Introduction, Principles of Application layer protocols, The Web and HTTP, File Transfer, Domain Name Service [DNS]: Working of DNS, DNS records, DNS messages.

The application layer is a layer in the Open Systems Interconnection (OSI) seven-layer model and in the TCP/IP protocol suite. It consists of protocols that focus on process-to-process communication across an IP network and provides a firm communication interface and end-user services.

The application layer is the seventh layer of the OSI model and the only one that directly interacts with the end user.

The application layer provides many services, including:

- Simple Mail Transfer Protocol => SMTP
- File transfer => FTP
- Web surfing => HTTP
- Web chat => Telnet
- Email clients => POP, IMAP
- Network data sharing
- Virtual terminals
- Various file and data operations
- Network Management => SNMP, NFS, TFTP
- Name Management => DNS

The application layer provides full end-user access to a variety of shared network services for efficient OSI model data flow. This layer has many responsibilities, including error handling and recovery, data flow over a network and full network flow. It is also used to develop network-based applications.

In particular, an application-layer protocol defines:

- the types of messages exchanged, for example, request messages and response messages
- the syntax of the various message types, such as the fields in the message and how the fields are delineated
- the semantics of the fields, that is, the meaning of the information in the fields
- rules for determining when and how a process sends messages and responds to messages

The Web and HTTP

Hypertext transfer protocol (HTTP) works with the world wide web (WWW) which is the fastest growing and most used part of the internet. It is popular because of the ease with which it allows access to information. A web browser is a client-server application, which means that it requires both a client and a server component in order to function. A web browser presents data in multimedia formats on the web pages that use text, graphics, sound and video. The web pages are created with a format language called hypertext Markup language (HTML). HTML directs a web browser on a particular web page to produce the appearance of the page in a specific manner. In addition HTML specifies locations for the placement of text, files and objects that are to be transferred from the web server to the web browser.

Hyper links make the World Wide Web easy to navigate. A hyper link is a object, world phrase or picture on a webpage. When that hyperlink is clicked it directs the browser to a new webpage. The webpage contains an address location known as a uniform resource locator (URL).

In the URL, http://www.ekantipur.com/np/pictures, here the http:// tells the browser which protocol we use. The second part “www” is the host name or a name of a specific machine with a specific IP address. The last part “/pictures/” identifies the specific folder, location on the server that contains the default web page.

A web browser usually opens to a starting or home page. The URL of the homepage, has already been stored in a configuration are of the web browser and can be changed at any time. From the starting page, click on one of the webpage hyperlinks or type a URL in the address bar of the browser. The web browser examines the protocol to determine, if it needs to open the other program and then determines the IP address of the web browser using DNS. Then transport layer, network layer, data link layer and physical layer work together to initiate a session with the server contains the folder name of the webpage location. The data can also contain a specific file name for HTML page. If no name is given then the default name as specified in the configuration on the server is used.

The server response to the request by sending to the web client all of the text, audio, video and graphic files specified in the HTML instructions. The client browser reassembles all the files to create a view of the webpage and then terminates the session. If another page that is located on the same or different server is clicked the whole process begins again.

