Hyper Text Transfer Protocol (HTTP)

- The **Hyper Text Transfer Protocol (HTTP)** is used to define how the client-server programs can be written to retrieve web pages from the Web.
- 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.

Non persistent versus Persistent Connections:

Non persistent Connections:

- In a **non persistent 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.

Persistent Connections:

- 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.
- 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.

Message Formats:

- The HTTP protocol defines the format of the request and response messages. The first section in the request message is called the*request line;* the first section in the response message is called the *status line*.
- The otherthree sections have the same names in the request and response messages.

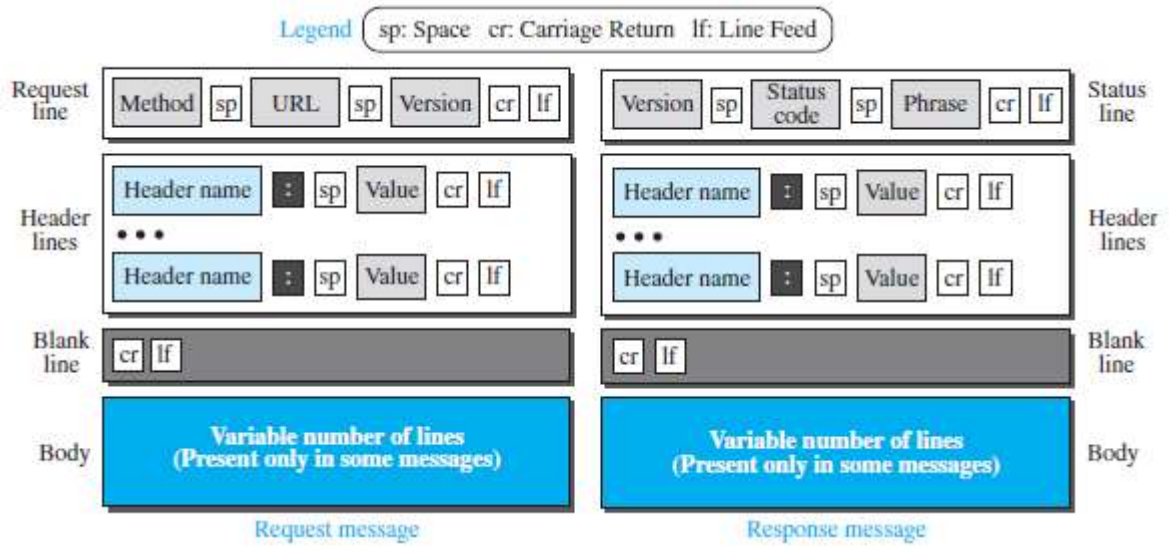

Fig: Formats of the request and response messages.

Request Message:

- The first line in a request message is called a request line. There are three fields in this line separated by one space. The fields are called *method, URL*,and *version*.The method field defines the request types.
- The second field, URL. It defines the address and name of the corresponding web page. The third field, version, gives the version of the protocol; the most current version of HTTP is 1.1.

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.

Table: Methods.

- After the request line, we can have zero or more *request header* lines. Each header line sends additional information from the client to the server.
- The body can be present in a request message.

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 (explained later)
If-Modified-Since	If the file is modified since a specific date

Table: Request header names.

Response Message:

- A response message consists of a status line, header lines, a blank line, and sometimes a body. The first line in a response message is called the *status line*.
- There are three fields in this line separated by spaces and terminated by a carriage return and line feed. The first field defines the version of HTTP protocol, currently 1.1.
- The status code field defines the status of the request. It consists of three digits. Whereas the codes in the100 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. 400 range indicate an error at the client site. Finally, the codes in the 500 range indicate an error at the server site. the status line, we can have zero or more *response header* lines.
- Each headerline sends additional information from the server to the client.

Header	Description
Date	Shows the current date
Upgrade	Specifies the preferred communication protocol
Server	Gives information about the server
Set-Cookie	The server asks the client to save a cookie
Content-Encoding	Specifies the encoding scheme

Content-Language	Specifies the language
Content-Length	Shows the length of the document
Content-Type	Specifies the media type
Location	To ask the client to send the request to another site
Accept-Ranges	The server will accept the requested byte-ranges
Last-modified	Gives the date and time of the last change

Table: Response header names

Cookies:

Creating and Storing Cookies:

- The creation and storing of cookies depend on the implementation; however, the principle is the same.
    1. When a server receives a request from a client, it stores information about the client in a file or a string. The information may include the domain name of the client, the contents of the cookie (information the server has gathered about the client such as name, registration number, and so on), a timestamp, and other information depending on the implementation.
    2. The server includes the cookie in the response that it sends to the client.
    3. When the client receives the response, the browser stores the cookie in the cookie directory, which is sorted by the server domain name.

Using Cookies:

- When a client sends a request to a server, the browser looks in the cookie directory to see if it can find a cookie sent by that server. If found, the cookie is included in the request. When the server receives the request, it knows that this is an old client, not anew one.
    - An *electronic store* (e-commerce) can use a cookie for its client shoppers. When a client selects an item and inserts it in a cart, a cookie that contains information about the item, such as its number and unit price, is sent to the browser. If the client selects a second item, the cookie is updated with the new selection information, and so on. When the client finishes shopping and wants to check out, the last cookie is retrieved and the total charge is calculated.
    - The site that restricts access to *registered clients* only sends a cookie to the clientwhen the client registers for the first time. For any repeated access, only those clientsthat send the appropriate cookie are allowed.

- A web *portal* uses the cookie in a similar way. When a user selects her favorite pages, a cookie is made and sent. If the site is accessed again, the cookie is sent to the server to show what the client is looking for.
- A cookie is also used by *advertising* agencies. An advertising agency can place bannerads on some main website that is often visited by users.

Web Caching: Proxy Servers:

- 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. proxy server acts as both server and client.
- When it receives a request from a client for which it has a response, it acts as a server and sends the response to the client. When it receives a request from a client for which it does not have a response, it first acts as a client and sends a request to the target server.
- When the response has been received, it acts again as a server and sends the response to the client.

Proxy Server Location:

- The proxy servers are normally located at the client site.

1. A client computer can also be used as a proxy server, in a small capacity, thatstores responses to requests often invoked by the client.
2. In a company, a proxy server may be installed on the computer LAN to reduce the load going out of and coming into the LAN.
3. An ISP with many customers can install a proxy server to reduce the load going out of and coming into the ISP network.

Cache Update:

- A very important question is how long a response should remain in the proxy server before being deleted and replaced. Several different strategies are used for this purpose.

- One solution is to store the list of sites whose information remains the same for a while.
- Another recommendation is to add some headers to show the last modification time of the information.

HTTP Security:

- HTTP canbe run over the Secure Socket Layer (SSL). In this case, HTTP is referred to as HTTPS.HTTPS provides confidentiality, client and server authentication, and data integrity.