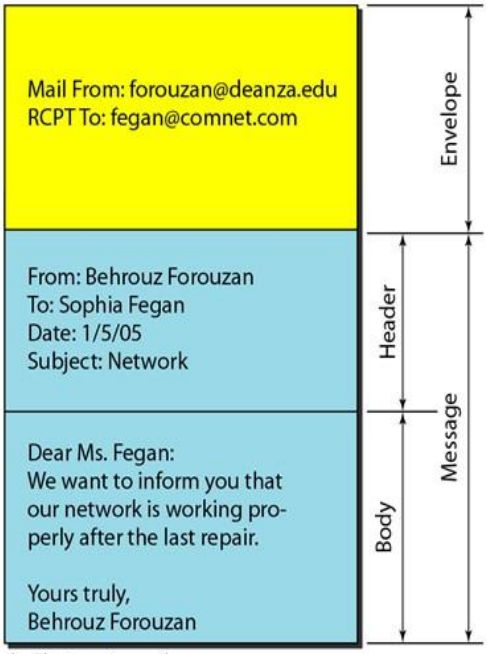
2. GUI-Based

- 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.
- Some examples of GUI-based user agents are Eudora, Microsoft's Outlook, and Netscape.

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*



b. Electronic mail

Envelope

The envelope usually contains the sender and the receiver addresses.

Message

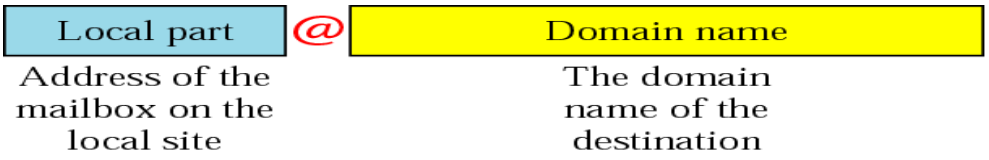
- The message contains the header and the body.
- The header of the message defines the sender, the receiver, the subject of the message,
- The body of the message contains the actual information to be read by the recipient.

Receiving Mail

- 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.
- The user can select any of the messages and display its contents on the screen.

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 .

Local Part

The local part defines the name of a special file, called the user mailbox, where all the mail received for a user is stored for retrieval by the message access agent.

Domain Name

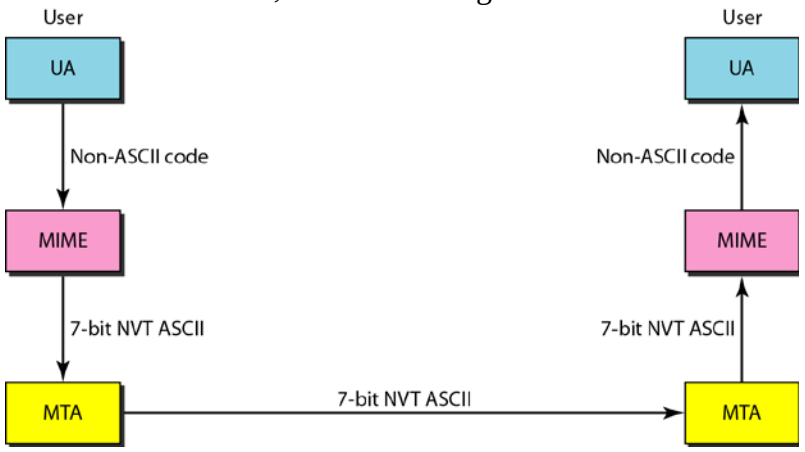
An organization usually selects one or more hosts to receive and send e-mail; the hosts are sometimes called *mail servers* or *exchangers.*

MIME (Multipurpose Internet Mail Extensions)

Electronic mail has a simple structure. It can send messages only in NVT 7-bit ASCII format. For example, it cannot be used for languages that are not supported by 7-bit ASCII characters (such as French, German, Hebrew, Russian, Chinese, and Japanese). Also, it cannot be used to send binary files or video or audio data.

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 them to the client MTA to be sent through the Internet. The message at the receiving side is transformed back to the original data.

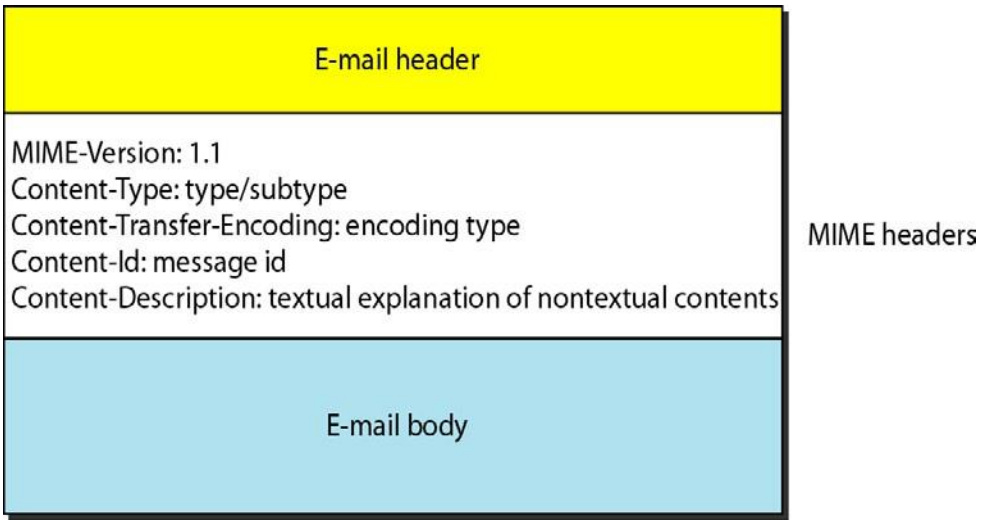
MIME is a set of software functions that transforms non-ASCII data (stream of bits) to ASCII data and vice versa, as shown in Figure



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

1. MIME-Version
2. Content-Type
3. Content-Transfer-Encoding
4. Content-Id
5. Content-Description

Figure shows the MIME headers



MIME-Version This header defines the version of MIME used. 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.

Content-Type: <type /subtype; parameters>

MIME allows seven different types of data.

Type	*Subtype*	*Description*
Text	Plain	Unformatted
	HTML	HTML format (see Chapter 27)
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but no order
	Digest	Similar to mixed subtypes, 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	IPEG	Image is in IPEG 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 (8-bit bytes)

Content-Transfer-Encoding This header defines the method used to encode the messages into Os and Is for transport:

Content-Transfer-Encoding: <type>

The five types of encoding methods

Type	*Description*
7-bit	NVT ASCII characters and short lines
8-bit	Non-ASCII characters and short lines
Binary	Non-ASCII characters with unlimited-length lines
Base-64	6-bit blocks of data encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters encoded as an equals sign followed by an ASCII code

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

1. MIME-Version
2. Content-Type
3. Content-Transfer-Encoding
4. Content-Id
5. Content-Description

Content-Id This header uniquely identifies the whole message in a multiple-message environment

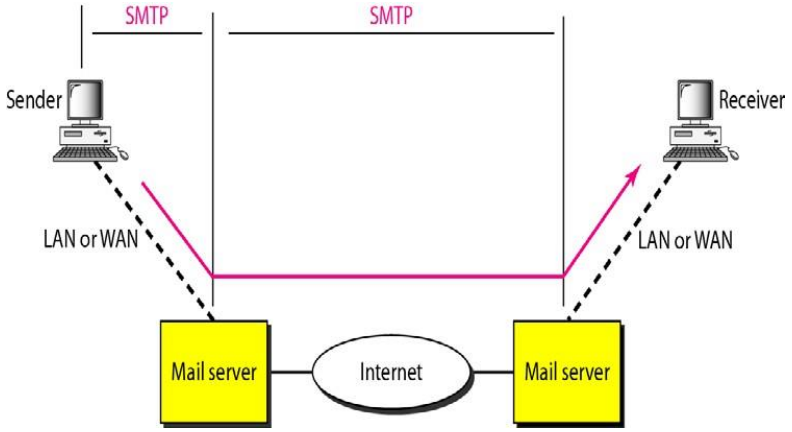
Content-Id: id=<content-id>

Content-Description This header defines whether the body is image, audio, or video.

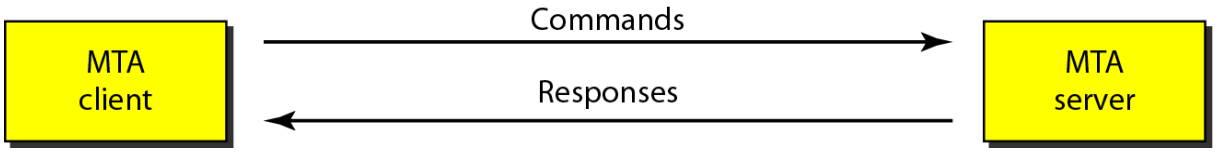
Content-Description: <description>

Simple Mail Transfer Protocol (*SMTP*) Architecture:

The actual mail transfer is done through message transfer agents. To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA. The formal protocol that defines the MTA client and server in the Internet is called the Simple Mail Transfer Protocol (SMTP). Figure shows the range of the SMTP protocol in this scenario.



Commands and responses



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.

- SMTP defines 14 commands.
- The first **five** are mandatory; every implementation must support these five commands.
- The next **three** are often used and highly recommended.

