
18CSC302J- Computer Networks

Unit-3

Syllabus

1. DNS- DNS in the Internet, DNS Resolution, DNS Messages
2. TELNET – SSH
3. FTP- TFTP
4. **WWW Architecture, Web Documents**
5. **HTTP, HTTP Request and Reply**
6. DHCP Operation, DHCP Configuration
7. SMTP, POP3, IMAP, MIME

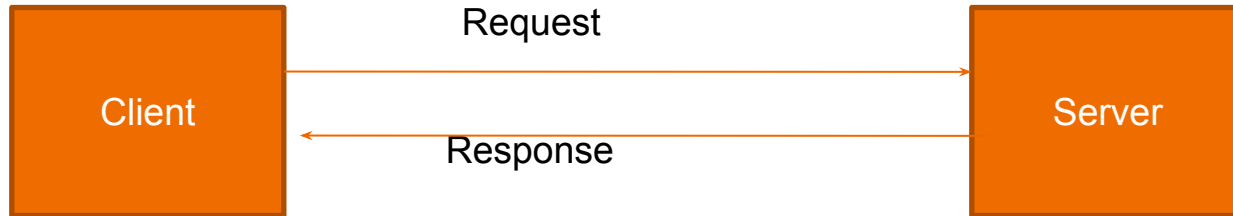
Learning Resources

1. Douglas E. Comer, Internetworking with TCP/IP, Principles, protocols, and architecture, Vol 1 5th Edition, 2006 ISBN: 0131876716, ISBN: 978-0131876712

WWW Architecture

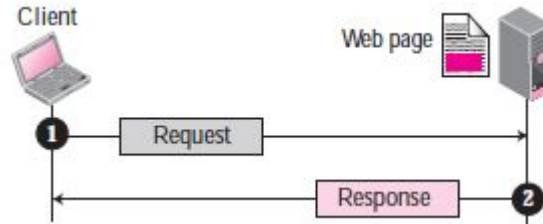
WWW Architecture

- WWW is a networked information system (repository of information)
- The WWW provides 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 locations called **sites**.
- Each site holds one or more documents, referred to as Web pages.
- Each Web page, however, can contain some links to other Web pages in the same or other sites.
- In other words, a Web page can be **simple or composite**.
- A simple Web page has no link to other Web pages;
- A composite Web page has one or more links to other Web pages.
- Each Web page is a file with a name and address.



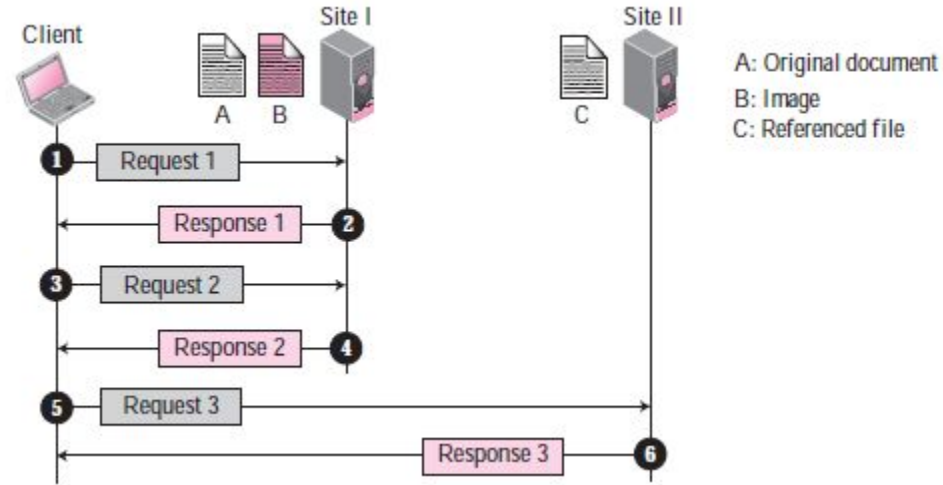
Example-1

- ❑ Retrieve a Web page that contains the biography of a famous character with some pictures, which are embedded in the page itself.
- ❑ Since the pictures are not stored as separate files, the whole document is a simple Web page.
- ❑ It can be retrieved using one single request/response transaction,



Example-2

- ❑ Assume we need to retrieve a scientific document that contains one reference to another text file and one reference to a large image
- ❑ The main document and the image are stored in two separate files in the same site (file A and file B);
- ❑ the referenced text file is stored in another site (file C).
- ❑ Since we are dealing with three different files,
- ❑ we need three transactions if we want to see the whole document.
- ❑ The first transaction (request/response) retrieves a copy of the main document (file A), which has a reference (pointer) to the second and the third files.

