UNIT V – THE APPLICATION LAYER & NETWORK SECURITY

Application layer protocols: HTTP, FTP, SMTP, DNS

1.1 HTTP (HYPERTEXT TRANSFER PROTOCOL)

- The HyperText Transfer Protocol (HTTP) is used to define how the client-server programs can be written to retrieve web pages from the Web.
- It is a protocol used to access the data on the World Wide Web (WWW). The HTTP protocol can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on.
- HTTP is a stateless request/response protocol that governs client/server communication. An HTTP client sends a request; an HTTP server returns a response.
- The server uses the port number 80; the client uses a temporary port number.
- HTTP uses the services of TCP, a connection-oriented and reliable protocol. HTTP is a text-oriented protocol.
- It contains embedded URL known as links.
- When hypertext is clicked, browser opens a new connection, retrieves file from the server and displays the file.
- Each HTTP message has the general form

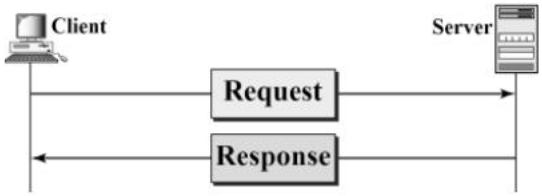
START_LINE <CRLF>
MESSAGE_HEADER <CRLF>
<CRLF>MESSAGE_BODY<CRLF>
where <CRLF> stands for carriage-return-line-feed.

Features of HTTP

- Connectionless protocol: HTTP is a connectionless protocol. HTTP client initiates a request and waits for a response from the server. When the server receives the request, the server processes the request and sends back the response to the HTTP client after which the client disconnects the connection. The connection between client and server exist only during the current request and response time only.
- Media independent: HTTP protocol is a media independent as data can be sent as long asboth the client and server know how to handle the data content. It is required for both the client and server to specify the content type in MIME-type header.
- Stateless: HTTP is a stateless protocol as both the client and server know each other only during the current request. Due to this nature of the protocol, both the client and server do not retain the information between various requests of the web pages.

HTTP REQUEST AND RESPONSE MESSAGES

The HTTP protocol defines the format of the request and response messages.



- Request Message: The request message is sent by the client that consists of a request line, headers, and sometimes a body.

- Response Message: The response message is sent by the server to the client that consists of a status line, headers, and sometimes a body

HTTP REQUEST MESSAGE

Request Line
Request Header : Value
Body (optional)

- The first line in a request message is called a request line.
- After the request line, we can have zero or more request header lines.
- The body is an optional one. It contains the comment to be sent or the file to be published on the website when the method is PUT or POST.

Request Line

- There are three fields in this request line - Method, URL and Version. The Method field defines the request types.
- The URL field defines the address and name of the corresponding web page.
- The Version field gives the version of the protocol; the most current version of HTTP is 1.1.
- Some of the Method types are

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

Request Header

- Each request header line sends additional information from the client to the server.
- Each header line has a header name, a colon, a space, and a header value. The value field defines the values associated with each header name.
- Headers defined for request message include

Header	*Description*
User-agent	Identifies the client program
Accept	Shows the media format the client can accept
Accept-charset	Shows the character set the client can handle
Accept-encoding	Shows the encoding scheme the client can handle
Accept-language	Shows the language the client can accept
Authorization	Shows what permissions the client has
Host	Shows the host and port number of the client
Date	Shows the current date
Upgrade	Specifies the preferred communication protocol
Cookie	Returns the cookie to the server
If-Modified-Since	If the file is modified since a specific date

Body

- The body can be present in a request message. It is optional.
- Usually, it contains the comment to be sent or the file to be published on the website when the method is PUT or POST.

Conditional Request

- A client can add a condition in its request.
- In this case, the server will send the requested web page if the condition is metor inform the client otherwise.
- One of the most common conditions imposed by the client is the time and datethe web page is modified.
- The client can send the header line If-Modified-Since with the request to tell theserver that it needs the page only if it is modified after a certain point in time.

HTTP RESPONSE MESSAGE

Status Line
Response Header : Value
Body

- The first line in a request message is called a status line.
- After the request line, we can have zero or more response header lines.

- The body is an optional one. The body is present unless the response is an error message

Status Line

- The Status line contains three fields - HTTP version, Status code, Status phrase The first field defines the version of HTTP protocol, currently 1.1.
- The status code field defines the status of the request. It classifies the HTTP result.
- It consists of three digits. 1xx–Informational, 2xx– Success, 3xx–Redirection, 4xx–Client error, 5xx–Server error
- The Status phrase field gives brief description about status code in text form.
- Some of the Status codes are

Code	Phrase	Description
100	Continue	Initial request received, client to continue process
200	OK	Request is successful
301	Moved permanently	Requested URL is no longer in use
404	Not found	Document not found
500	Internal server error	An error such as a crash, at the server site

Response Header

- Each header provides additional information to the client.
- Each header line has a header name, a colon, a space, and a header value.
- Some of the response headers are:

Response Header	Description
Content-type	specifies the MIME type
Expires	date and time up to which the document is valid
Last-modified	date and time when the document was last updated
Location	specifies location of the created or moved document

Body

The body contains the document to be sent from the server to the client.
The body is present unless the response is an error message.

