1.2 HTTP Request and HTTP Response

HTTP Request

HTTP request comprises of lines which contains: Request line, Header Fields and Message body. The first line i.e. the Request line specifies the request method i.e. Get or Post. The second line specifies the header which indicates the domain name of the server from where index.htm is retrieved.

HTTP Response

Like HTTP request, HTTP response also has certain structure. HTTP response contains: Status line, Headers and Message body.

Domain Name System (DNS)

➢ When DNS was not into existence, one had to download a host file containing host names and their corresponding IP address.

➢ But with increase in number of hosts of internet, the size of host file also increased. This resulted in increased traffic on downloading this file. To solve this problem the DNS system was introduced

> *Domain Name System helps to resolve the host name to an address. It uses a hierarchical naming scheme and distributed database of IP addresses and associated names.*

Domain Name System Architecture

The Domain name system comprises of Domain Names, Domain Name Space, Name Server that have been described below:

➢ **Domain Names**

Domain Name is a symbolic string associated with an IP address. There are several domain names available. Some of them are generic such as com, edu, gov, net etc, while some country level domain names such as au, in, za, us etc.

➢ **Domain Name Space**
The domain name space refers a hierarchy in the internet naming structure. This hierarchy has multiple levels (from 0 to 127), with a root at the top. Each domain can be partitioned into sub domains and these can be further partitioned and so on.

➢ **Name Server**

Name server contains the **DNS database**. This database comprises of various names and their corresponding IP addresses. Since it is not possible for a single server to

maintain entire DNS database, therefore, the information is distributed among many DNS servers.

- Hierarchy of server is same as hierarchy of names.
- The entire name space is divided into the zones

Zones

Zone is collection of nodes (sub domains) under the main domain. The server maintains a database called zone file for every zone. If the domain is not further divided into sub domains then domain and zone refers to the same thing. The information about the nodes in the sub domain is stored in the servers at the lower levels however; the original server keeps reference to these lower levels of servers.
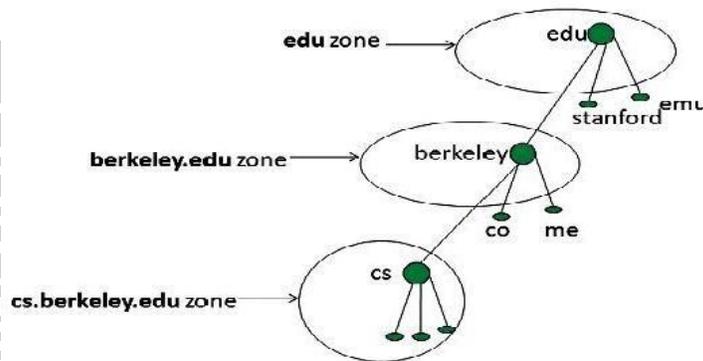

Fig 1.6 Zones in DNS

➢ **Types of Name Servers**

Following are the three categories of Name Servers that manages the entire Domain Name System:

- **Root Server:** Root Server is the top level server which consists of the entire DNS tree. It does not contain the information about domains but delegates the authority to the other server.

- **Primary Server:** Primary Server stores a file about its **zone.** It has authority to create, maintain, and update the zone file.

- **Secondary Server:** Secondary Server transfers complete information about a zone from another server which may be primary or secondary server. The secondary server does not have authority to create or update a zone file.

➢ **DNS Working**

DNS translates the domain name into IP address automatically. Following are the steps in domain resolution process:

1. When we type www.abc.com into the browser, it asks the local DNS Server for its IP address. Here the local DNS is at ISP end.

2. When the local DNS does not find the IP address of requested domain name, it forwards the request to the root DNS server and again enquires about IP address of it.

3. The root DNS server replies with delegation that 'I do not know the IP address of www.abc.com but know the IP address of DNS Server'.

4. The local DNS server then asks the com DNS Server the same question.

5. The com DNS Server replies the same that it does not know the IP address of www.abc.com but knows the address of server that contains www.abc.com.