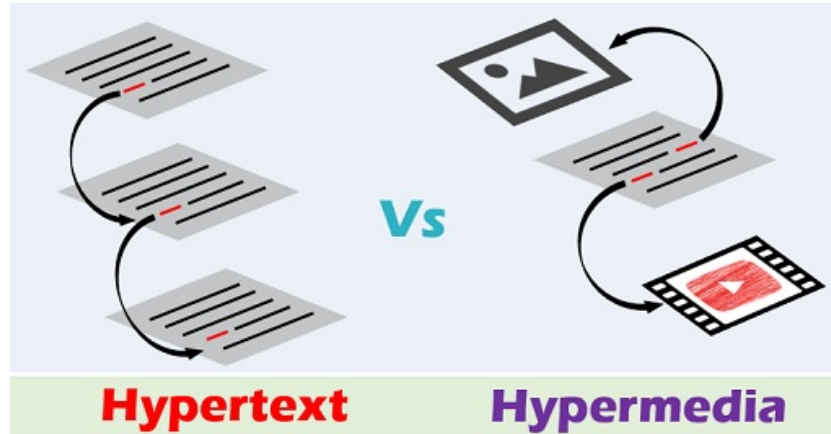


Example-3

- ❑ A very important point we need to remember is that file A, file B, and file C in previous Example are independent Web pages, each with independent names and addresses.
- ❑ Although references to file B or C are included in file A, it does not mean that each of these files cannot be retrieved independently.
- ❑ A second user can retrieve file B with one transaction.
- ❑ A third user can retrieve file C with one transaction.

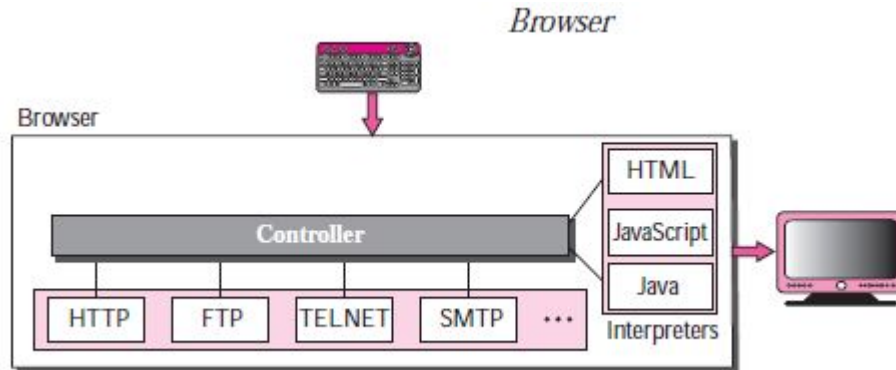
Hypertext and Hypermedia

- Hypertext –creating a document that in turn refer to other document. In a hypertext document, a part of text can be defined as a link to another document.
- Hypermedia is a term applied to document that contains links to other textual document or documents containing graphics, video, or audio.



Web Client (Browser)

- It is an application software that allows us to view and explore information on the web.
- User can request for any web page by just entering a URL into address bar.
- Web browser can show text, audio, video, animation and more
- It is the responsibility of a web browser to interpret text and commands contained in the web page.
- A variety of vendors offer commercial browsers that interpret and display a Web document, and all of them use nearly the same architecture. Each browser usually consists of three parts:
 - a controller – receives input from keyboard
 - client protocol – access the document
 - Interpreters – display document on screen



Web Server

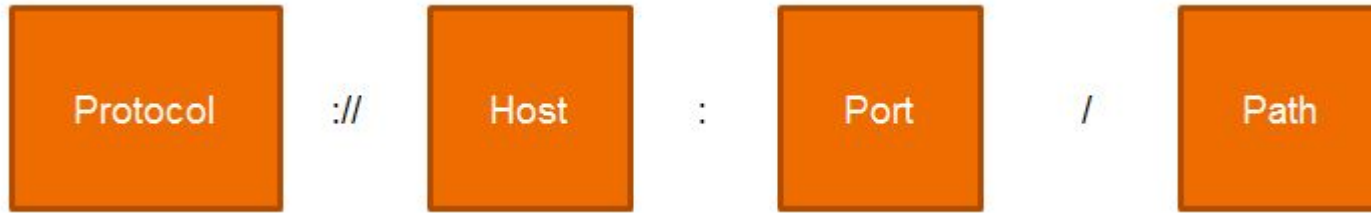
- Web site is collection of web pages while web server is a software that respond to the request for web resources.
- When client sends request for a web page, the web server search for the requested page if requested page is found then it will send it to client with an HTTP response.
- If the requested web page is not found, web server will the send an HTTP response : Error 404 Not found.
- A server can also become more efficient through multithreading or multiprocessing.
- In this case, a server can answer more than one request at a time.
- Some popular Web servers include Apache and Microsoft Internet Information Server.