Keyword	*Argument(s)*
HELO	Sender's host name
MAIL FROM	Sender of the message
RCPT TO	Intended recipient of the message
DATA	Body of the mail
QUIT	
RSET	
VRFY	Name of recipient to be verified
NOOP	
TURN	
EXPN	Mailing list to be expanded
HELP	Command name

Command format

Responses

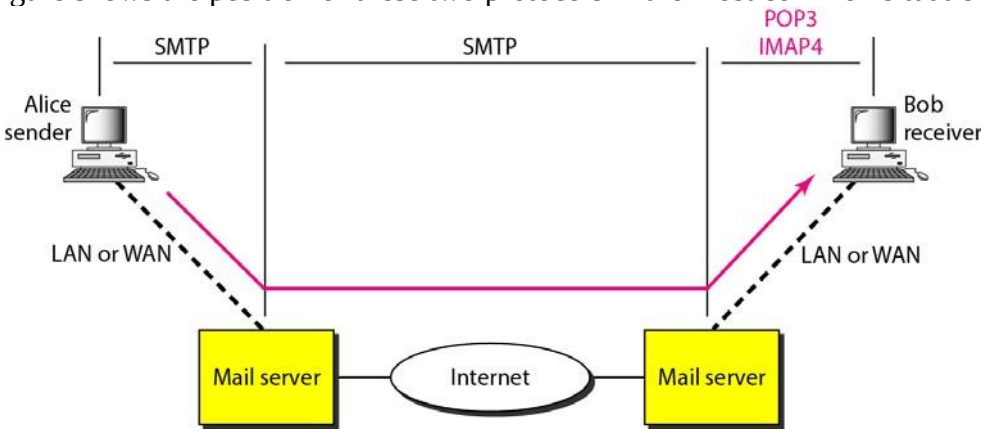
Responses are sent from the server to the client. A response is a three digit code that may be followed by additional textual information

Permanent Negative Completion Reply	
500	Syntax error; unrecognized command
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command temporarily not implemented
550	Command is not executed; mailbox unavailable
551	User not local
552	Requested action aborted; exceeded storage location
553	Requested action not taken; mailbox name not allowed
554	Transaction failed

Message Access Agent: POP and IMAP

The first and the second stages of mail delivery use SMTP. However, SMTP is not involved in the third stage because SMTP is a *push* protocol; it pushes the message from the client to the server. On the other hand, the third stage needs a *pull* protocol; the client must pull messages from the server. The third stage uses a message access agent. Currently two message access protocols are available: Post Office Protocol, version 3 (POP3) and Internet Mail Access Protocol, version 4 (IMAP4).

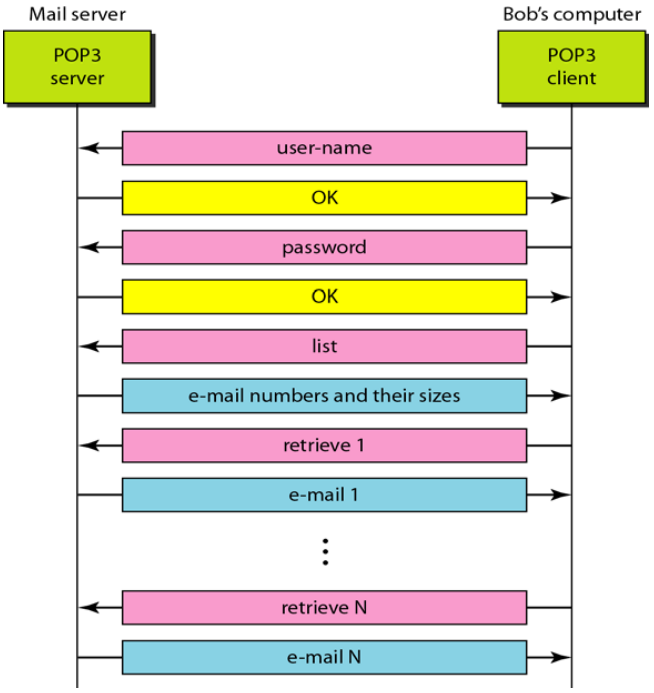
Figure shows the position of these two protocols in the most common situation



Post Office Protocol (POP3)

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 then list and retrieve the mail messages, one by one.

The exchange of commands and responses in 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. 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.

.

- In the **keep mode**, the mail remains in the mailbox after retrieval. The keep mode is normally used when the user accesses her mail away from her primary computer. The mail is read but kept in the system for later retrieval and organizing. (e.g., a laptop).

Limitations of POP3

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

Internet Mail Access Protocol) *IMAP4*

Another mail access protocol 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:

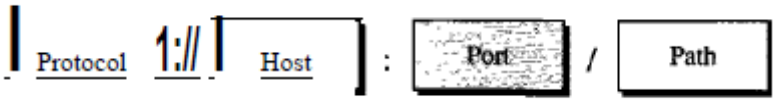
- A user can check the e-mail header prior to downloading.
- A user can search the contents of the e-mail for a specific string of characters prior to downloading.
- A user can partially download e-mail. This is especially useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth 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

2. HTTP (Hyper Text Transfer Protocol)

The Hypertext Transfer Protocol (HTTP) is a protocol used mainly to access data on the World Wide Web. HTTP functions as a combination of FTP and SMTP. It is similar to FTP because it transfers files and uses the services of TCP.

Uniform Resource Locator

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. The uniform resource locator (URL) is a standard for specifying any kind of information on the Internet. The URL defines four things: protocol, host computer, port, and path.



The ***protocol*** is the client/server program used to retrieve the document. Many different protocols can retrieve a document; among them are FTP or HTTP. The most common today is HTTP. The ***host*** is the computer on which the information is located. The URL can optionally contain the port number of the server. If the ***port*** is included, it is inserted between the host and the path, and it is separated from the host by a colon.

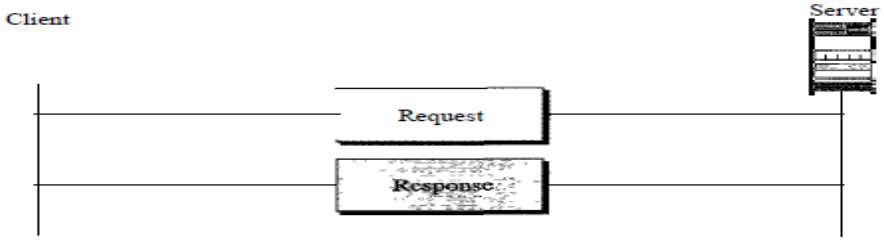
Path is the pathname of the file where the information is located. Note that the path can itself contain slashes that, in the UNIX operating system, separate the directories from the subdirectories and files.

HTTP vs HTML

- HTML: hypertext markup language
  - Definitions of tags that are added to Web documents to control their appearance
- HTTP: hypertext transfer protocol
  - The rules governing the conversation between a Web client and a Web server

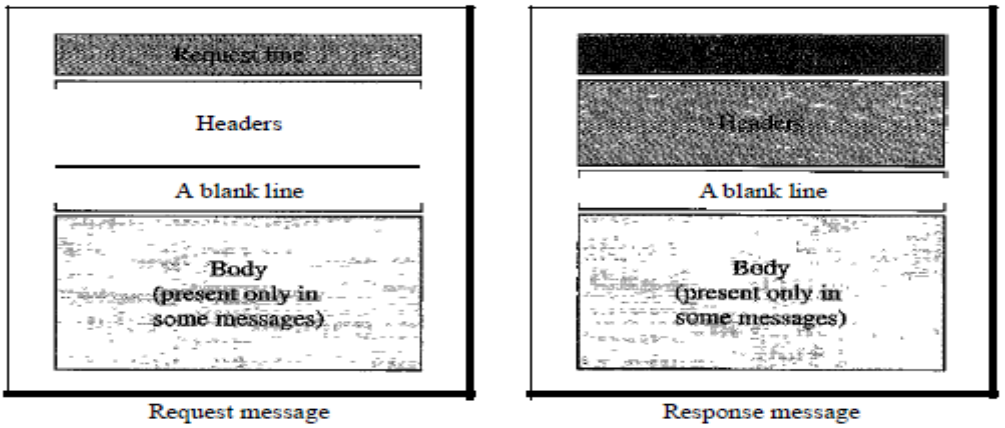
HTTP Transaction

The client initializes the transaction by sending a request message. The server replies by sending a response.



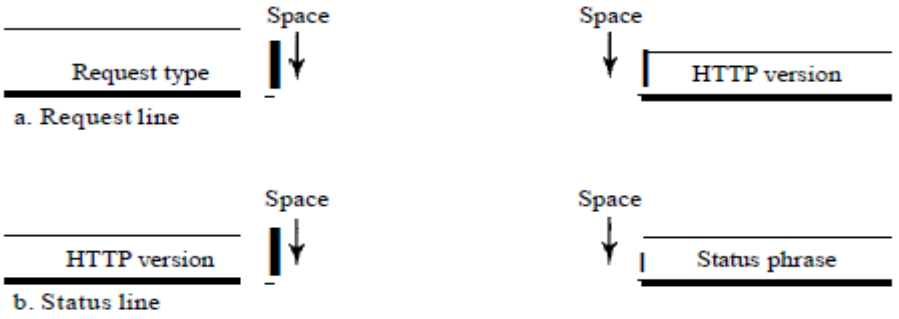
Messages

The formats of the request and response messages are similar. A request message consists of a request line, a header, and sometimes a body. A response message consists of a status line, a header, and sometimes a 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, several request types are defined. The request type is categorized into *methods.*

Method	*Action*
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
POST	Sends some information from the client to the server
PUT	Sends a document from the server to the client
TRACE	Echoes the incoming request
CONNECT	Reserved
OPTION	Inquires about available options

Version: The most current version of HTTP is 1.1.

Status code: This field is used in the response message. The status code field is similar to those in the FTP and the SMTP protocols. It consists of three digits. Whereas the codes in the 100 range are only informational, the codes in the 200 range indicate a successful request. The codes in the 300 range redirect the client to another URL, and the codes in the 400 range indicate an error at the client site. Finally, the codes in the 500 range indicate an error at the server site.