HTTP Message format

i) HTTP request message format

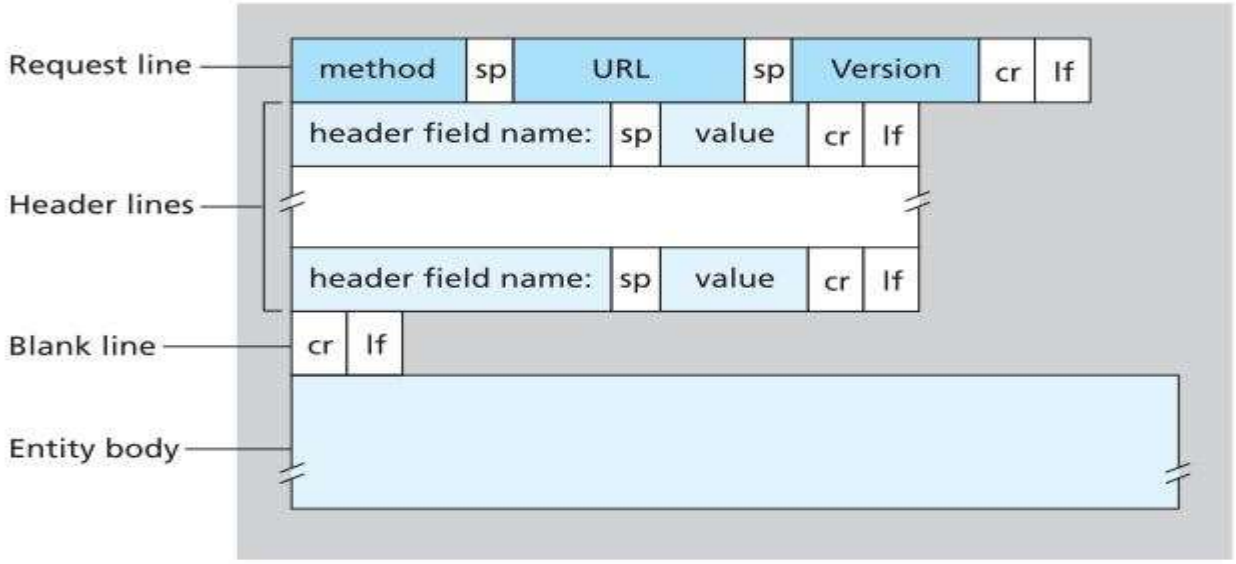


Fig: General format of an HTTP request message

Below we provide a typical HTTP request message:

GET: /somedir/page.html HTTP/1.1
HOST: www.espnsoccernet.com
Connection: close
User-agent: Mozilla/4.0
Accept-language: fr

→ Message consists of five lines (may be more), each followed by a carriage return (cr) and line feed (lf)
→ First line is called request line; the subsequent lines are called the header lines.
→ The request line has 3 fields.

i) The method field

- GET –to browse a particular website
- POST –to search with keywords (entity body is not empty)
- HEAD –requests a HTTP message but leaves out the requested object. Application developers
- PUT –used with web publishing tools to upload objects.
- DELETE –to delete an object on a web server.

ii) URL field

- -/somedir/page.html.

iii) the HTTP version field

- -HTTP/1.1 is the version 1.1 of HTTP.

→ Let's look at the header lines

- Host: www.espnsoccernet.com specifies the host on which the object resides.
- Connection: close is telling the server to close the connection after sending the requesting object.
- User-agent: specifies the browser type.
- Accept-language: fr indicates that the user prefers to receive. French version of the object, if exists on the server, otherwise the server should sent its default version.