Uniform Resource Locator (URL)

- A client that wants to access a Web page needs the file name and the address.
- To facilitate the access of documents distributed throughout the world, HTTP uses locators.
- The uniform resource locator (URL) is a standard locator for specifying any kind of information on the Internet.
- The URL defines four things: protocol, host computer, port, and path

Uniform Resource Locator (URL)

- A URL (Uniform Resource Locator) is a unique identifier used to locate a resource on the internet.



- Protocol - client-server application program used to retrieve the document (http)
- Host - domain name of the computer on which the information is located (www)
- Port - (optional) 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 - pathname of the file where the information is located.



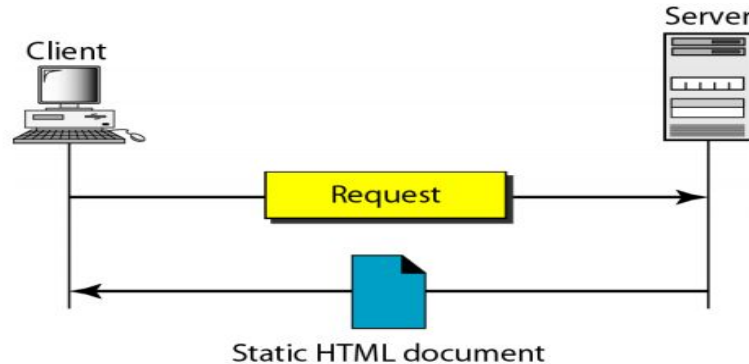
WWW Documents (Web Documents)

Web Documents

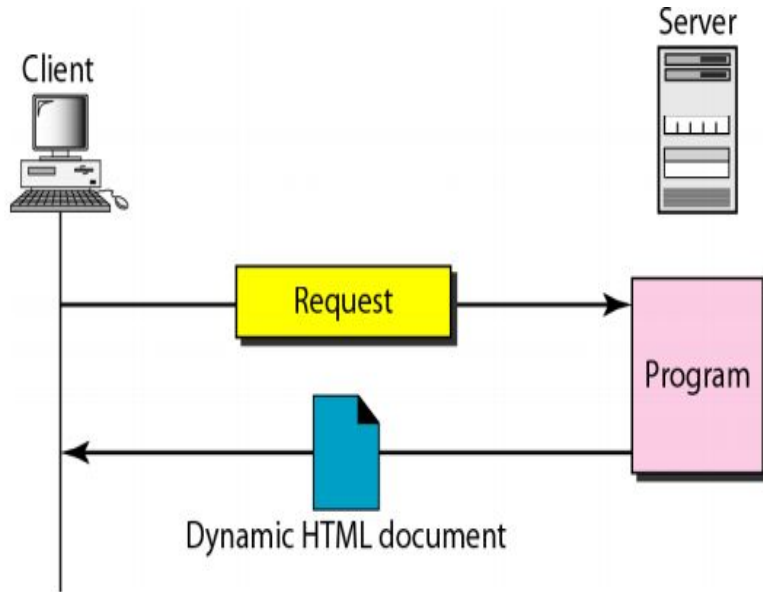
- The documents in the WWW can be grouped into three broad categories: static, dynamic, and active.
- The category is based on the time the contents of the document are determined.

Static Documents

- Static documents are fixed-content documents that are created and stored in a server.
- The client can get a copy of the document only.
- In other words, the contents of the file are determined when the file is created, not when it is used.
- Of course, the contents in the server can be changed, but the user cannot change them.
- When a client accesses the document, a copy of the document is sent.
- The user can then use a browsing program to display the document
- Since the contents do not change, each request for a static document results in exactly the same response.
- Static documents are prepared using –Hypertext Markup Language (HTML), Extensible Markup Language (XML), Extensible Style Language (XSL), and Extended Hypertext Markup Language (XHTML)



Dynamic Documents



- A dynamic web document does not exist in a predefined form.
- A dynamic document is created by a Web server whenever a browser requests the document.
- When a request arrives, the Web server runs an-application program or a script that creates the dynamic document.
- The server returns the output of the program or script as a response to the browser that requested the document.
- Because a fresh document is created for each request, the contents of a dynamic document may vary from one request to another.
- A very simple example of a dynamic document is the retrieval of the time and date from a server.
- Time and date are kinds of information that are dynamic in that they change from moment to moment.