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

Code	*Phrase*	*Description*
	Informational	
100	Continue	The initial part of the request has been received, and the client may continue with its request.
101	Switching	The server is complying with a client request to switch protocols defined in the upgrade header.
	Success	
200	OK	The request is successful.
201	Created	A new URL is created.
202	Accepted	The request is accepted, but it is not immediately acted upon.
204	No content	There is no content in the body.
	Redirection	
301	Moved permanently	The requested URL is no longer used by the server.
302	Moved temporarily	The requested URL has moved temporarily.
304	Not modified	The document has not been modified.
	Client Error	
400	Bad request	There is a syntax error in the request.
401	Unauthorized	The request lacks proper authorization.
403	Forbidden	Service is denied.
404	Not found	The document is not found.
405	Method not allowed	The method is not supported in this URL.
406	Not acceptable	The format requested is not acceptable.
	Server Error	
500	Internal server error	There is an error, such as a crash, at the server site.
501	Not implemented	The action requested cannot be performed.
503	Service unavailable	The service is temporarily unavailable, but may be requested in the future.

Header: The header exchanges additional information between the client and the server. A header line belongs to one of four categories: general header, request header, response header, and entity header. A ***request message*** can contain only general, request, and entity headers. A ***response message***, on the other hand, can contain only general, response, and entity headers.

- ***General header:*** The general header gives general information about the message and can be present in both a request and a response.

Header	*Description*
Cache-control	Specifies infonnation about caching
Connection	Shows whether the connection should be closed or not
Date	Shows the current date
MIME-version	Shows the MIME version used
Upgrade	Specifies the preferred communication protocol

Request header: The request header can be present only in a request message. It specifies the client's configuration and the client's preferred document format.

Header	*Description*
Accept	Shows the medium fonnat the client can accept
From	Shows the e-mail address of the user

Response header: The response header can be present only in a response message. It specifies the server's configuration and special information about the request.

Header	*Description*
Accept-range	Shows if server accepts the range requested by client
Server	Shows the server name and version number

Entity header: The entity header gives information about the body of the document.

Header	*Description*
Allow	Lists valid methods that can be used with a URL
Content-encoding	Specifies the encoding scheme
Content-language	Specifies the language
Content-length	Shows the length of the document
Content-range	Specifies the range of the document
Content-type	Specifies the medium type
Etag	Gives an entity tag
Expires	Gives the date and time when contents may change
Last-modified	Gives the date and time of the last change
Location	Specifies the location of the created or moved document

Body: The body can be present in a request or response message.

Persistent Versus Nonpersistent Connection:

HTTP prior to version 1.1 specified a nonpersistent connection, while a persistent connection is the default in version 1.1.

Nonpersistent Connection:

In a nonpersistent connection, one TCP connection is made for each request/response.

The following lists the steps 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; it then closes the connection.

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

- HTTP version 1.1 specifies a persistent connection by default. In a persistent connection, the server leaves the connection open for more requests after sending a response.
- The server can close the connection at the request of a client or if a time-out has been reached. The sender usually sends the length of the data with each response. However, there are some occasions when the sender does not know the length of the data.
- This is the case when a document is created dynamically or actively. In these cases, 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.
- The proxy server reduces the load on the original server, decreases traffic, and improves latency. However, to use the proxy server, the client must be configured to access the proxy instead of the target server.

3. DNS (Domain Name System)

Domain Name System (DNS) is used to resolve human-readable hostnames like www.google.com into machine-readable IP addresses like 216.58.197.68. DNS is like a phone book for the Internet

History

During early days of internet, there were only few hundred hosts. A central authority called the Network Information Center (NIC) maintained name-to-address bindings in a flat-file called *hosts.txt*. A new host that joins the internet would mail its name and IP address to NIC. NIC updates *hosts.txt* and mails to all hosts. Name server resolved domain

names using a simple *lookup* on hosts.txt. As hosts grew to thousands and millions, the flat file approach failed, leading to evolution of DNS in mid 1980s.

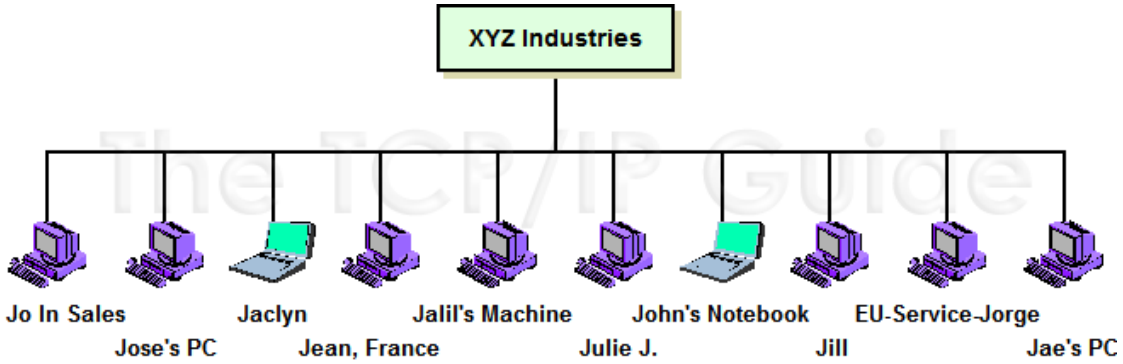
NAME SPACE

The names assigned to machines must be unique, the binding between the names and IP addresses should be perfect. A name space that maps each address to a unique name can be organized in two ways:

1. Flat
2. Hierarchical.

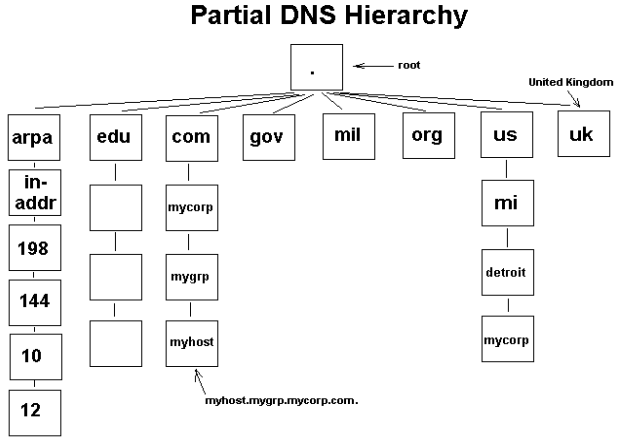
Flat Name Space

In a flat name space, a name is assigned to an address. A name in this space is a sequence of characters without structure. The names may or may not have a common section; if they do, it has no meaning. The main disadvantage of a flat name space is that it 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

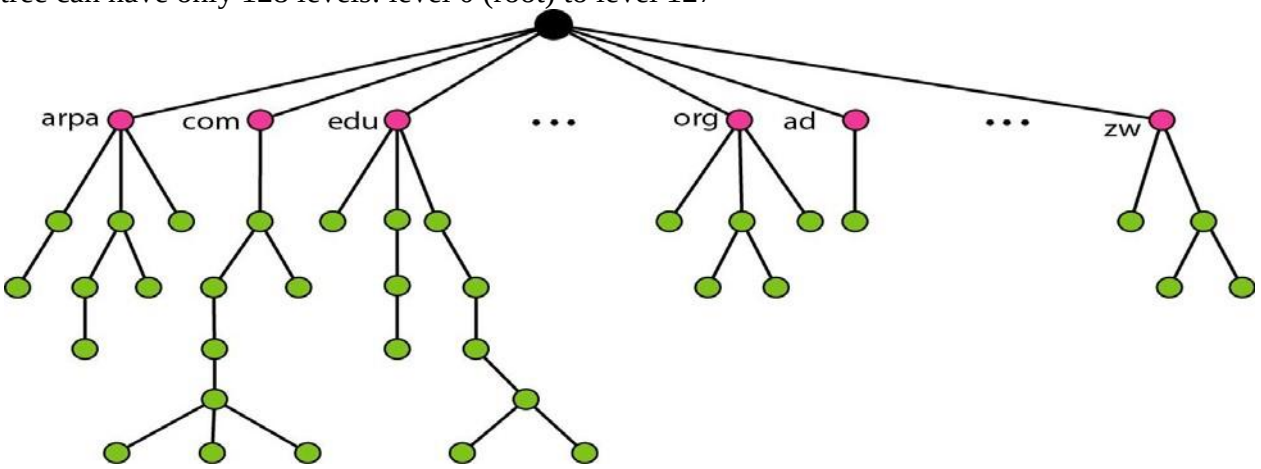
DNS uses hierarchical name space for domains in the Internet. Hierarchical naming permits use of same sub-domain name in different domains. Domain names are case insensitive and can be up to 63 characters. DNS names are processed from right to left
and use periods (.) as separator. DNS can be used to map names to values, not necessarily domain names to IP address.



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 can have only 128 levels: level 0 (root) to level 127



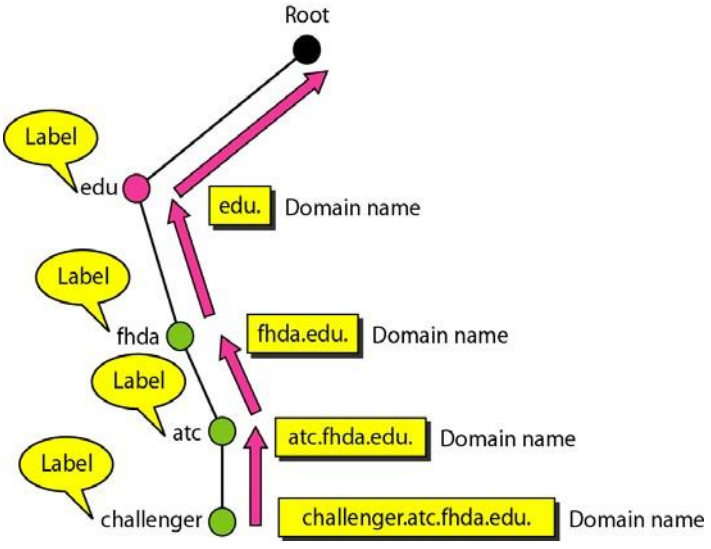
Label

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 that (nodes that branch from the same 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 (.). The domain names are always read from the node up to the root.