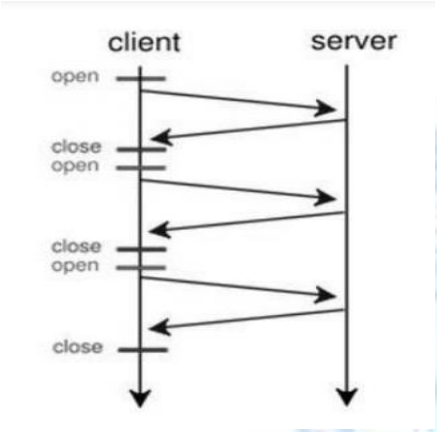
HTTP CONNECTIONS

- HTTP Clients and Servers exchange multiple messages over the same TCP connection.
- If some of the objects are located on the same server, we have two choices: to retrieve each object using a new TCP connection or to make a TCP connection and retrieve them all.
- The first method is referred to as a non-persistent connection, the second as a persistent connection.

- HTTP 1.0 uses non-persistent connections and HTTP 1.1 uses persistent connections

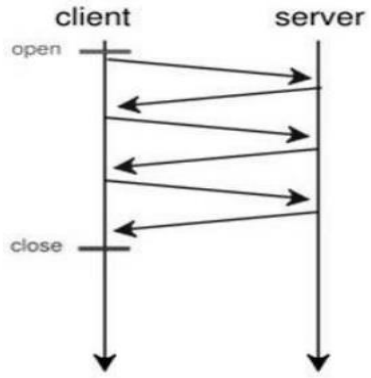
NON-PERSISTENT CONNECTIONS

- In a non-persistent connection, one TCP connection is made for each request/response.
- Only one object can be sent over a single TCP connection
- The client opens a TCP connection and sends a request.
- The server sends the response and closes the connection.
- The client reads the data until it encounters an end-of-file marker.
- It then closes the connection.



PERSISTENT CONNECTIONS

- HTTP version 1.1 specifies a persistent connectionby default. Multiple objects can be sent over a single TCP connection.
- 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.
- Time and resources are saved using persistent connections. Only one set of buffers and variables needs to be set for the connection at each site.
- The round-trip time for connection establishment and connection termination is saved.



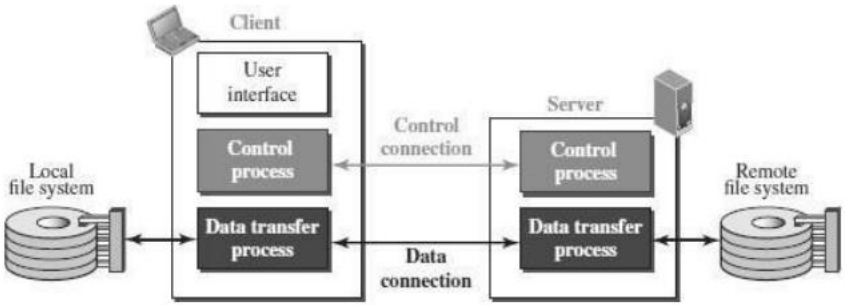
1.2 FTP (FILE TRANSFER PROTOCOL)

- FTP stands for File transfer protocol.
- FTP is a standard internet protocol provided by TCP/IP used for transmitting the files from one host to another.
- It is mainly used for transferring the web page files from their creator to the computer that acts as a server for other computers on the internet.
- It is also used for downloading the files to computer from other servers.
- Although we can transfer files using HTTP, FTP is a better choice to transfer large files or to transfer files using different formats.

FTP OBJECTIVES

- It provides the sharing of files.
- It is used to encourage the use of remote computers.
- It transfers the data more reliably and efficiently.

FTP MECHANISM

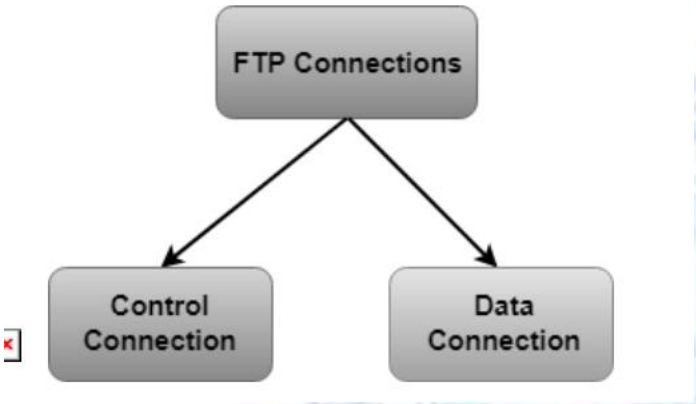


The above figure shows the basic model of the FTP.

- ✧ The FTP client has three components:
  - o user interface, control process, and data transfer process.
- ✧ The server has two components:
  - o server control process and server data transfer process.

FTP CONNECTIONS

- There are two types of connections in FTP -
  Control Connection and Data Connection.
- The two connections in FTP have different lifetimes.
- The control connection remains connected during the entire interactive FTP session.
- The data connection is opened and then closed for each file transfer activity.
- 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. FTP uses two well-known TCP ports:
- o Port 21 is used for the control connection
- o Port 20 is used for the data connection.



Control Connection:

o The control connection uses very simple rules for communication.
o Through control connection, we can transfer a line of command or lineof response at a time.
o The control connection is made between the control processes.
o The control connection remains connected during the entire interactive FTP session.

Data Connection:

o The Data Connection uses very complex rules as data types may vary. o The data connection is made between data transfer processes.
o The data connection opens when a command comes for transferring the files and closes when the file is transferred.