Dynamic Documents

- The **Common Gateway Interface (CGI)** is a technology that creates and handles dynamic documents. CGI is a set of standards that defines how a dynamic document is written, how data are input to the program, and how the output result is used.
- CGI is not a new language; instead, it allows programmers to use any of several languages such as C, C ++, Bourne Shell, Korn Shell, C Shell, Tcl, or Perl.
- The only thing that CGI defines is a set of rules and terms that the programmer must follow.
- The term common in CGI indicates that the standard defines a set of rules that is common to any language or platform.
- The term gateway here means that a CGI program can be used to access other resources such as databases, graphic packages, and so on.
- The term interface here means that there is a set of predefined terms, variables, calls, and so on that can be used in any CGI program.
- A CGI program in its simplest form is code written in one of the languages supporting CGI. Any programmer who can encode a sequence of thoughts in a program and knows the syntax of one of the abovementioned languages can write a simple CGI program.

Dynamic Documents

Input

- The input from a browser to a server is sent using a form.
- If the information in a form is small (such as a word), it can be appended to the URL after a question mark. For example, the following URL is carrying form information (23, a value):

```
http://www.deanza/cgi-bin/prog.pl?23
```

- When the server receives the URL, it uses the part of the URL before the question mark to access the program to be run, and it interprets the part after the question mark (23) as the input sent by the client. It stores this string in a variable.
- When the CGI program is executed, it can access this value.
- If the input from a browser is too long to fit in the query string, the browser can ask the server to send a form.
- The browser can then fill the form with the input data and send it to the server.
- The information in the form can be used as the input to the CGI program.

Dynamic Documents

Output

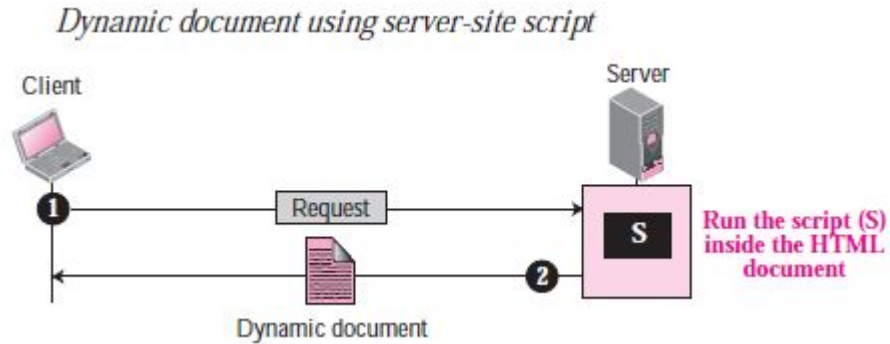
- The whole idea of CGI is to execute a CGI program at the server site and send the output to the client (browser).
- The output is usually plain text or a text with HTML structures; however, the output can be a variety of other things.
- It can be graphics or binary data, a status code, instructions to the browser to cache the result, or instructions to the server to send an existing document instead of the actual output.

Dynamic Documents

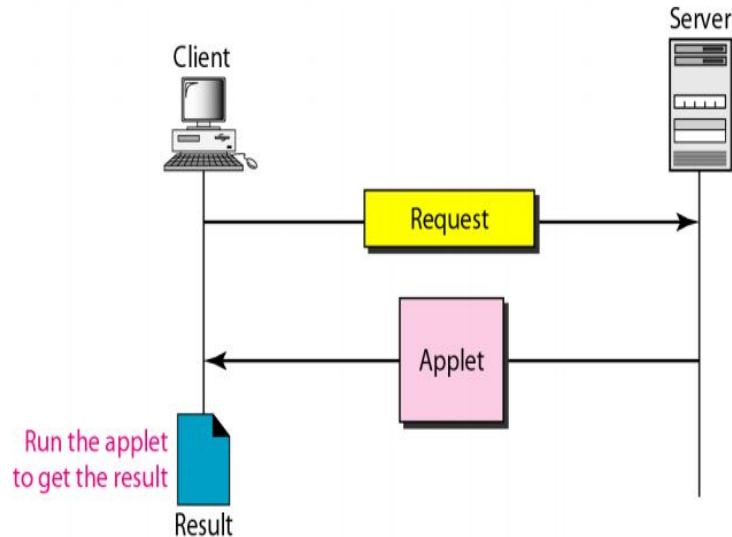
Scripting Technologies for Dynamic Documents