Domain names and labels



Fully Qualified Domain Name(FQDN)

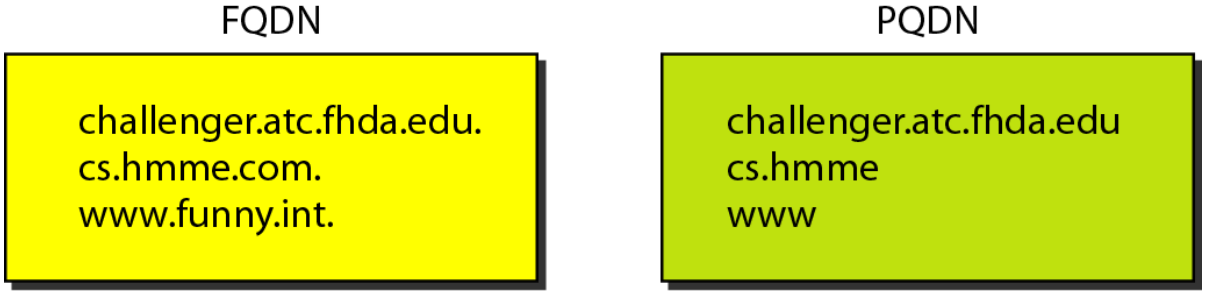
If a label is terminated by a null string, it is called a fully qualified domain name(FQDN). An FQDN is a domain name that contains the full name of a host. It contains all labels, from the most specific to the most general, that uniquely define the name of the host. For example, the domain name **challenger.ate.tbda.edu.** is the FQDN of a computer named *challenger* installed at the Advanced Technology Center (ATC) at De Anza College. A DNS server can only match an FQDN to an address.

Note that the name must end with a null label, but because null means nothing, the label

ends with a dot (.)

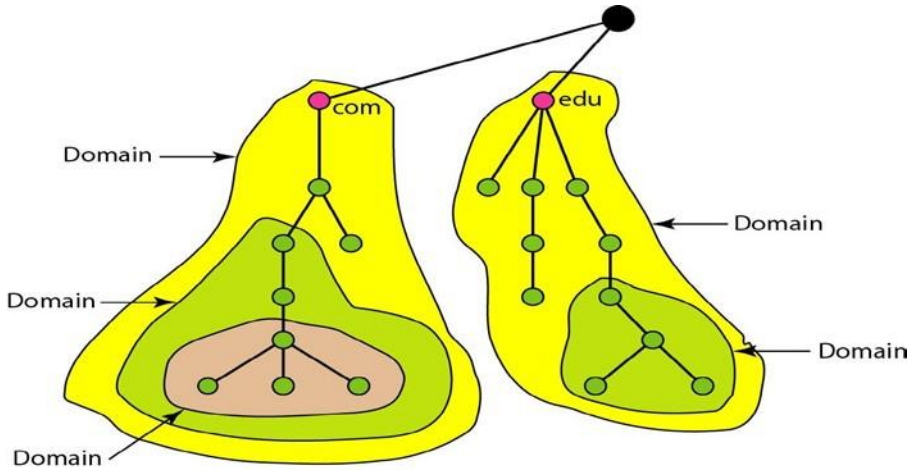
Partially Qualified Domain Name (PQDN)

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.



Domain

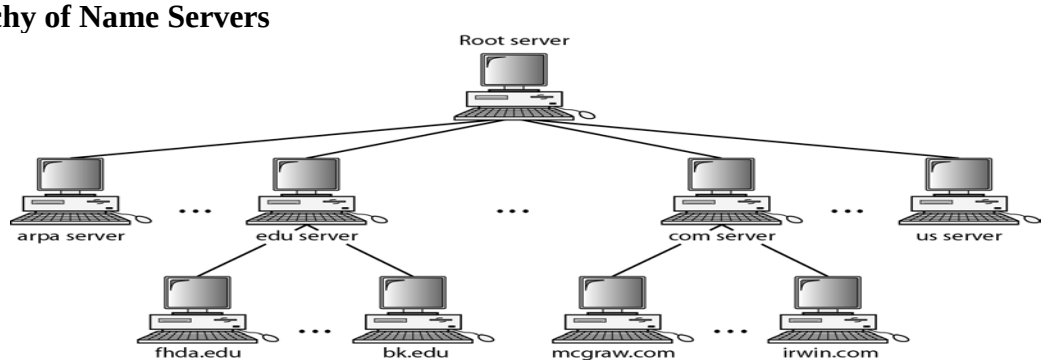
A **domain** is a subtree of the domain name space. The name of the domain is the domain name of the node at the top of the subtree.



Distribution of Name Space

The information contained in the domain name space must be stored. However, it is very inefficient and also unreliable 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. It is not unreliable because any failure makes the data inaccessible. The solution to these problems is to distribute the information among many computers called DNS servers. One way to do this is to divide the whole space into many domains based on the first level.

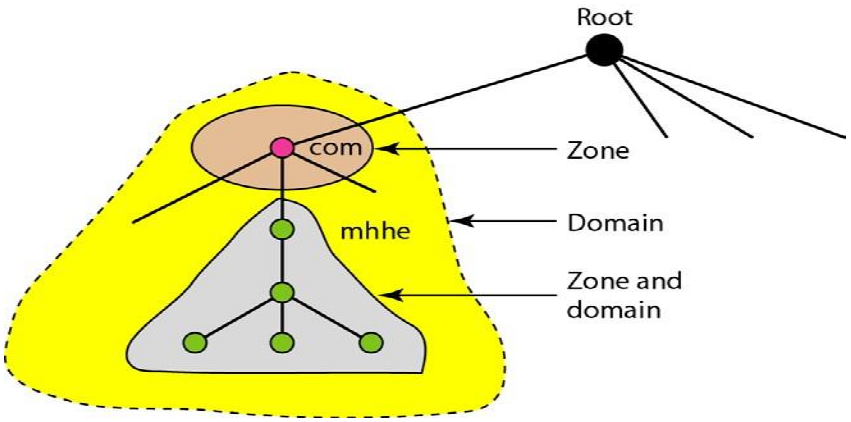
Hierarchy of Name Servers



Zone

Since the complete domain name hierarchy cannot be stored on a single server, it is divided among many servers What a server is responsible for or has authority over is called a zone. We can define a zone as a contiguous part of the entire tree. The domain hierarchy is partitioned into *zones*. Topmost domains are managed by NIC. Each zone acts as *central* authority for that part of the sub-tree. Each zone can be further sub-divided that manage using their own name servers

Zones and Domains



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. There are several root servers, each covering the whole domain name space. The servers are distributed all around the world.

Primary and Secondary Servers

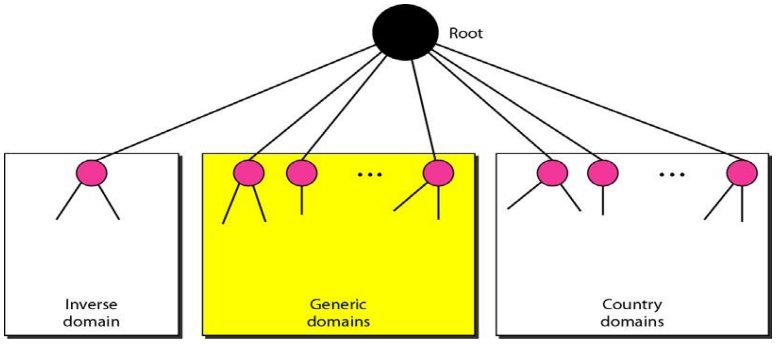
DNS defines two types of servers: primary and secondary. A primary server is a server that stores a file about the zone for which it is an authority. It is responsible for creating, maintaining, and updating the zone file. It stores the zone file on a local disk. A secondary server is a server that transfers 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.

DNS in the Internet

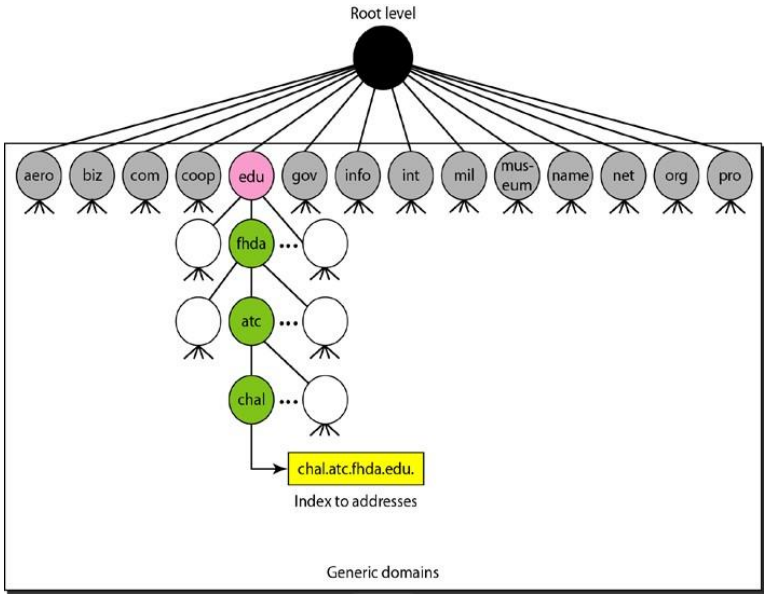
DNS is a protocol that can be used in different platforms. In the Internet, the domain name space (tree) is divided into three different sections:

1.Generic Domains
2.Country Domains
3.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

Label	***Description***
aero	Airlines and aerospace companies
biz	Businesses or firms (similar to "com")
com	Commercial organizations
coop	Cooperative business organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers
int	International organizations
mil	Military groups
museum	Museums and other nonprofit organizations
name	Personal names (individuals)
net	Network support centers
org	Nonprofit organizations
pro	Professional individual organizations

Country Domains

The country domains section uses two-character country abbreviations.