FTP COMMUNICATION

- FTP Communication is achieved through commands and responses.
- FTP Commands are sent from the client to the server FTP responses are sent from the server to the client.
- FTP Commands are in the form of ASCII uppercase, which may or may not be followed by an argument.
- Some of the most common commands are

Command	*Description*
ABOR	Abort the previous command
CDUP	Change to parent directory
CWD	Change to another directory
DELE	Delete a file
LIST	List subdirectories or files
MKD	Create a new directory
PASS	Password
PASV	Server chooses a port
PORT	Client chooses a port
PWD	Display name of current directory
QUIT	Log out of the system
RETR	Retrieve files; files are transferred from server to client
RMD	Delete a directory
RNFR	Identify a file to be renamed
RNTO	Rename the file
STOR	Store files; file(s) are transferred from client to server
STRU	Define data organization (**F**: file, **R**: record, or **P**: page)
TYPE	Default file type (**A**: ASCII, **E**: EBCDIC, **I**: image)
USER	User information
MODE	Define transmission mode (**S**: stream, **B**: block, or **C**: compressed

- ✧ Every FTP command generates at least one response.
- ✧ A response has two parts: a three-digit number followed by text.
- ✧ The numeric part defines the code; the text part defines needed parameter.

Code	*Description*	*Code*	*Description*
125	Data connection open	**250**	Request file action OK
150	File status OK	**331**	User name OK; password is needed
200	Command OK	**425**	Cannot open data connection
220	Service ready	**450**	File action not taken; file not available
221	Service closing	**452**	Action aborted; insufficient storage
225	Data connection open	**500**	Syntax error; unrecognized command
226	Closing data connection	**501**	Syntax error in parameters or arguments
230	User login OK	**530**	User not logged in

FTP FILE TYPE

FTP can transfer one of the following file types across the data connection: ASCII file, EBCDIC file, or image file.

FTP DATA STRUCTURE

- ✧ FTP can transfer a file across the data connection using one of the following data structure : file structure, record structure, or page structure.
- ✧ The file structure format is the default one and has no structure. It is a continuous stream of bytes.

- ✧ In the record structure, the file is divided into records. This can be used only with text files.
- ✧ In the page structure, the file is divided into pages, with each page having a page number and a page header. The pages can be stored and accessed randomly or sequentially.

FTP TRANSMISSION MODE

- ✧ FTP can transfer a file across the data connection using one of the following three transmission modes: stream mode, block mode, or compressed mode.
- ✧ The stream mode is the default mode; data are delivered from FTP to TCP as a continuous stream of bytes.
- ✧ In the block mode, data can be delivered from FTP to TCP in blocks. In the compressed mode, data can be compressed and delivered from FTP to TCP.

FTP FILE TRANSFER

- ✧ 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 three things:

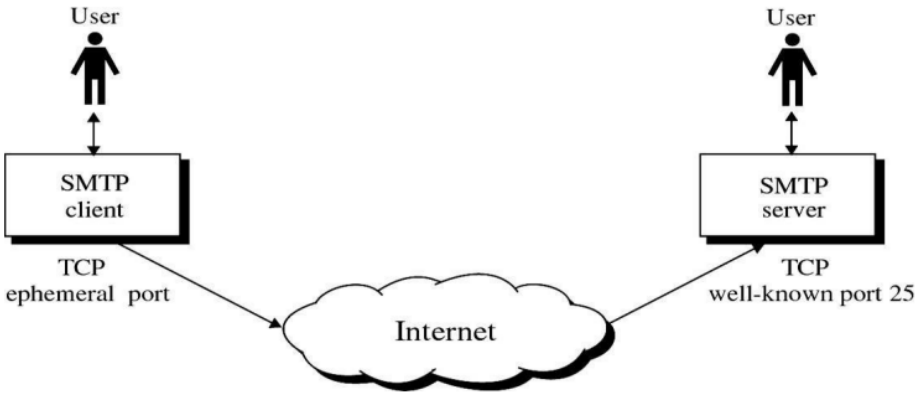
o retrieving a file (server to client)
o storing a file (client to server)
o directory listing (server to client).

FTP SECURITY

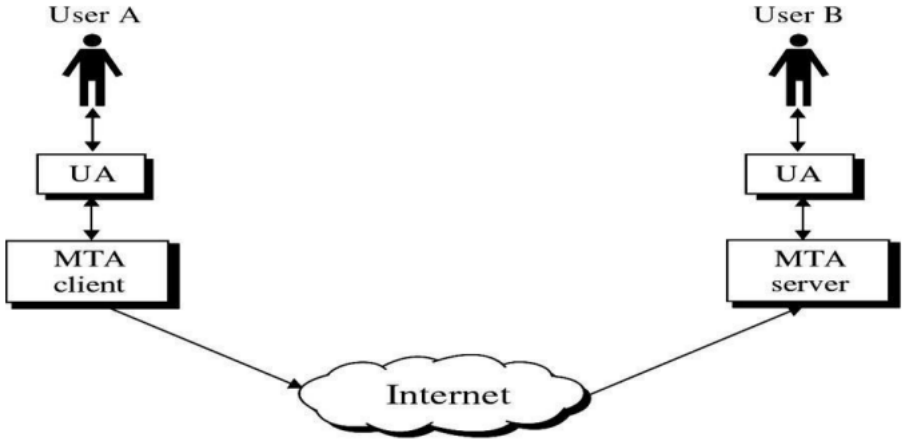
- ✧ FTP requires a password, the password is sent in plaint-ext which is encrypted.
- ✧ This means it can be intercepted and used by an attacker.
- ✧ The data transfer connection also transfers data in plain-text, which is insecure.
- ✧ To be secure, one can add a Secure Socket Layer between the FTP application layer and the TCP layer.
- ✧ In this case FTP is called SSL-FTP.