- The problem with CGI technology is the inefficiency that results if part of the dynamic document that is to be created is fixed and not changing from request to request.
- If we use CGI, the program must create an entire document each time a request is made.
- The solution is to create a file containing the fixed part of the document using HTML and embed a script, a source code, that can be run by the server to provide the varying availability and price section.
- A few technologies have been involved in creating dynamic documents using scripts.
- Among the most common are Hypertext Preprocessor (PHP), which uses the Perl language; Java Server Pages (JSP), which uses the Java language for scripting; Active Server Pages (ASP), a Microsoft product, which uses Visual Basic language for scripting; and ColdFusion, which embeds SQL database queries in the HTML Document.

Dynamic Documents



Active Documents



- An active web document consists of a computer program that the server sends to the browser and that the browser must run locally.
- For example, suppose we want to run a program that creates animated graphics on the screen or a program that interacts with the user.
- When it runs, the active document program can interact with the user and change the display continuously.
- Active documents are sometimes referred to as client-site dynamic documents.

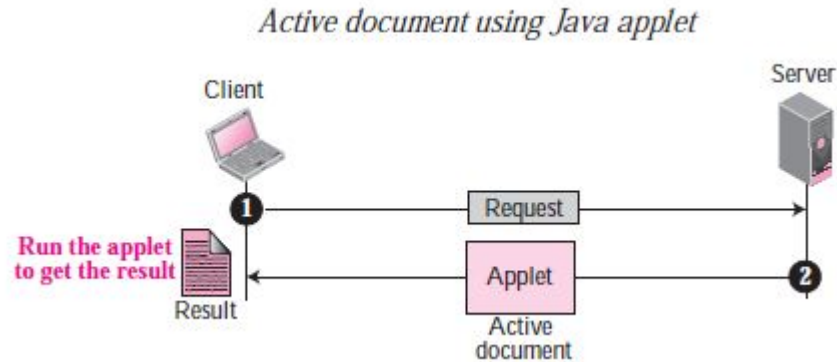
Active Documents

Java Applets

- An active web document consists of a computer program that the server sends to the browser and that the browser must run locally.
- One way to create an active document is to use Java applets.
- Java is a combination of a high-level programming language, a run-time environment, and a class library that allows a programmer to write an active document (an applet) and a browser to run it.
- It can also be a stand-alone program that doesn't use a browser.
- An applet is a program written in Java on the server.
- It is compiled and ready to be run.
- The document is in bytecode (binary) format.
- The client process (browser) creates an instance of this applet and runs it.

Active Documents

- A Java applet can be run by the browser in two ways.
- In the first method, the browser can directly request the Java applet program in the URL and receive the applet in binary form.
- In the second method, the browser can retrieve and run an HTML file that has embedded the address of the applet as a tag.



Active Documents

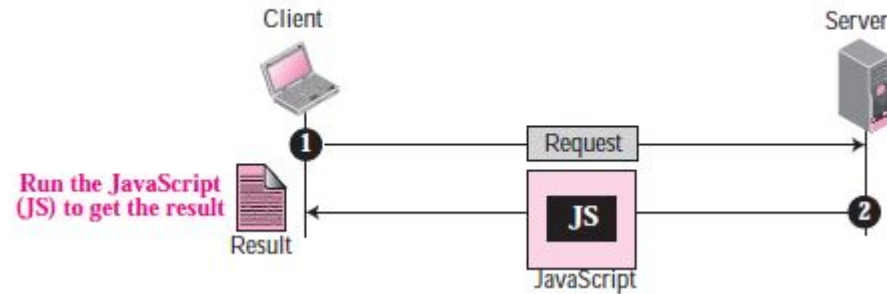
JavaScript

- The idea of scripts in dynamic documents can also be used for active documents.
- If the active part of the document is small, it can be written in a scripting language; then it can be interpreted and run by the client at the same time.
- The script is in source code (text) and not in binary form.
- The scripting technology used in this case is usually JavaScript.
- JavaScript, which bears a small resemblance to Java, is a very high level scripting language developed for this purpose.