6. Then the local DNS asks the abc.com DNS server the same question.

7. Then abc.com DNS server replies with IP address of www.abc.com.

8. Now, the local DNS sends the IP address of www.abc.com to the computer that sends the request.

9. When a DNS server receives a DNS reply it cache the information in the reply in its local memory. This is called **DNScache.**

HTTP Request Message

Whenever a URL is entered in the address box of the browser, the browser translates the URL into a request message according to the HTTP protocol; and sends the request message to the server. When the request message reaches the server, the server can take either one of these actions:

- The server interprets the request received, maps the request into a file under the server's document directory, and returns the file requested to the client.

- The server interprets the request received, maps the request into a program kept in the server, executes the program, and returns the output of the program to the client.

- The request cannot be satisfied, the server returns an error message.

The following are the parts of the HTTP request message:

Request-Line or Start line: This begins with a method token, followed by the Request-URI and the protocol version, and ending with CRLF (Carriage Return and Line Feed). The elements are separated by space.

Request Method: This indicates the method to be performed on the resource identified by the given Request-URI. The method is case-sensitive and should always be mentioned in uppercase. The following are the methods:

- **GET** - return the resource specified by the Request-URI as the body of a response message.

- **POST**- pass the body of this request message on as data to be processed by the resource specified by the Request-URI.

- **HEAD-** return the same HTTP header fields that would be returned if a GET method were used, but not return the message body that would be returned to a GET (this provides information about a resource without the communication overhead of transmitting the body of the response, which may be quite large).

- **OPTIONS**- return (in Allow header field) a list of HTTP methods that may be used to access the resource specified by the Request-URI.

- **PUT**- store the body of this message on the server and assign the specified Request-URI to the data stored so that future GET request messages containing this Request-URI will receive this data in their response messages.

- **DELETE**- respond to future HTTP request messages that contain the specified Request-URI with a response indicating that there is no resource associated with this Request-URI.

- **TRACE**- return a copy of the complete HTTP request message, including start line, header fields, and body, received by the server. Used primarily for test purposes.

Header Fields

The request-header fields allow the client to pass additional information about the request, and about the client itself, to the server. These fields act as request modifiers.Here is a list of some important Request-header fields that can be used based on the requirement: Accept-Charset, Accept-Encoding, Accept-Language, Authorization, Expect, From, Host, If- Match, If-Modified-Since, If-None-Match, If-Range, If-Unmodified-Since, Max-Forwards, Proxy-Authorization, Range, Referer, TE, User-Agent.

Multimedia Internet Mail Extension (MIME) type

MIME are standards that are used to pass variety of types of information through internet message protocol. It has two-parts specified as the content type of the message. **Examples:** text/html and image/jpeg. The following are the content types supported by the HTTP:

- Application- Data that does not fit within another content type and that is intended to be processed by application software, or that is itself an executable binary.

- Audio- Audio data. Subtype defines audio format.

- Image- Image data, typically static. Subtype defines image format. Requires appropriate software and hardware in order to be displayed.

- Message- Another document that represents a MIME-style message.

- Video- Animated images, possibly with synchronized sound.

- Model- Structured data, generally numeric, representing physical or behavioral models.

- Multipart- Multiple entities, each with its own header and body.

- Text- Displayable as text. That is, a human can read this document without the need for special software, although it may be easier to read with the assistance of other software.

```
GET /doc/test.html HTTP/1.1 ──────→ Request Line
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate ──→ Request Headers
User-Agent: Mozilla/4.0
Content-Length: 35

                                 ──→ A blank line separates header & body
bookId=12345&author=Tan+Ah+Teck  ─} Request Message Body
```
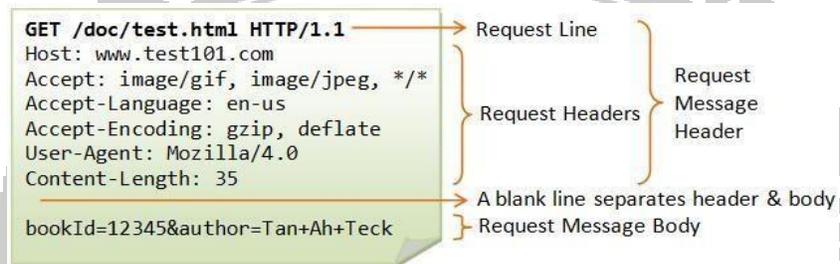
Fig 1.7 HTTP Request Message

HTTP Response Message

The following are contents of HTTP Response message:

- **Status line:** It has three fields namely HTTP version, numeric status code and text string which informs about the information represented by numeric status code.