SMTP-SIMPLE MAIL TRANSFER PROTOCOL

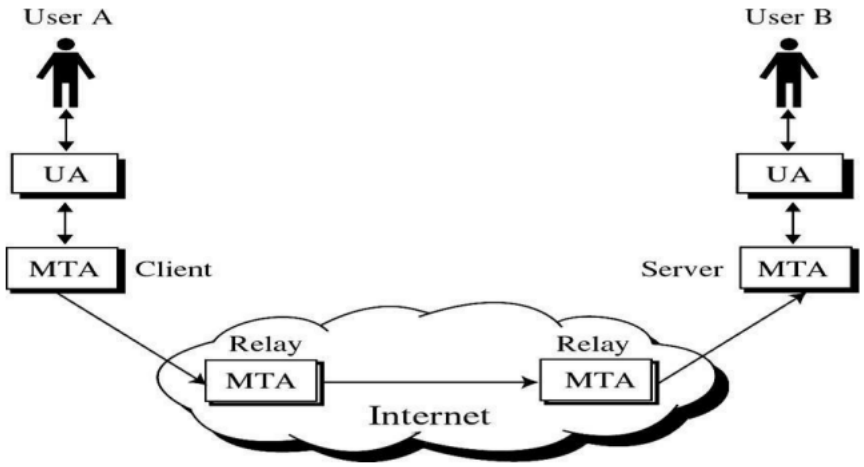
- ✧ SMTP is the standard protocol for transferring mail between hosts in the TCP/IP protocol suite.
- ✧ SMTP is not concerned with the format or content of messages themselves.
- ✧ SMTP uses information written on the envelope of the mail (message header),but does not look at the contents (message body) of the envelope.



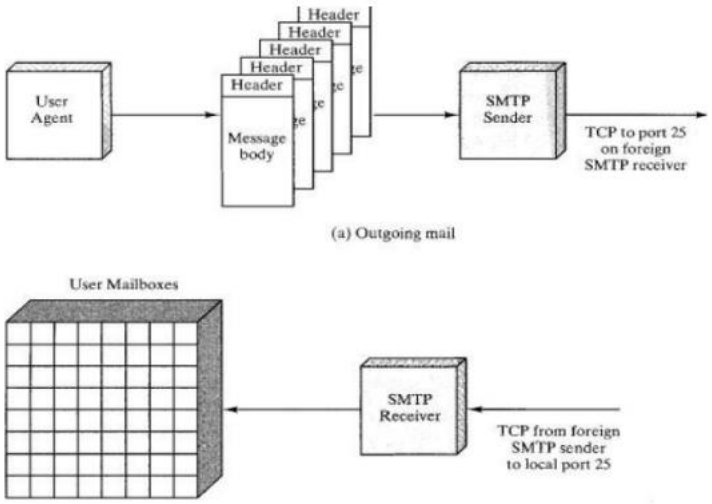
SMTP clients and servers have two main components
o User Agents(UA) – Prepares the message, encloses it in an envelope. o Mail Transfer Agent (MTA) – Transfers the mail across the internet



SMTP also allows the use of Relays allowing other MTAs to relay the mail



SMTP MAIL FLOW



- ✧ To begin, mail is created by a user-agent program in response to user input.
- ✧ Each created message consists of a header that includes the recipient's email address and other information, and a message body containing the message to be sent.
- ✧ These messages are then queued in some fashion and provided as input to an SMTP Sender program.

SMTP COMMANDS AND RESPONSES

- ✧ The operation of SMTP consists of a series of commands and responses exchanged between the SMTP sender and SMTP receiver.
- ✧ The initiative is with the SMTP sender, who establishes the TCP connection. Once the connection is established, the SMTP sender sends commands over the connection to the receiver.
- ✧ The command is from an MTA client to an MTA server; the response is from an MTA server to the MTA client.

SMTP 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.

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

SMTP 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.

SMTP Responses

Code	*Description*
	Positive Completion Reply
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
	Positive Intermediate Reply
354	Start mail input
	Transient Negative Completion Reply
421	Service not available
450	Mailbox not available
451	Command aborted: local error
452	Command aborted; insufficient storage
	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

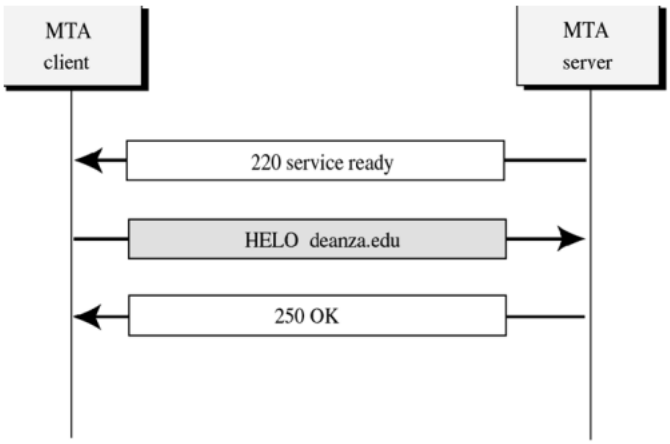
SMTP OPERATIONS

Basic SMTP operation occurs in three phases:
1. Connection Setup
2. Mail Transfer
3. Connection Termination

Connection Setup

- An SMTP sender will attempt to set up a TCP connection with a target host when it has one or more mail messages to deliver to that host.
- The sequence is quite simple:

1. The sender opens a TCP connection with the receiver.
2. Once the connection is established, the receiver identifies itself with "Service Ready".
3. The sender identifies itself with the HELO command.
4. The receiver acceptsthe sender's identification with "OK".
5. If the mail service on the destination is unavailable, the destination host returns a "Service Not Available" reply in step 2, and the process is terminated.