Active Documents

JavaScript

Active document using client-site script





HTTP

HTTP

- The Hypertext Transfer Protocol (HTTP) is a protocol used mainly to access data on the WWW.
- HTTP functions as a combination of FTP and SMTP.
- It is similar to FTP because it transfers files and uses the services of TCP.
- However, it is much simpler than FTP because it uses only one TCP connection.
- There is no separate control connection; only data are transferred between the client and the server.
- HTTP is like SMTP because the data transferred between the client and the server look like SMTP messages.
- In addition, the format of the messages is controlled by MIME-like headers.
- Unlike SMTP, the HTTP messages are not destined to be read by humans; they are read and interpreted by the HTTP server and HTTP client (browser).
- SMTP messages are stored and forwarded, but HTTP messages are delivered immediately.
- The commands from the client to the server are embedded in a request message.
- The contents of the requested file or other information are embedded in a response message.
- HTTP uses the services of TCP on well-known port 80.

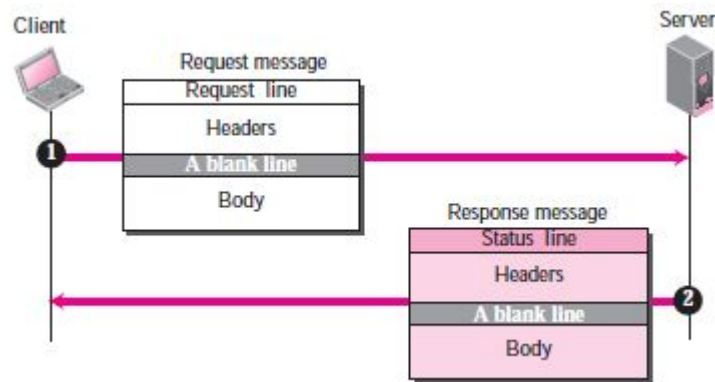
HTTP Transaction

Request message-

- A request message consists of a request line, a header, and sometimes a body.

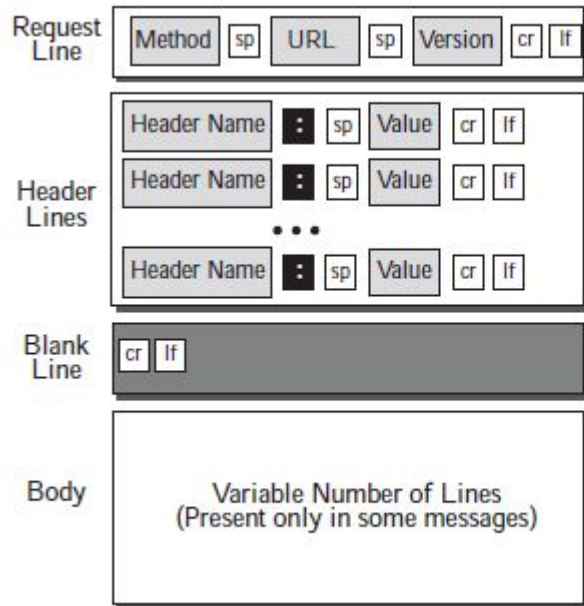
Request Line-

- The first line in a request message is called a request line.
- There are three fields in this line separated by some character delimiter.
- The fields are called methods, URL, and Version.
- These three should be separated by a space character.
- At the end two characters, a carriage return followed by a line feed terminate the line.
- The method field defines the request type.



HTTP Transaction (Cont...)

- Format of request message



Legend

sp: Space
cr: Carriage Return
lf: Line Feed

<i>Method</i>	<i>Action</i>
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
DELETE	Remove the Web page
OPTIONS	Enquires about available options

HTTP Transaction (Cont...)

Header Lines In Request Message

- 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.
- For example, the client can request that the document be sent in a special format.
- Each header line has a header name, a colon, a space, and a header value