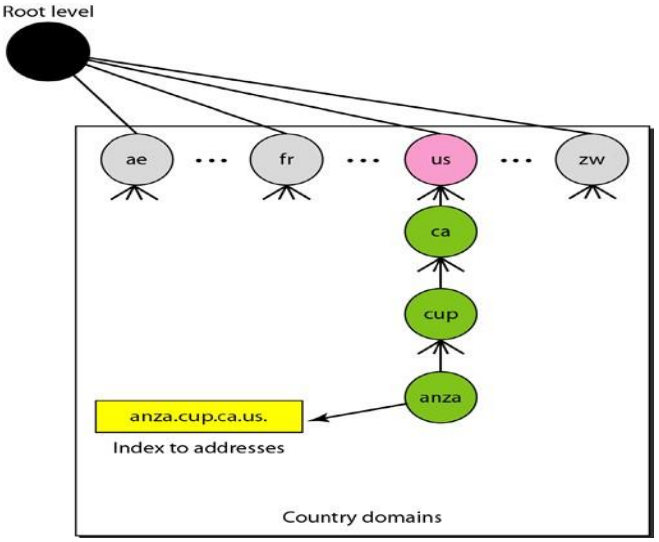
United States- .us

India - .in

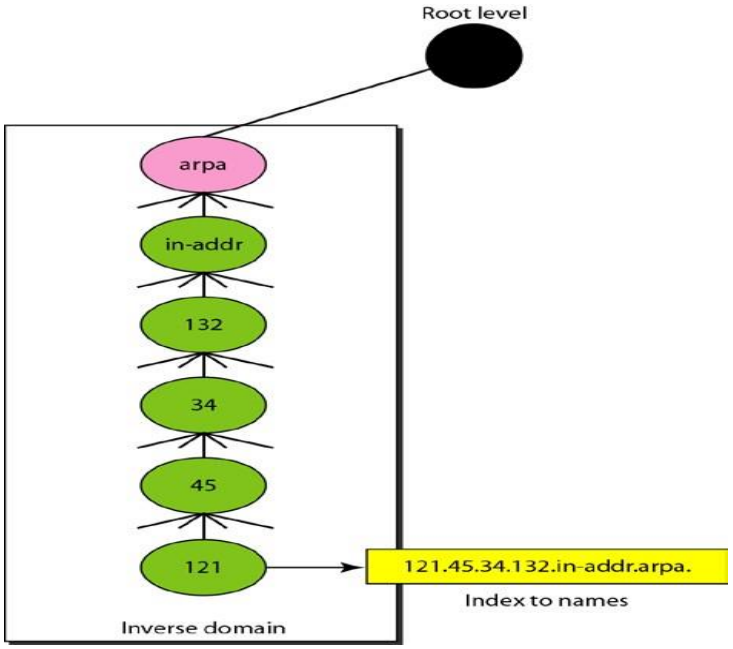
Inverse Domain

The inverse domain is used to map an address to a name. This may happen when a server has received a request from a client to do a task.

Country domains



Inverse domain



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 satisfies the resolver; otherwise, it either refers the resolver to other servers or asks other servers to provide the information.

Mapping Names to Addresses

Most of the time, the resolver gives a domain name to the server and asks for the corresponding address. In this case, the server checks the generic domains or the country

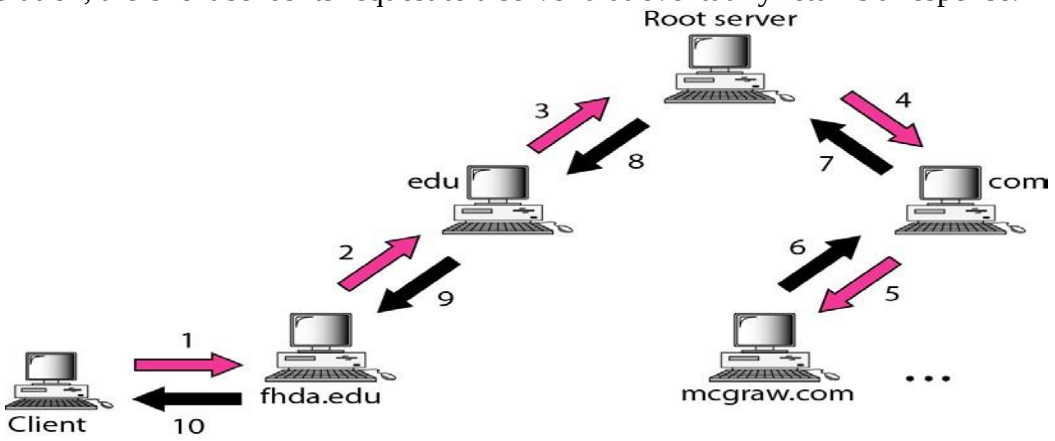
domains to find the mapping. If the domain name is from the generic domains section the query is sent by the resolver to the local DNS server for resolution. If the local server cannot resolve the query, it either refers the resolver to other servers or asks other servers directly.

Mapping Addresses to Names

A client can send an IP address to a server to be mapped to a domain name. To answer queries of this kind, DNS uses the inverse domain. However, in the request, the IP address is reversed and the two labels *in-addr* and *arpa* are appended to create a domain acceptable by the inverse domain section.

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 usually) 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 is called recursive resolution. In Recursive Resolution, the client sends its request to a server that eventually returns a response.



Iterative Resolution

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 it is not, it returns (to the client) the IP address of the server that it thinks can resolve the query. The client is responsible for repeating the query to this second server. Now the client must repeat the query to the server. This process is called iterative resolution because the client repeats the same query to multiple servers. In Iterative Resolution , the client may send its request to multiple servers before getting an answer.

Caching

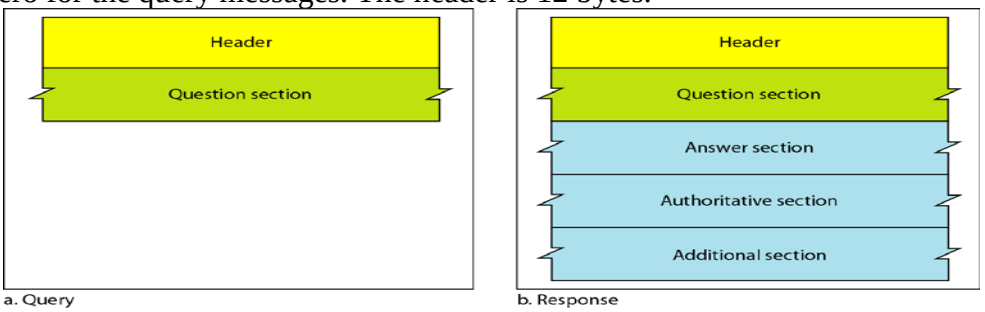
Caching is a method whereby an answer to a query is stored in memory (for a limited time) for easy access to future requests. If a server caches a mapping for a long time, it may send an outdated mapping to the client. To counter this, two techniques are used. First, the authoritative server always adds information to the mapping called *time-to-live* (TTL). It defines the time in seconds that the receiving server can cache the information. After that time, the mapping is invalid and any query must be sent again to the authoritative server.. Second, DNS requires that each server keep a TTL counter for each mapping it caches. The cache memory must be searched periodically, and those mappings with an expired TTL must be purged.

DNS MESSAGES

DNS has two types of messages: **query and response.** 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

Both query and response messages have the same header format with some fields set to zero for the query messages. The header is 12 bytes.



a. Query b. Response

Question Section

This is a section consisting of one or more question records. It is present on both query and response messages. We will discuss the question records in a following section.

Answer Section

This is a section consisting of one or more resource records. It is present only on response messages. This section includes the answer from the server to the client (resolver).

Authoritative Section

This is a section consisting of one or more resource records. It is present only on response messages. This section gives information (domain name) about one or more authoritative servers for the query.

Additional Information Section

This is a section consisting of one or more resource records. It is present only on response messages. This section provides additional information that may help the resolver. For 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

Question Record

A question record 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. Resource records are also what is returned by New domains are added to DNS through a registrar, a commercial entity accredited by ICANN. A registrar first verifies that the requested domain name is unique and then enters it into the DNS database.

DYNAMIC DOMAIN NAME SYSTEM (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. The size of today's Internet does not allow for this kind of manual operation. The DNS master file must be updated dynamically. The Dynamic Domain Name System (DDNS) automatically updates the DNS master file.

Client Server Model:

* Most important and most widely used distributed system architecture.

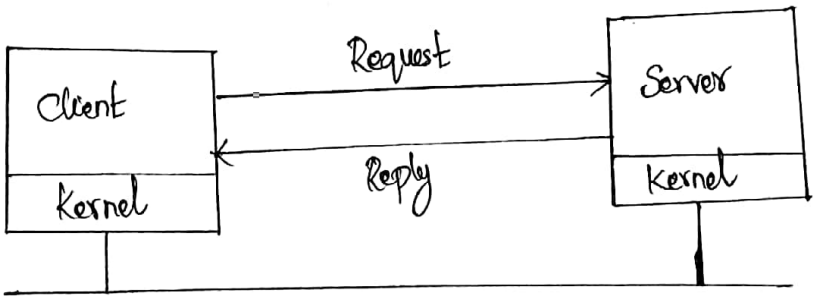
* Client & Server roles are assigned and changable.

* In Client Server model, any process can act as server or client. It is not the type of machine, size of machine or its computing power which makes it server.