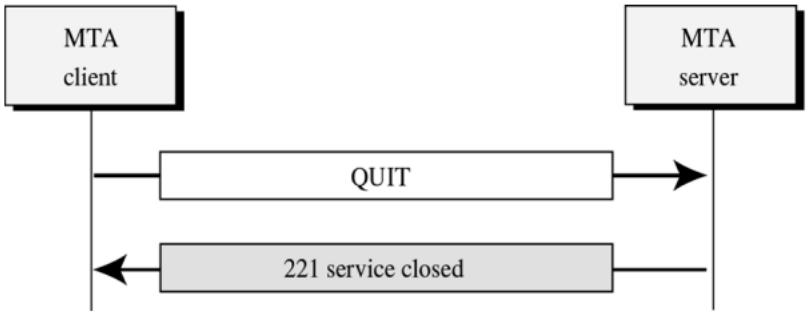


Mail Transfer

- Once a connection has been established, the SMTP sender may send one or more messages to the SMTP receiver.
- There are three logical phases to the transfer of a message:
    1. A MAIL command identifies the originator of the message.
    2. One or more RCPT commands identify the recipients for this message.
    3. A DATA command transfers the message text.

Connection Termination

- The SMTP sender closes the connection in two steps.
- First, the sender sends a QUIT command and waits for a reply.
- The second step is to initiate a TCP close operation for the TCP connection.
- The receiver initiates its TCP close after sending its reply to the QUIT command.



LIMITATIONS OF SMTP

- SMTP cannot transmit executable files or other binary objects.
- SMTP cannot transmit text data that includes national language characters, as these are represented by 8-bit codes with values of 128 decimal or higher, and SMTP is limited to 7-bit ASCII.
- SMTP servers may reject mail message over a certain size.

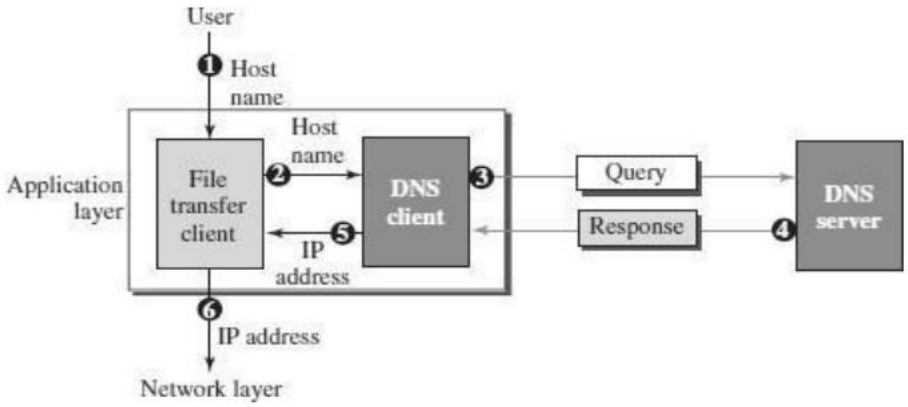
- SMTP gateways that translate between ASCII and the character code EBCDIC do not use a consistent set of mappings, resulting in translation problems.
- Some SMTP implementations do not adhere completely to the SMTP standards defined.
- Common problems include the following:

1. Deletion, addition, or recording of carriage return and linefeed.
2. Truncating or wrapping lines longer than 76 characters
3. Removal of trailing white space (tab and space characters).
4. Padding of lines in a message to the same length. Conversion of tab characters into multiple-space characters

1.4 DOMAIN NAME SYSTEM(DNS)

- Domain Name System was designed in 1984.
- DNS is used for name-to-address mapping.
- The DNS provides the protocol which allows clients and servers to communicate with each other.
- Eg: Host name like www.yahoo.com is translated into numerical IP addresses like 207.174.77.131
- Domain Name System (DNS) is a distributed database used by TCP/IP applications to map between host names and IP addresses and to provide electronic mail routing information.
- Each site maintains its own database of information and runs a server program that other systems across the Internet can query

WORKING OF DNS



The following six steps shows the working of a DNS. It maps the host name to an IP address:

1. The user passes the host name to the file transfer client.
2. The file transfer client passes the host name to the DNS client.
3. 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.
4. The DNS server responds with the IP address of the desired file transfer server.
5. The DNS server passes the IP address to the file transfer client. 6. The file transfer client now uses the received IP address to access the file transfer server.

NAME SPACE

- To be unambiguous, the names assigned to machines must be carefully selected from a name space with complete control over the binding between the names and IP address.
- The names 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 (or) hierarchical.

Flat Name Space

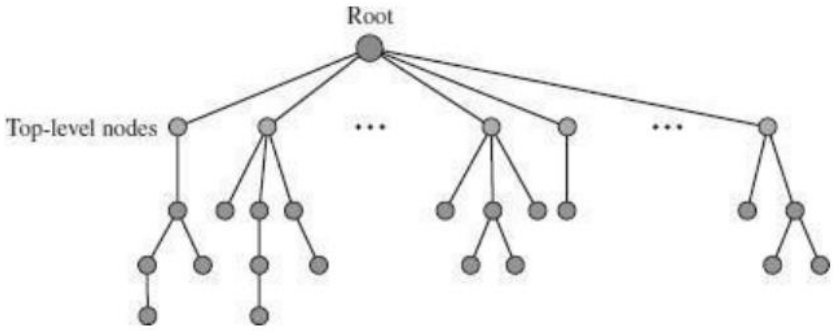
- In a flat name space, a name is assigned to an address.
- A name in this space is a sequence of characters without structure.
- The main disadvantage of a flat name space is that it cannot be used in a large system such as Internet because it must be centrally controlled to avoid ambiguity and duplication.