Status code	Recommended reason phrase	Usual meaning
200	OK	Request processed normally.
301	Moved Permanently	URI for the requested resource has been changed. All future requests should be made to URI contained in the location header field of the response. Most browsers will automatically send a second request to the new URI and display the second response.
307	Temporary redirect	URI for the request has been changed at least temporarily. This request should be fulfilled by making a second request to the URI contained in the location header field of the reqponse.

401	Unauthorized	The resource is password protected, and the user has not yet supplied a valid password.
403	Forbidden	The resource is present on the server but is read protected.
404	Not found	No resource corresponding to the given request-URI was found at this server.
500	Internal server error	Server software detected an internal failure.

- Header field(s) (one or more): It contains Connection, ContentType, and Content-Length, are also valid in response messages. The Content-Type of aresponse can be any one of the MIME type values specified by the Accept header fieldof the corresponding request. Some of the common response header fields are:

  – Host- Specify authority portion of URL

  – User-Agent -A string identifying the browser or other software that is sending the request.

  – Accept MIME- types of documents that are acceptable as the body of the response, possibly with indication of preference ranking.

  – Accept Language- Specifies preferred language(s) for the response body. A server may have several translations of a document, and among these should return the one that has the highest preference rating in this header field.

  – Accept-Encoding- Specifies preferred encoding(s) for the response body.

  – Accept-Charset- Allows the client to express preferences to a server that can return a document using various character sets.

  – Connection- Indicates whether or not the client would like the TCP connection kept open after the response is sent.

  – Keep-Alive- Number of seconds TCP connection should be kept open.

  – Content-Type -The MIME type of the document contained in the message body, if one is present.

  – Content-Length- Number of bytes of data in the message body, if one is present.

  – Referer -The URI of the resource from which the browser obtained the Request-URI value for this HTTP request.

- Blank line

- Message body (optional)

Cache control: Web browsers automatically cache on the client machine many of the resources that they request from servers via HTTP. But information in a cache can become invalid. One approach to guaranteeing that a cached copy of a resource is valid is for the client to ask the server whether or not the client's copy is valid. This can be done with relatively little communication by sending an HTTP request for the resource using the HEAD method, which returns only the status line and header portion of the response.

Character Sets: A character set defines the mapping between these integers, or code points, and characters. Each US-ASCII character can be represented by a 7-bit integer, which is convenient in part because the messages transmitted by the Internet Protocol are viewed as streams of 8-bit bytes, and therefore each character can be represented by a single byte. The Unicode Standard's Basic Multilingual Plane (BMP), uses characters in every modern language, uses 16-bit character codes, and the full character code space of the Unicode Standard extends to 21-bit integers.
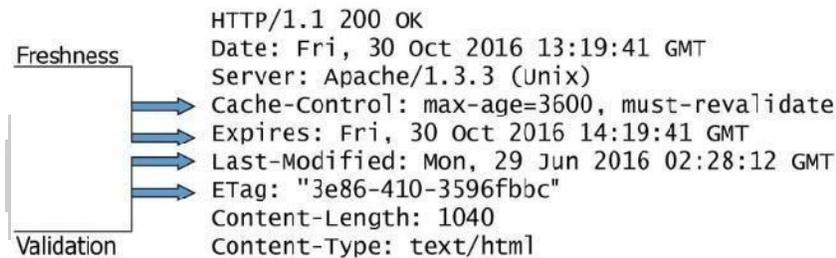
```
                    HTTP/1.1 200 OK
                    Date: Fri, 30 Oct 2016 13:19:41 GMT
Freshness           Server: Apache/1.3.3 (Unix)
              ───▶  Cache-Control: max-age=3600, must-revalidate
              ───▶  Expires: Fri, 30 Oct 2016 14:19:41 GMT
              ───▶  Last-Modified: Mon, 29 Jun 2016 02:28:12 GMT
              ───▶  ETag: "3e86-410-3596fbbc"
                    Content-Length: 1040
Validation          Content-Type: text/html
```

Fig 1.8 HTTP Response