ii) HTTP Response Message Format

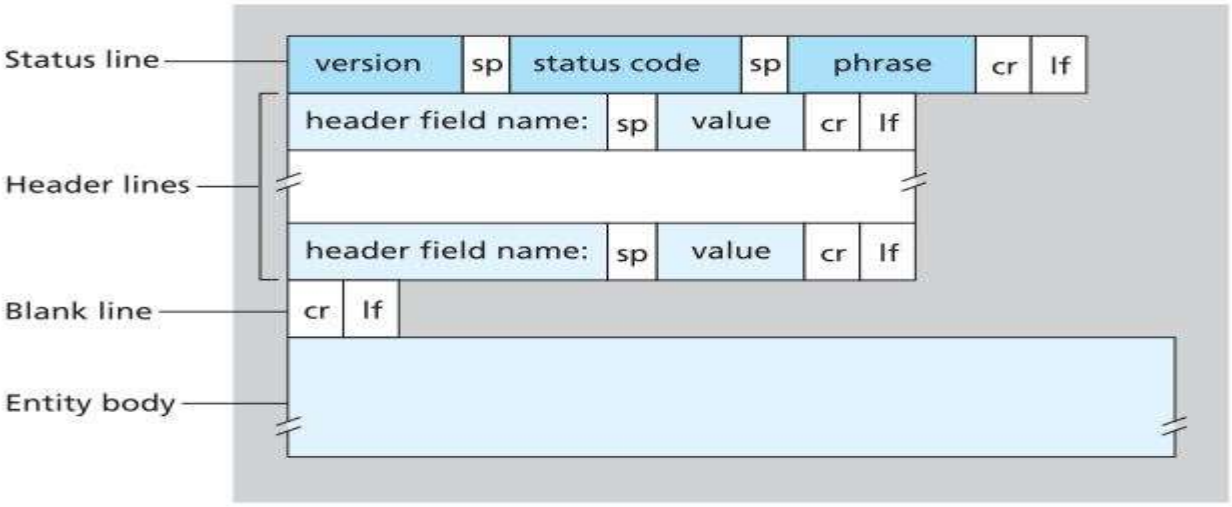


Fig: General format of HTTP response message.
Consists of two parts:

i) Status line
ii) Header Lines

The status line has 3 sections

- The protocol version field.
- The status code.
- Corresponding status message.

A few details about status code and their phrases
200 OK – request succeeded and the information is returned in response.
301 Moved permanently – requested object has been permanently moved, the new URL specified in location: header of the response message.
400 Bad Request – This is a generic code indicating that the request could not be understood by the server.
404 Not Found – The requested document doesn't exist on the server.
505 HTTP Version Not Supported – The requested HTTP protocol version is not supported by the server.

Example:
HTTP/1.1 200 OK
Connection: close
Date: Sun, 09 Mar 2014 09:45 GMT
Server: Apache/1.3.0 (UNIX)
Last–modified: Fri, 6 Jan 2014 08:00:15 GMT
Content-length: 6978
Content-Type: text/html

Connection: Close
— To tell the client that it is going to close the TCP connection after sending the message.
Date:
— Indicates the time and date when the HTTP response was created and sent by the server.
Server:
— Indicates that the message was generated by an Apache web server.
Last-modified:
— Indicates the time and date when the object was created or last modified.
Content-Length:
— Indicates the number of bytes in the object being sent.
Content-Type:

— Indicates that the object in the entity body is HTML text.

Cookies

→ As HTTP server is stateless, web servers can handle thousands of simultaneous TCP connections.
→ It is often desirable for a website to identify users, either because the server wishes to restrict user access or because it wants to serve content as a function of the user identify.
→ For the purpose, HTTP user cookies, which allow sites to keep track of users.
→ Cookie technology has four components.
- A cookie header line in the HTTP response message.
- A cookie header line in the HTTP request message.
- A cookie file kept on the user's end system and managed by the user's browser.
- A back-end database at the website.

→ In the HTTP response message,
→ Set cookie : 1783
→ In the HTTP request message.
Cookie: 1783

FTP (File Transfer Protocol)

FTP is reliable, connection-oriented service that uses TCP to transfer files between systems that support FTP. The main purpose of FTP is to transfer files form one computer to another by copying and moving files from servers to clients, and from clients to servers. When files are copied from a server, FTP first establishes a Control Connection between the client and the server. Then, a second connection is established, which is a link between the computers through which data is transferred. Data transfer can occur in ASCII mode or in binary mode. These modes determine the encoding used for data file, which in the OSI model is a presentation layer task. After the file transfer has ended, the data connection terminates automatically when the entire session of copying and moving files is complete, the command link is closed when the user logs off and ends the session.

Fig: Control and data connections

Client

- Initiates a control TCP connection.
- Sends the user identification and password over the control connection.
- Also sends over the control connection, commands to change the remote directory.

Server

- When a request receives, FTP server starts TCP data connection.
- Sends exactly one file over the data connection and then closes the data connection.
- FTP data connection opens again for another data transfer.
- Non-persistent connection.

FTP Commands and Replies

→ Each successive command follows CR and LF.
→ Each command consists of fur uppercase ASCII characters, some with optional arguments.
→ Some of the more common commands are given below :
- USER username: used to send the user identification to the server.

- PASS password: used to send the user password to the server.
- LIST: used to ask the server to send back a list of all the files in the current remote directory. The list of files is sent over a (new and non-persistent) data connection rather than the control TCP connection.
- RETR filename: used to retrieve (i.e., get) a file from the current directory of the remote host.
- STOR filename: used to store (i.e., put) a file into the current directory of the remote host.