Hierarchical Name Space

- In a hierarchical name space, each name is made of several parts. The first part can define the organization, the second part can define the name,the third part can define departments, and so on.
- 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.
- The responsibility for the rest of the name can be given to the organization itself.
- Suffixes can be added to the name to define host or resources. The management of the organization need not worry that the prefix chosen fora host is taken by another organization because even if part of an address is the same, the whole address is different.
- The names are unique without the need to be assigned by a central authority.
- The central authority controls only part of the name, not the whole name.

DOMAIN NAME SPACE

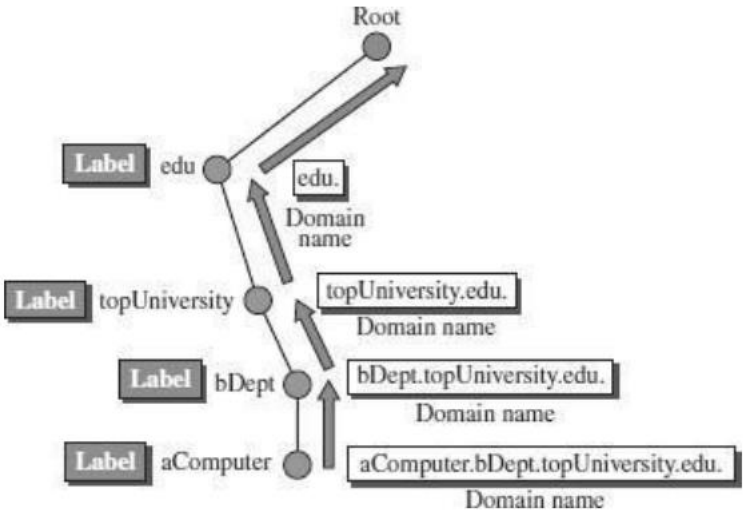
- To have a hierarchical name space, a domain name space was designed. In this design, the names are defined in an inverted-tree structure with the root at the top.
- Each node in the tree has a label, which is a string with a maximum of 63 characters.
- The root label is a null string.
- DNS requires that children of a node have different labels, which guarantees the uniqueness of the domain names.



- 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 (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.

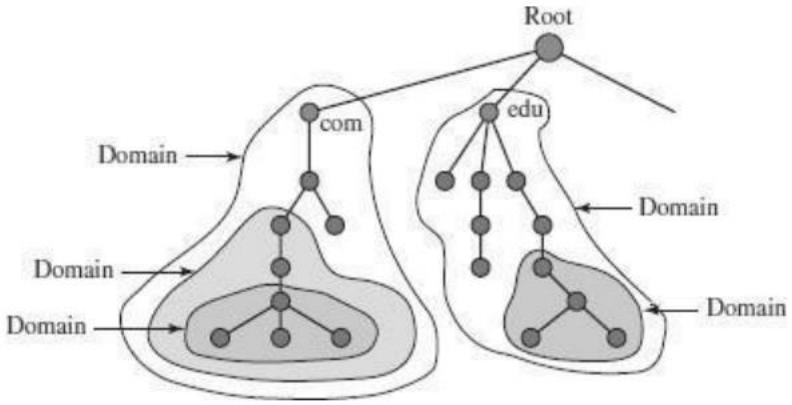
Domain Name

- Each node in the tree has a label called as 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.
- 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.
- If a label is terminated by a null string, it is called a fully qualified domainname (FQDN).
- If a label is not terminated by a null string, it is called a partially qualified domain name (PQDN).



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 sub-tree.
- A domain may itself be divided into domains.

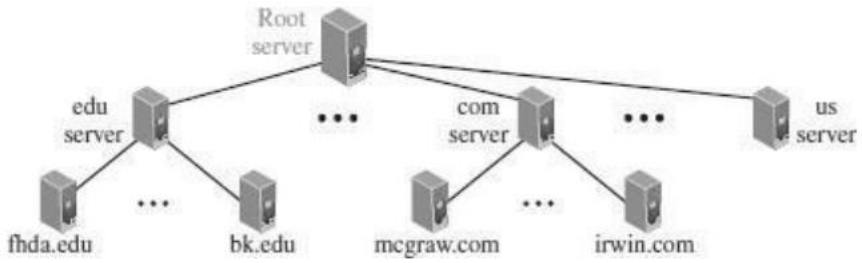


DISTRIBUTION OF NAME SPACE

- The information contained in the domain name space must be stored.
- But it is very inefficient and also 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.
- It is not reliable because any failure makes the data inaccessible.
- The solution to these problems is to distribute the information among many computers called DNS servers.