* It is the ability of server request that makes a machine, a server. A system can acts as client and server simultaneously ie) one act as server other as client.

* A Server host runs one (or) more server programs which share their resources with clients. A client does not share any of its resources but requests a server's content or service function.

* Examples of client Server model are email, network printing, world wide web.



* Client and Server is having a kernel. kernel is important component in operating system which is capable of managing all devices. client is going to request and server has to respond. When client sends the request, the server has to provide

the service, the time the server is replying, the client has to wait for result.

Client and Server Communication:

* Clients and Servers exchange messages in a request-response messaging pattern. The client sends a request and the server returns a response. This exchange of message is an example of inter-process communication.

* To communicate, the computers must have a common language and they must follow rules so that both the client and the server know what to expect. The language and rules of communication are defined in the communication protocol. All client server protocol operate in the application layer.

* The application layer protocol defines the basic patterns of the data exchange. To formulate the data exchange even further, the server may implement an application programming interface (API).

* API is an abstraction layer for accessing a service. By abstracting access, it facilitates cross platform data exchange.

Application Programming Interface (Sockets):

* API is like a socket interface which can be ported to other operating systems also.

* Socket is a point where an application process

connects to a network. Both client and server establish their own socket.

Steps involved in establishing a socket on client side,

* Create a socket with socket system call.
* Connect the socket to the address of the server using connect system call.
* Send and receive data.

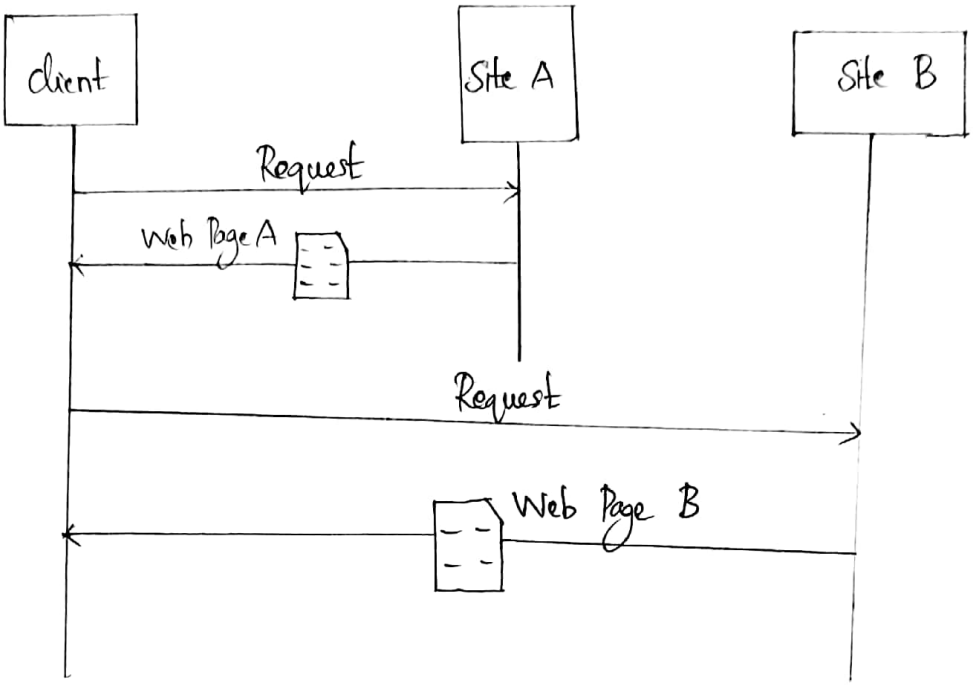
Steps involved in establishing a socket on server side,

* Create a socket with socket system call.
* Bind socket to an address using bind system call.
* Listen for connections with listen system call.
* Accept connection with accept system call. This call typically blocks until a client connects with server.
* Send and receive data.

World Wide Web (WWW)

* World Wide Web is an information space where documents and other web resources are identified by Uniform Resource Locators (URL), interlinked by hypertext links and can be accessed via the Internet.

* WWW is a distributed client/server service in which a client using a browser can access a service using a server. However the service provided is distributed over many location called sites.

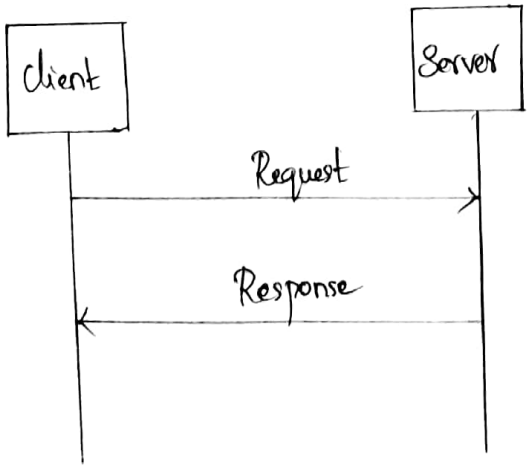


* Web Pages are primarily text documents formatted and annotated with Hypertext Markup Language (HTML). In addition to formatted text, web pages may contain images, video, audio and software components.

Components of WWW:

1. HTTP: (Hyper Text Transport Protocol)

* Web Pages are organized and retrieved information using HTTP protocol. HTTP is an application protocol that is used to retrieve web pages from remote servers.



* HTTP is the protocol to exchange or transfer hypertext.

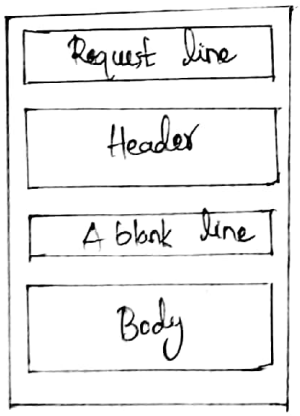
* An HTTP Session is a sequence of network request-response transactions. An HTTP client initiates a request by establishing a TCP connection to a particular port on a server.

Message format of HTTP:

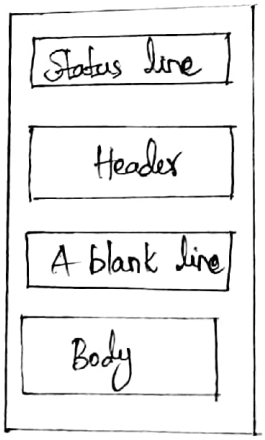
* The client and server communicate by sending plain text (ASCII) messages. The client sends requests to the server and the server sends responses.

The request message consists of

- → A request line
- → Request header files
- → An empty line
- → An optional message body.



Request Message



Response Message

The response message consists of

→ A status line, which includes the status code and reason message, eg. HTTP/1.1 200 OK, which indicates that the client's request succeeded.

→ Response header fields
→ An empty line
→ An optional message body.

2. URL

* User access web page by opening a URL. A Uniform Resource Locator also called as web address, which specifies the location of the file.

* Most web browsers display URL above the page in the address bar.

WWW Architecture:

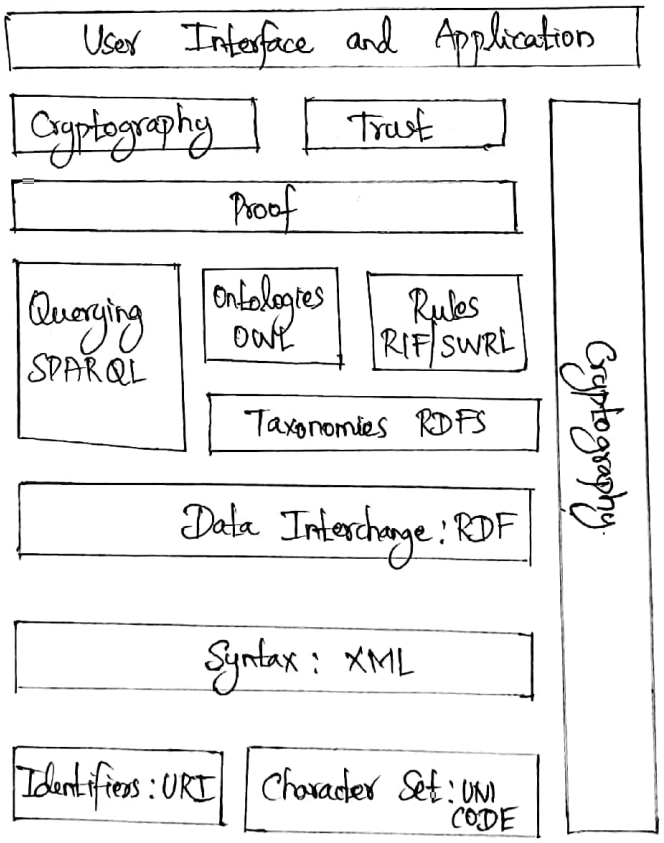


Figure: Architecture of WWW

1. Identifiers and Character Set:

URI is used to identify resources on the web and UNICODE makes it possible to build web pages that can be read and write in human languages.

2. Syntax:

XML (Extensible Markup Language) helps to define common syntax in semantic web.

3. Data Interchange:

Resource Description Framework (RDF) framework helps in defining core representation of data for web. RDF represents data about resource in graph form.

4. Taxonomies:

RDF Scheme (RDFS) allows more standardized description of taxonomies and other ontological constructs.

5. Ontologies:

Web Ontology Language (OWL) offers more constructs over RDFS.

- → OWL Lite for taxonomies and simple constraints.
- → OWL DL for full description logic support.
- → OWL for more syntactic freedom of RDF.

6. Rules:

RIF and SWRL offers rules beyond the constructs that are available from RDF's and OWL. Simple protocol and RDF Query language (SPARQL) is SQL like language used for quering RDF data and OWL ontologies.

7. Proof:

All semantic and rules that are executed at layers below proof and their result will be used to prove deductions.