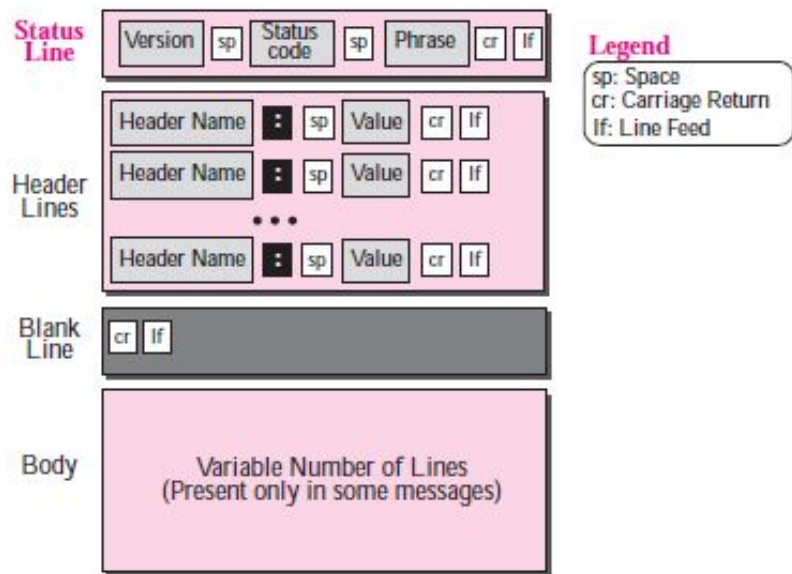
Body In Request Message

- The body can be present in a request message.
- Usually, it contains the comment to be sent.

<i>Header</i>	<i>Description</i>
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	Returns the cookie to the server

HTTP Transaction (Cont...)

- **Format of response message**



Response Message

- A response message consists of a status line, header lines, a blank line and sometimes a body.

Status Line

- The first line in a response message is called the status line.

Header Lines In Response Message

- After the status line, we can have zero or more response header lines.
- Each header line sends additional information from the server to the client.
- Each header line has a header name, a colon, a space, and a header value

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 Transaction (Cont...)

Response Header Names

<i>Header</i>	<i>Description</i>
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

Status Codes and Status Phrases

<i>Status Code</i>	<i>Status Phrase</i>	<i>Description</i>
Informational		
100	Continue	The initial part of the request received, continue.
101	Switching	The server is complying to switch protocols.
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 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.

Conditional Request

- A client can add a condition in its request.
- The server will send the requested Web page if the condition is met or inform the client otherwise.
- The client can send the header line **If-Modified-Since** to the request to tell the server that it needs the page if it is modified after a certain point of time.
- Example conditions - time and date the Web page is modified.

- Request

GET http://www.commonServer.com/information/file1 HTTP/1.1

If-Modified-Since: Thu, Sept 04 00:00:00 GMT

- Response

HTTP/1.1 304 Not Modified

Date: Sat, Sept 06 08 16:22:46 GMT

Server: commonServer.com

(Empty Body)