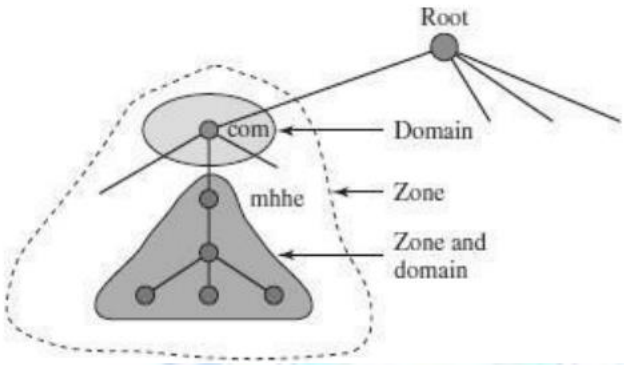
HIERARCHY OF NAME SERVERS

- The way to distribute information among DNS servers is to divide the whole space into many domains based on the first level.
- Let the root stand-alone and create as many domains as there are first level nodes.
- Because a domain created this way could be very large, DNS allows domains to be divided further into smaller domains.
- Thus, we have a hierarchy of servers in the same way that we have a hierarchy of names



ZONE

- What a server is responsible for, or has authority over, is called a zone.
- The server makes a database called a zone file and keeps all the information for every node under that domain.
- If a server accepts responsibility for a domain and does not divide the domains into smaller domains, the domain and zone refer to the same thing.
- But if a server divides its domain into sub domains and delegates parts of its authority to other servers, domain and zone refer to different things.
- The information about the nodes in the sub domains is stored in the servers at the lower levels, with the original server keeping some sort of references to these lower-level servers.
- But still, the original server does not free itself from responsibility totally.
- It still has a zone, but the detailed information is kept by the lower level servers.



ROOT SERVER

- A root sever 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.
- Currently there are more than 13 root servers, each covering the whole domain name space.
- The servers are distributed all around the world.

PRIMARY AND SECONDARY SERVERS

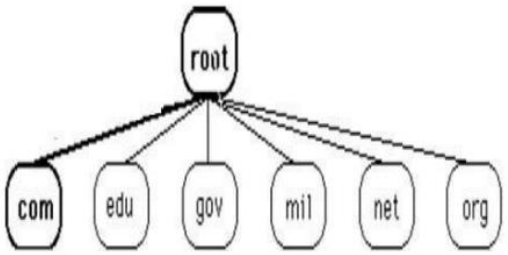
- DNS defines two types of servers: primary and secondary.
- A Primary Server is a server that stores a file about the zone for which it is an authority.
  - Primary Servers are responsible for creating, maintaining, and updating the zone file. □
  - Primary Server stores the zone file on a local disc.
- 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 disc.
- If updating is required, it must be done by the primary server, which sends the updated version to the secondary.
- A primary server loads all information from the disk file; the secondary server loads all information from the primary server.

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 - Generic domains, Country domains, and 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.
- The first level in the generic domains section allows seven possible three character levels. These levels describe the organization types as listed in following table



.COM	Commercial Organizations
.EDU	Educational institutions
.GOV	Government institutions
.MIL	Military groups
.NET	Major network support centers
.INT	International Organizations
.ORG	Nonprofit Organizations

Country Domains

- The country domains section follows the same format as the generic domains but uses two characters for country abbreviations
- E.g.; in for India, us for United States etc) in place of the three character organizational abbreviation at the first level.
- Second level labels can be organizational, or they can be more specific,national designation.
- India for example, uses state abbreviations as a subdivision of the country domain us. (e.g., ca.in.)

Inverse Domains

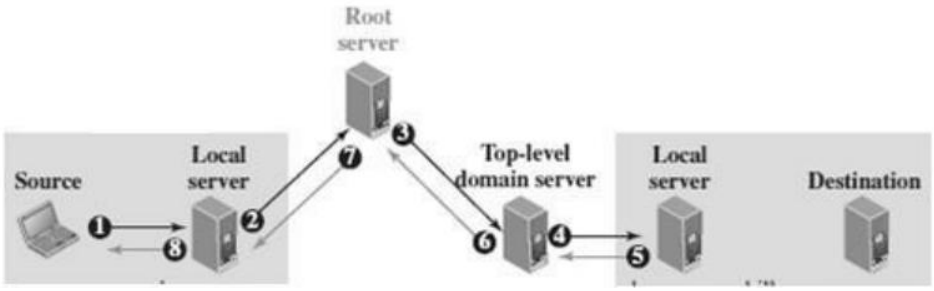
- Mapping an address to a name is called Inverse domain.
- The client can send an IP address to a server to be mapped to a domain name and it is called PTR(Pointer) query.
- To answer queries of this kind, DNS uses the inverse domain

DNS RESOLUTION

- Mapping a name to an address or an address to a name is called name address resolution.

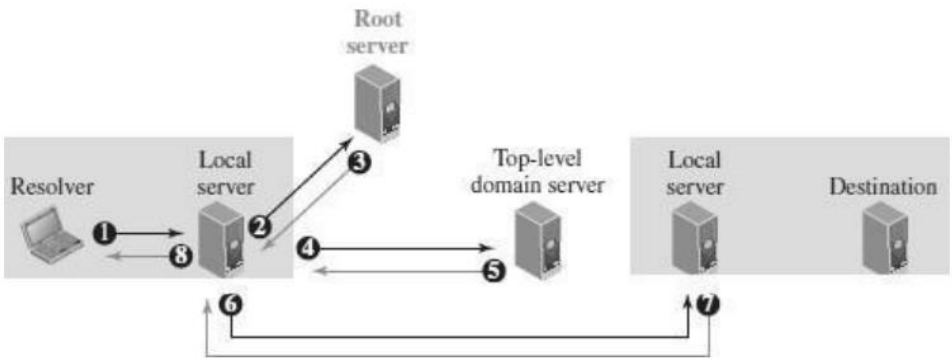
- DNS is designed as a client server application. A host that needs to map an address to a name or a name to an address calls a DNS client named a Resolver.
- The Resolver accesses the closest DNS server with a mapping request.
- If the server has the information, it satisfies the resolver; otherwise, it either refers the resolver to other servers or asks other servers to provide the information.
- After the resolver receives the mapping, it interprets the response to see if it is areal resolution or an error and finally delivers the result to the process that requested it.
- A resolution can be either recursive or iterative.