8. Cryptography:

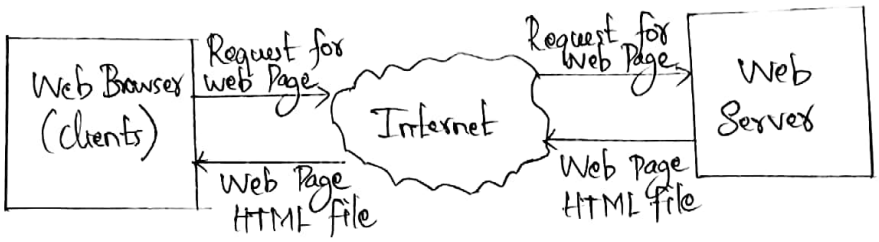
Cryptography means such as digital signature for

Verification of the origin of sources is used.

9. User Interface and Applications

On the top of layer User Interface and application layer is built for user interaction.

WWW Operation:



WWW works on client-server approach.

* User enters the URL of the web page in the address bar of web browser.
* Then browser requests the Domain Name Server for the IP addresses corresponding to URL.
* After receiving IP address, browser sends the request for web page to the web server using HTTP protocol which specifies the way the browser and web server communicates.
* Then web Server receives request using HTTP protocol and checks its search for the requested web page. If found it returns it back to the web browser and close the HTTP Connection.
* Now the web browser receives the web page, it interprets it and displays the contents of web page in web browser's window.

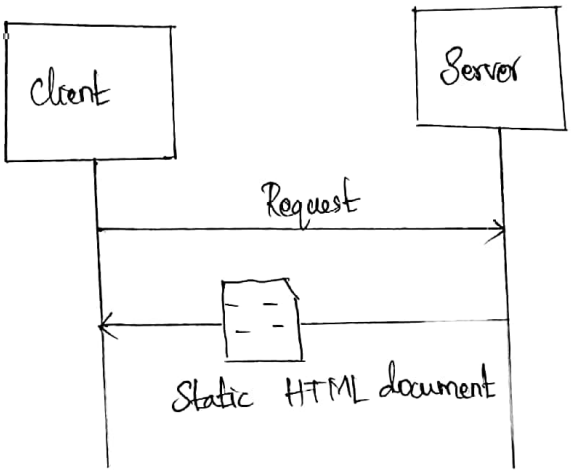
Web Documents:

WWW can group web documents into three categories.

* Static
* Dynamic
* Active

* Static Web Document:

* A Static web document resides in a file that it is associated with a web server
* The author determines the content at the time the document is written. Because the contents do not change, each request for a static document results in same response.



Advantages:

* Simple, reliable
* The browser can place a copy in a cache on a local disk.

Disadvantages:

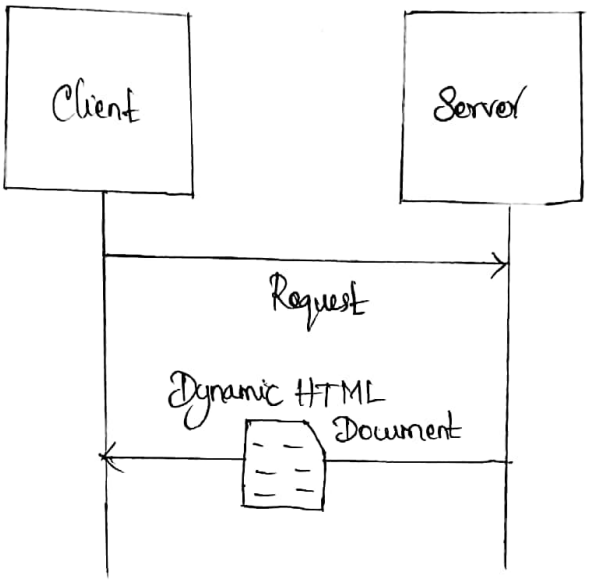
* Changes are time consuming because they require human to edit the file.

* Dynamic Web Document:

* Dynamic web document does not exist in predefined

form. When a request arrives the web server runs an application program that creates the document.

* Fresh document is created for each request, the contents of a dynamic document can vary from one request to another.



Advantages:

* Ability to report current information.

Disadvantages:

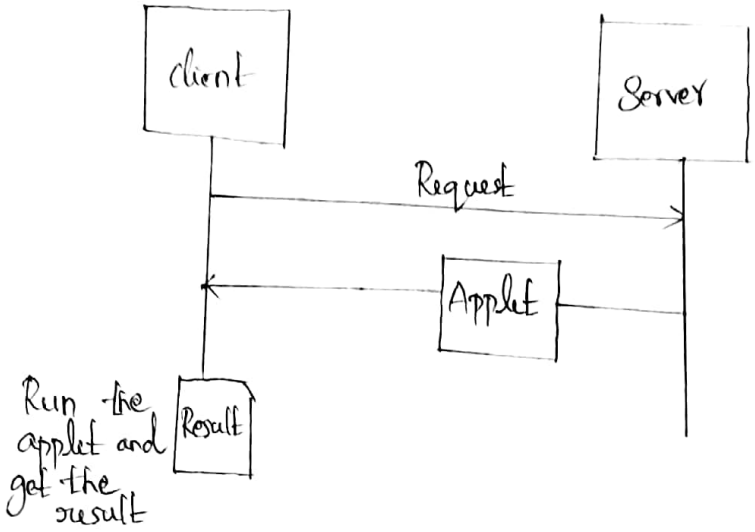
* Increased cost.

* Active Web Document:

* An active web document consists of a computer program that the server sends to the browser and that the browser must run locally.

* When it runs, the active document program can interact with the user and change the display continuously.

* The active documents are written in source code.



Advantages:

* Ability to update information continuously.

Disadvantages:

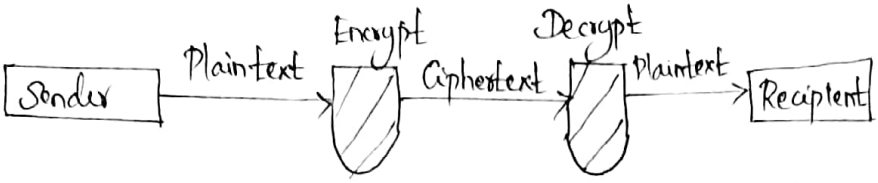
* Involves risk because the document can export or import information.

Cryptography:

* It is the practise and study of techniques for secure communication in the presence of third parties.
* Cryptography refers to the science and art of transforming messages to make them secure and immune to attacks.

Applications

* E-commerce
* Chip based payment card
* Digital Currencies
* Computer password
* Military Communication



* Original message before being transformed is called plaintext. After the message is transformed is called ciphertext.

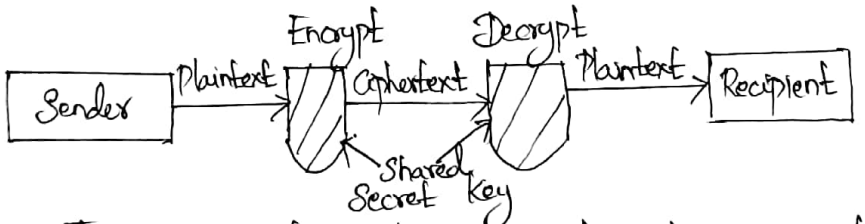
* An encryption algorithm transforms the plaintext to ciphertext, a decryption algorithm transforms the ciphertext back to plaintext. The term cipher is used to refer to encryption and decryption algorithms.

Categories of Cryptography:

Divided into two types

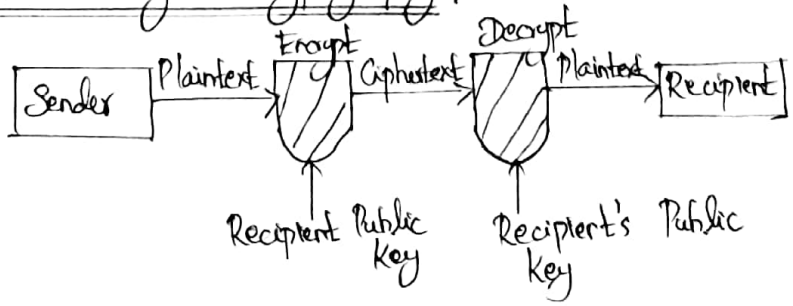
* Symmetric key (Secret key)
* Asymmetric key (Public key)

Symmetric Key Cryptography:



* In symmetric key cryptography, same key is used by both parties. The sender uses the key and an encryption algorithm to encrypt data, the receiver uses the same key and the corresponding decryption algorithm to decrypt the data.

Asymmetric Key Cryptography:



* In asymmetric or public key cryptography, there are two keys, a private key and a public key. The private key is kept by the receiver. The public key is announced to the public.

Traditional Ciphers:

Traditional Ciphers is divided into two broad categories

1. Substitution Ciphers — Mono alphabetic, Poly alphabetic
2. Transposition Ciphers

Substitution Ciphers:

* Substitution Cipher substitutes one symbol with another. If the symbols in the plain text are alphabetic characters, we replace one character with another.

Two types

1. Mono alphabetic
* Poly Alphabetic

Monoalphabetic:

* In monoalphabetic, a character or symbol in the plaintext is always changed to same character or symbol in the ciphertext regardless of its position in the text.

Polyalphabetic:

* In polyalphabetic, each occurance of a character can have a different substitute. Eg) Character A can be changed

to D in the beginning of the text and can be changed to N at the middle of the text.

Transposition Cipher:

* In transposition cipher, there is no substitution of characters, instead their location changes.
* A character in the first position of the plaintext may appear in the 10th position of the ciphertext. The transposition cipher reorders the symbols in a block of symbols.

Network Security:

* Network Security consists of policies and practices adopted to prevent and monitor unauthorized access, misuse of computer resources.
* Network Security involves the authorization of access to data in a network which is controlled by network administrator. Users choose are assigned an ID and password that allows them access to information and programs within their authority.