→ Each command is followed by a reply, sent from server to client.
→ The replies are 3-digit numbers, with an optional message following the number.
→ Similar in structure to the status code and phrase in the status line of the HTTP response message.
→ Some typical relies, along with their possible messages, are as follow :

- 331 username ok, password required
- 125 data connection already open, transfer starting
- 425 can't open data connection
- 452 error writing file.

DNS (Domain Name System)

→ DNS is
  - A distributed database implemented in a hierarchy of DNS servers.
  - An application layer protocol that allows hosts to query the distributed database.

→ DNS protocol runs over UDP and users port 53.
→ DNS is commonly employed by other application-layer protocols including HTTP, SMTP and FTP to translate user supplied hosts names to IP addresses.

Mail Server Aliasing

- Permits Company's mail server and web server to have identical (aliased) hostnames.
- E.g. wlink.com: for mail server and for web server also.
- Load distribution

→ Busy sites are replicated over multiple servers. With each server running on a different end system and each having a different IP address.
→ For replicated web servers, a set of IP address is thus associated with one canonical hostname.

Working Of DNS:

→ In order of the user's host to be able to send on HTTP request message to the web server www.facebook.com , the user's host must first obtain the IP address of www.facebook.com . This is done as follows :

i) The same user machine runs the client side of the DNS application.
ii) The browser extracts the host name www.facebook.com from the URL and passes the host name to the client site of the DNS application.
iii) The DNS client sends a query containing the host name to a DNS server.
iv) The DNS client eventually receives a reply which includes the IP address for the host name.
v) Once the browser receives the IP address from DNS, it can initiate a TCP connection to the HTTP server located at port 89 at that IP address. A simple design for DNS world has one DNS server that contains all the mappings. In the centralized design. Clients simply direct all queries to the single DNS server and the DNS server responds directly to the querying clients. The problem with a centralized design include :

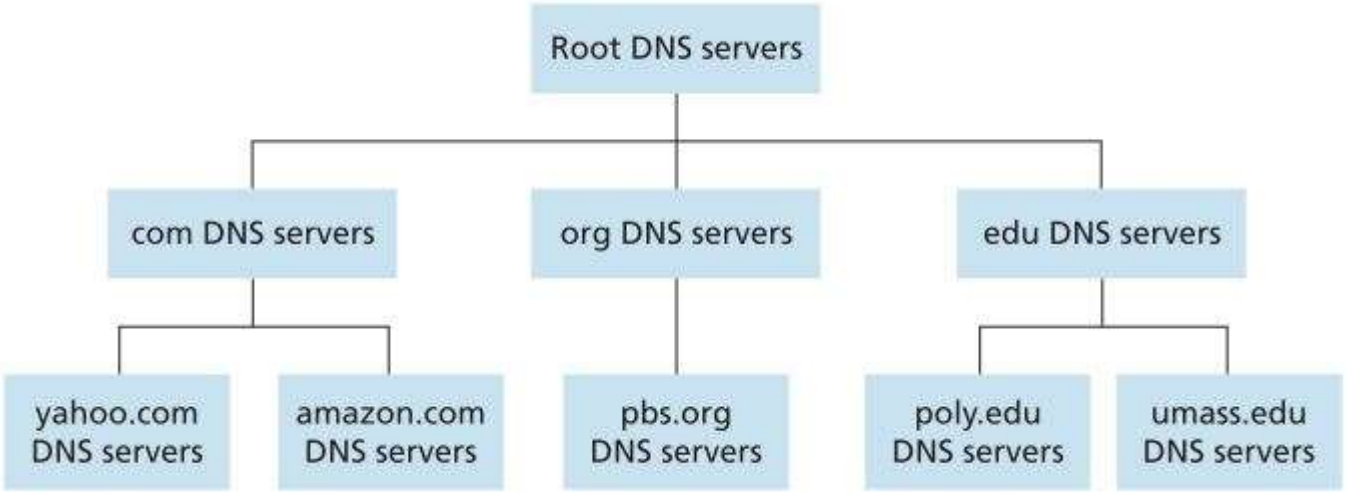
- A single point of failure
  If the DNS server crashes, so does the entire internet.
- Traffic volume
  A single DNS server would have to handle all DNS queries.
- Distant centralized database:
  A single DNS server cannot be close to all the querying clients. If we put the single DNS server in US, then all queries from Australia must travel to the other side of the globe, perhaps over slow and congested links. This kind leads to significant delays.
- Maintenance

The single DNS server would have to keep records for all internet hosts. Not only would this centralized database be huge, but it would have to be updated frequently to account for every new host.