Recursive Resolution



- The application program on the source host calls the DNS resolver (client) to find the IP address of the destination host. The resolver, which does not know this address, sends the query to the local DNS server of the source (Event 1)
- The local server sends the query to a root DNS server (Event 2)
- The Root server sends the query to the top-level DNS server (Event 3)
- The top-level DNS server knows only the IP address of the local DNS server at the destination. So, it forwards the query to the local server, which knows the IP address of the destination host (Event 4)
- The IP address of the destination host is now sent back to the top-level DNS server (Event 5) then back to the root server (Event 6), then back to the source DNS server, which may cache it for the future queries (Event 7), and finally back to the source host (Event 8).

Iterative Resolution



- In iterative resolution, each server that does not know the mapping, sends theIP address of the next server back to the one that requested it.
- The iterative resolution takes place between two localservers. The original resolver getsthe final answer from the destination localserver.
- The messages shown by Events 2, 4, and 6 contain the same query.
- However, the message shown by Event 3 contains the IP address of the top-level domain server.
- The message shown by Event 5 contains the IP address of the destination local DNS server
- The message shown by Event 7 contains the IP address of the destination.
- When the Source local DNS server receives the IP address of the destination, it sends it to the resolver (Event 8).

DNS CACHING

- Each time a server receives a query for a name that is not in its domain, it needsto search its database for a server IP address.
- DNS handles this with a mechanism called caching. When a server asks for a mapping from another server and receives the response, it stores this information in its cache memory before sending it to the client.
- If the same or another client asks for the same mapping, it can check its cache memory and resolve the problem. However, 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.
- Caching speeds up resolution. Reduction of this search time would increase efficiency, but it can also be problematic.
- 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 RESOURCE RECORDS (RR) **

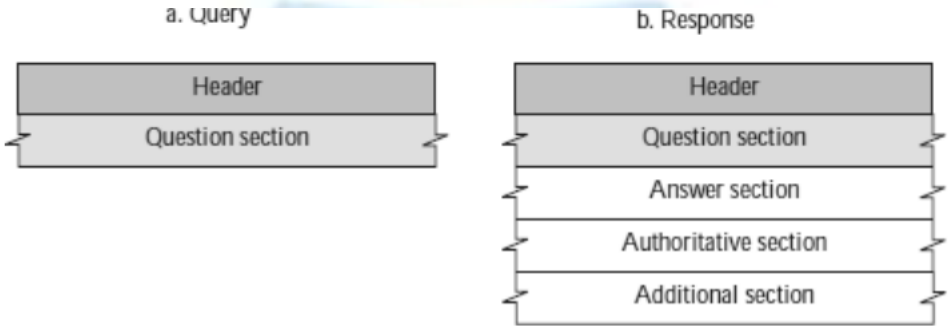
- The zone information associated with a server is implemented as a set of resource records.
- In other words, a name server stores a database of resource records.
- A resource record is a 5-tuple structure
  (Domain Name, Type, Class, TTL, Value)
- The domain name identifies the resource record.
- The type defines how the value should be interpreted.
- The value defines the information kept about the domain name.
- The TTL defines the number of seconds for which the information is valid.
- The class defines the type of network

Types of Resource Records

Type	*Interpretation of value*
A	A 32-bit IPv4 address
NS	Identifies the authoritative servers for a zone
CNAME	Defines an alias for the official name of a host
SOA	Marks the beginning of a zone
MX	Redirects mail to a mail server
AAAA	An IPv6 address

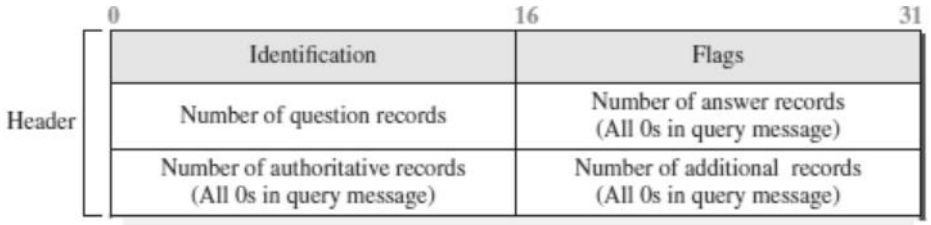
DNS MESSAGES

- DNS has two types of messages: query and response.
- Both types have the same format.
- The query message consists of a header and question section.
- The response message consists of a header, question section, answer section, authoritative section, and additional section



Header

- Both query and response messages have the same header format with some fields set to zero for the query messages.
- The header fields are as follows:



- 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 typein the message.
- Question Section
  - The question section consists of one or more question records. It is present in both query and response messages.
- Answer Section
  - The answer section consists of one or more resource records. It ispresent only in response messages.
- Authoritative Section
  - The authoritative section gives information (domain name) about one ormore authoritative servers for the query.
- Additional Information Section
  - The additional information section provides additional information thatmay help the resolver.

DNS CONNECTIONS

- DNS can use either UDP or TCP.
- In both cases the well-known 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 isused.

DNS REGISTRARS

- New domains are added to DNS through a registrar. A fee is charged.
- A registrar first verifies that the requested domain name is unique and thenenters it into the DNS database.
  - Today, there are many registrars; their names and addresses can be found at http://www.intenic.net
- To register, the organization needs to give the name of its server and the IP address of the server.
- For example, a new commercial organization named wonderful with a server named ws and IP address 200.200.200.5, needs to give the following information to one of the registrars:
  - Domain name: ws.wonderful.com IP address: 200.200.200.5