Need for Network Security:

* It is needed by an organization to prevent malicious use. The goal of network security is to keep the network running and safe for all users.
* It helps to protect workstations from harmful spyware.

Network Security Services:

1. Message Confidentiality
2. Message Integrity
3. Message Authentication
4. Message Non-Repudiation
5. Entity Authentication

Message Confidentiality:

* Message Confidentiality or privacy means that the sender and receiver expect confidentiality. The transmitted message must make sense to only the intended receiver.

* Eg) When a customer communicates with her bank, he or she expects that the communication is totally confidential.

Message Integrity:

* Message Integrity means that the data must arrive at the receiver exactly as they were sent.

* There must be no changes during the transmission, neither accidentally nor maliciously.

Message Authentication:

* Message Authentication is a service beyond message integrity. In message authentication the receiver needs to be sure of the sender's identity and that an imposter has not sent the message.

Message Non-Repudiation:

* It means that a sender must not be able

to deny sending a message that he or she, in fact the user did. The burden of proof falls on receiver.

Eg) When a customer sends a message to transfer money from one account to another, the bank must have proof that the customer actually requested this transaction.

Entity Authentication:

* Entity Authentication (or) User Identification, the entity or user is verified prior to access to the system resources.

Peer to Peer Networks (P2P):

* P2P networking is a distributed application architecture that partitions tasks or workloads between peers.

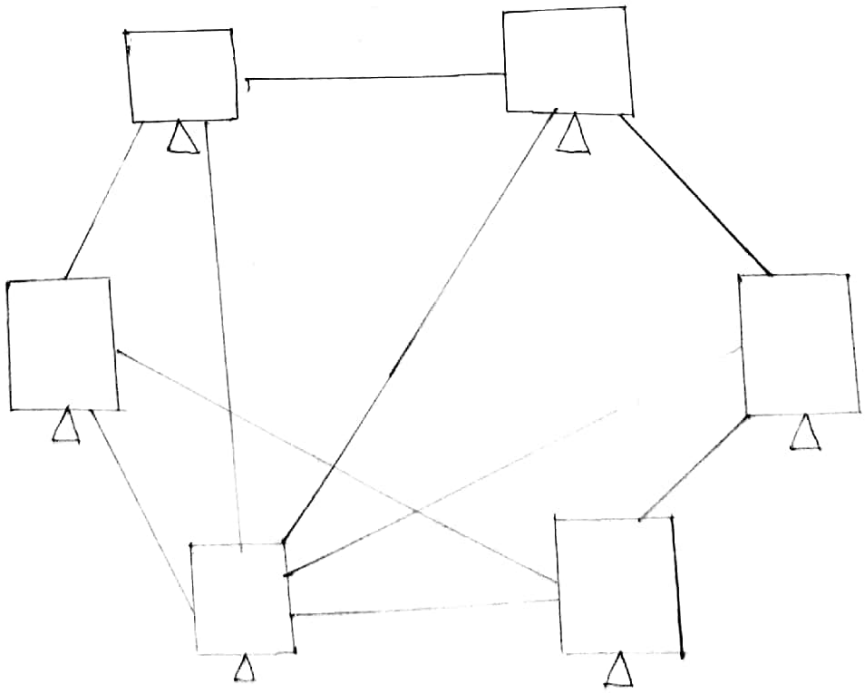


Fig: Peer to Peer Networks

* Peers make a portion of their resources such as processing power, disk storage or network bandwidth directly available to other network participants without the need of coordination by servers.

P2P Architecture:

* Peer to Peer architecture (P2P Architecture) is a commonly used computer networking architecture in which each workstation or node has the same capabilities and responsibilities.

* P2P is used to refer a single software program designed so that each instance of the program may act as both client and server with the same responsibilities.

* Routing and Resource Discovery:

* Data is exchanged directly over the TCP/IP network but at the application layer peers are able to communicate with each other directly.

* Overlays are used for indexing and peer discovery and make the P2P networks independent.

Based on the number of nodes linked to each other within overlay network and how many resources are indexed, it is classified into two types

* Unstructured Networks
* Structured Networks

Unstructured Networks:

* Do not impose a particular structure by design, but rather are formed by nodes that are randomly connected to each other.

* It is easy to build because there is no structure.

* Unstructured networks are highly robust in terms of high rates of churn, that is when large numbers of peers are frequently joining and leaving the network.

* Flooding causes a very high amount of signaling traffic in the network, uses more CPU memory.

* If a peer is looking for rare data share, it is highly unlikely that the search will be successful.

Structured Networks:

* Structured P2P networks, the overlay is organized into a specific topology and the protocol ensures that any node can efficiently search the network for a resource.

* Distributed Hash Table (DHT) is the common type of structured P2P networks implemented.

* In order to route traffic efficiently through the network nodes in a structured overlay must maintain lists of neighbour that satisfy specific criteria. This makes them less robust with a high rate of churn.

Advantages:

* Easy to Install
* Over all cost of building and maintaining the network is very less.
* All the resources and content are shared by all peers unlike server client model where server shares all content and resources.
* The failure of one peer doesn't affect the functioning of other peers.

Disadvantages:

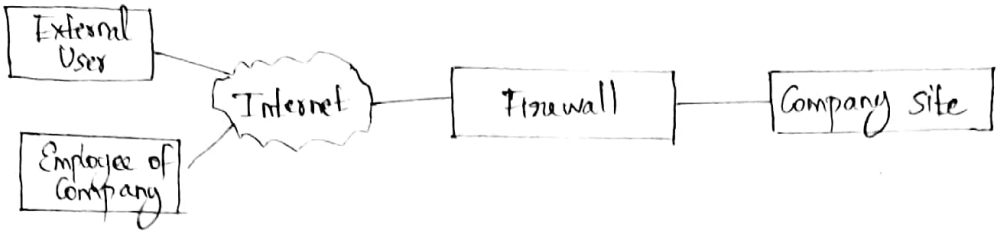
* The whole system is decentralized.
* Less security
* Data recovery or backup is very difficult.

Applications:

* Content Delivery
* File Sharing Networks
* Multimedia

Firewall:

* Firewall is used to prevent intruders from securing internet connection and making unauthorized access and denial of service attacks to the organization network.
* This could be a router, gateway or special purpose computer. The firewalls examine packet flowing into and out of the organization network and restrict access to the network.



Types of Firewall:

There are two types of firewall. They are,

1. Packet filtering firewall.
2. Application level gateway (Proxy Firewall).

1. Packet Filter Firewall:

* A firewall can be used as packet filter.
* It can forward or block packets based on the information in the network layer and transport layer header. Source and destination IP addresses, source and destination port addresses and type of protocol. (TCP (or) UDP)

Interface	Source IP	Source Port	Destination IP	Destination Port
	131.34.0.0	*	*	*
	*	*	*	23
	*	*	194.18.20.8	*
2	*	80	*	*

* A packet filter firewall, is a router that uses a filtering table to decide which packets must be discarded.

According to Table,

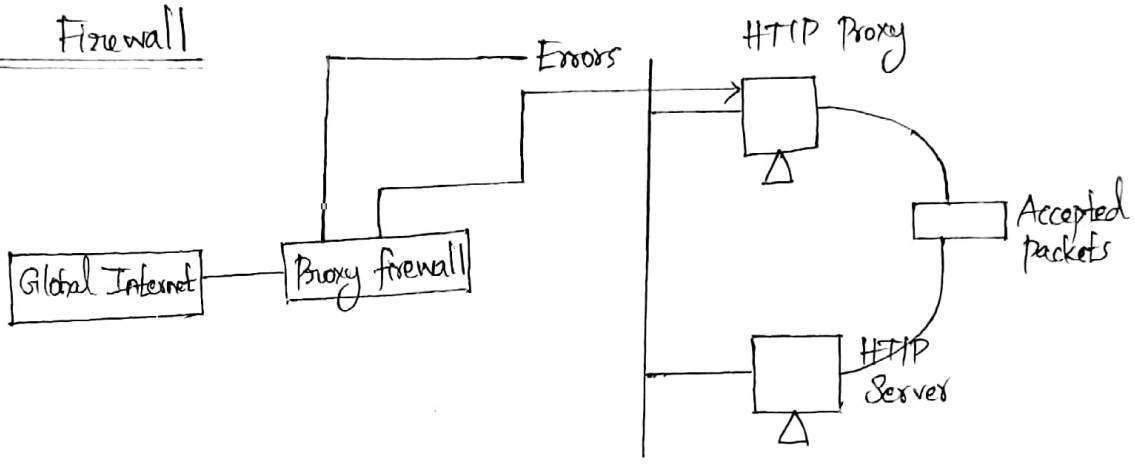
* Incoming packets from network 131.34.0.0 are blocked.

* Incoming packets destined for any internal TELNET server are blocked. (Port 23)

* Incoming packets destined for internal host 194.78.20.8 are blocked. The organization wants this host for internal use only.

* Outgoing packets destined for an HTTP server (Port 80) are blocked. The organization does not want employees to browse the Internet.

2. Proxy Firewall



* The packet filter firewall is based on the information available in the network layer and transport layers headers.

* Sometimes, a filter is needed, in the packet filter firewall.

* In Proxy firewall, proxy computer is installed which stands between the customer (user) and the corporation computer.

* When the user client process sends a message, the proxy firewall runs a server process to receive a request. The server opens the packet and finds out if the request is legitimate.

* If it is, the server acts as a client process and sends the message to the real server. If it is not, the

the message is dropped and an error message is sent to the external user. In this way, the requests of the external users are filtered based on the contents at the application layer.

Advantage of Firewall

* System administrator can manage the firewall to provide security.

Disadvantage of firewall:

* Severe vulnerability.
* Transmission of private information looks like legitimate communication.