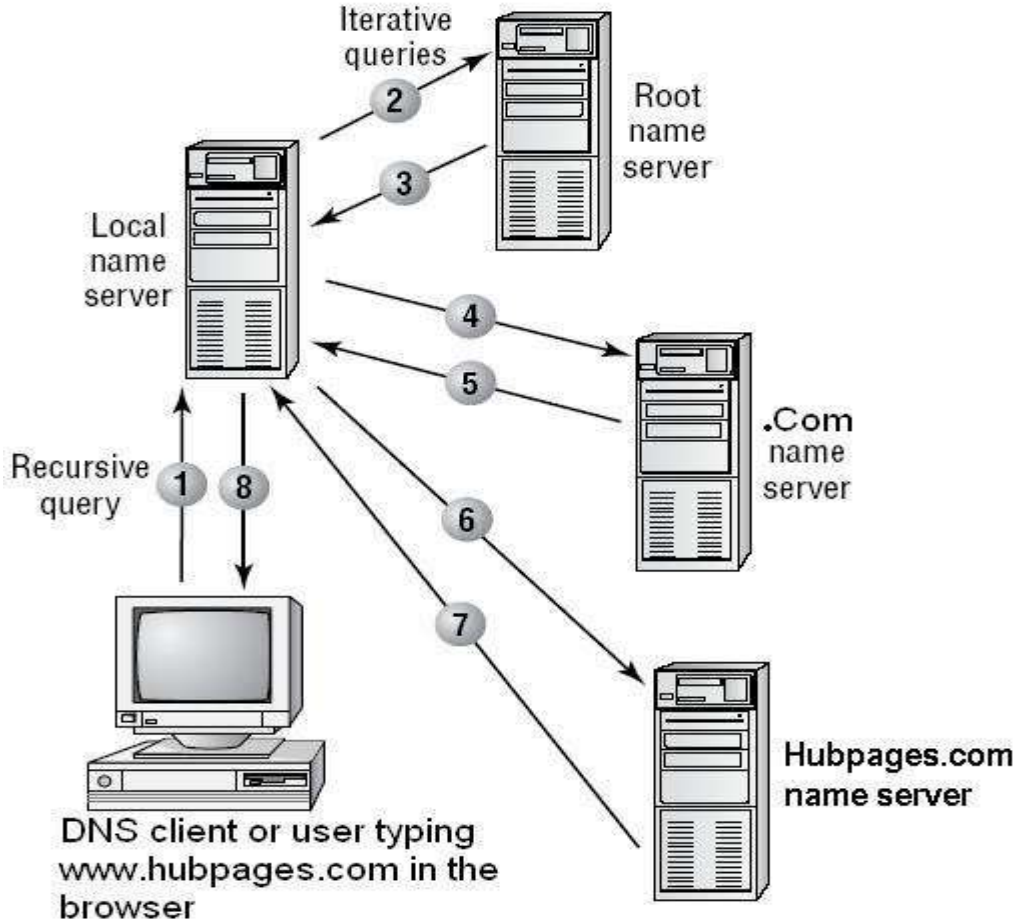
A distributed, Hierarchical database

3 classes of DNS servers:

i) Root DNS server
ii) Top-level domain (TLD) DNS servers.
iii) Authoritative DNS servers.



There is another important type of DNS server called the local DNS server.



1) The resolver sends a recursive DNS query to its local DNS server asking for the IP address of hubpages.com. The local name server is responsible for resolving the name and cannot refer the resolver to another name server.
2) The local name server checks its zones and finds no zones corresponding to the requested domain name.
3) The root name server has authority for the root domain and will reply with the IP address of a name server for the Com top-level domain.
4) The local name server sends an iterative query for hubpages.com to the Com name server.
5) The Com name server replies with the IP address of the name server servicing the hubpages.com domain.
6) The local name server sends an iterative query for hubpages.com to the hubpages.com name server.
7) The hubpages.com name server replies with the IP address corresponding to hubpages.com
8) The local name server sends the IP address of hubpages.com back to the original resolver.

DNS Records

The DNS servers that together implement the DNS distributed database store resource records (RRs), including RRs that provide hostname to IP address mappings. Each DNS replay message carries one or more resource records.
A resource record is a four tuple that contains the following fields.
(Name, Value, Type, TTL)
TTL is the time to live of the resource record; it determines when a resource should be removed from a cache. In example, we ignore the TTL field. The meaning of Name and Value depend on Type.
There are 5 types of DNS records: **A, CNAME, NS, MX and PTR**

i) Type =A
Address (A) records direct a hostname to a numerical IP address. For e.g., if you want www.urdomain.com to point to IP (which is for e.g. 192.168.0.1) you would enter a record that looks like (www.urdomain.com, 192.168.0.1, A)

ii) Type= CNAME
Canonical name (CNAME) allows a machine to be known by one or more host names. There must be always an A record first, and this is known as canonical name or official name. For e.g.: (www.urdomain.com,192.168.0.1,A)
Using CNAME, you can point other hostnames to the canonical (A record) address. For example:
(fitp.urdomain.com, urdomain.com, CNAME)
(mail.urdomain.com, urdomain.com, CNAME)
(ssh.urdomain.com, urdomain.com, CNAME)

iii) Type=NS
Name server (NS) records specify the authoritative name servers for the domain.
For e.g.: urdomain.com, dns.urdomain.com, NS)
Authoritative server

iv) Type = MX
Mail exchangers (MX) records serve the purpose of using mail server through its web server i.e., canonical name.
For e.g.: (urdomain.com, mail.urdomain.com, MX)
Mail server

v) Type=PTR
Pointer (PTR) records are used for reverse lookups. For e.g.: to make 192.168.0.1 resolve the www.urdomain.com the record would look like (1.0.168.192.in addr.arpa, www.urdomain.com, PTR)

DNS Message (Query and Reply)

Identification	Flags
Number of questions	Number of answer RRs
Number of Authority RRs	Number of Addition RRs
Questions Variable number of questions	

Answers Variable number of Resource Records (RRs)
Authority
Additional Information

Fig: DNS message Format

- The first 12 bytes is the header section which has a number of fields. The first field is a 16-bit number that identifies the query. This identifier is copied into the reply message to a query allowing the client to match received replies with sent queries. There are a number of flags in the flag field. A 1 bit query/reply flag indicates whether the message is a query (0) or a reply (1). A 1-bit authoritative flag is set in a reply message when a DNS server is an authoritative server for a queried name. A 1-bit recursion-desired flag is set when a client (host or DNS server) desires that the DNS server perform recursion when it doesn't have the record. A 1-bit recursion available field is set in a reply if the DNS server supports recursion. In the header, there are also four numbers of fields. These fields indicate the number of occurrences of the four types of data sections that follow the header.
- The question section contains information about the query that is being made. This section includes
  a) A name field that contains the name that is being queried.
  b) A type field that indicates the type of question being asked about the names for e.g. a host address associated with a name (Type A) or the mail server for the name (Type MX)
- In a reply form a DNS server the answer section contains the resource records for the name that was originally queried. Recall that in each resource record there is the Type (for e.g.: A, NS, CNAME, or MX), the value and the TTL. A reply can return multiple RRs in the answer.
- Since a hostname can have multiple IP addresses (for e.g. for replicated web servers, as discussed earlier in the section). The authority section contains records of other authoritative servers.

The additional section contains other helpful records. For e.g.: the answer field in a reply to an MX query contains a resource record providing the canonical hostname of a mail server. The additional section contains a Type A record providing the IP address for the canonical hostname of the mail server.