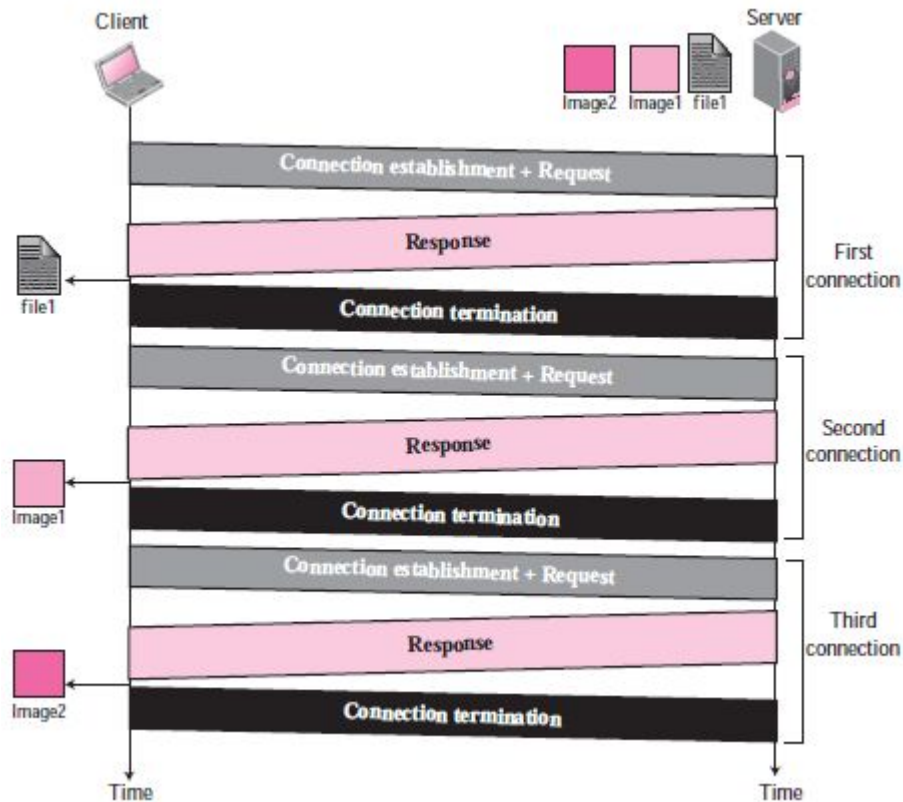
Persistence

- HTTP version 1.1 specifies a persistent connection by default.
- Connection is left open for more requests.
- Connection will be closed only after a request or if a time-out is reached.
- Length of data is sent by the sender on each response, but if it is unknown (Dynamic documents) then the server informs client and closes the connection.

Persistence

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.



Persistence

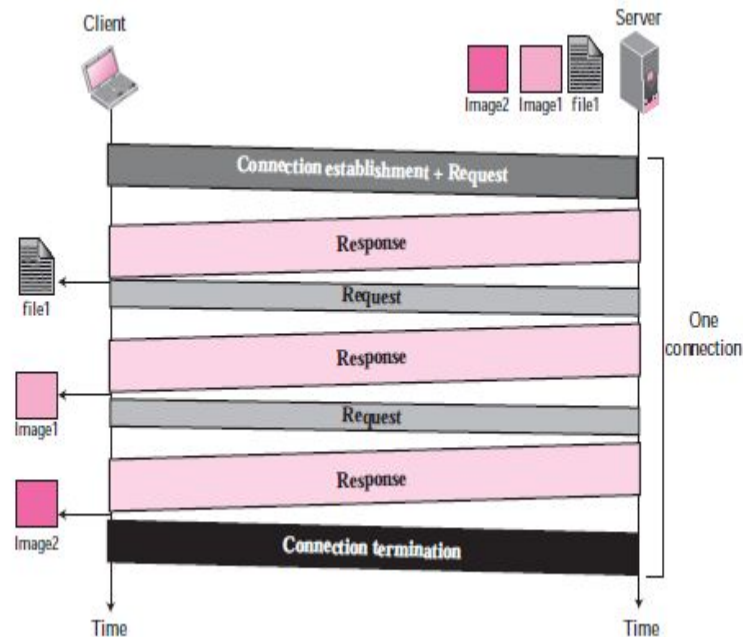
Nonpersistent Connection Problem

- In this strategy, if a file contains link to N different pictures in different files (all located on the same server), the connection must be opened and closed $N + 1$ times.
- The nonpersistent strategy imposes high overhead on the server because the server needs $N + 1$ different buffers and requires a slow start procedure each time a connection is opened.

Persistence

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.



Cookies

- It is a small piece of data stored in users system by the browser while browsing a website.
- When the client receives the response from server on request, the browser stores the cookie in the cookie directory.
- Next time, 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.
- Example – e-commerce

Cookies

- The World Wide Web was originally designed as a stateless entity.
- A client sends a request; a server responds.
- Their relationship is over.
- The original design of WWW, retrieving publicly available documents, exactly fits this purpose.
- Today the Web has other functions; some are listed below:
 - ❑ Websites are being used as electronic stores that allow users to browse through the store, select wanted items, put them in an electronic cart, and pay at the end with a credit card.
 - ❑ Some websites need to allow access to registered clients only.
 - ❑ Some websites are used as portals: The user selects the Web pages he wants to see.
 - ❑ Some websites are just advertising.
- For these purposes, the cookie mechanism was devised.

Cookies

Creating and Storing Cookies

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 domain server name.

Cookies

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 a new one.
- Note that the contents of the cookie are never read by the browser or disclosed to the user.
- It is a cookie made by the server and eaten by the server.
- Purposes of cookies are:
 1. An electronic store (e-commerce) can use a cookie for its client shoppers.
 2. The site that restricts access to registered clients only sends a cookie to the client when the client registers for the first time.
 3. A Web portal uses the cookie in a similar way.
 4. A cookie is also used by advertising agencies.

Web caching: Proxy server

- Proxy server acts as a gateway between client and 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.
- The proxy server acts both as a 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 server 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.

Web caching: Proxy server

Proxy Server Location

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

Cache Update

Question. How long a response should remain in the proxy server before being deleted and replaced.

- One solution is to store the list of sites whose information remains the same for a while.
 - For example, a news agency may change its news page every morning.
 - This means that a proxy server can get the news early in the morning and keep it until the next day.
- Another recommendation is to add some headers to show the last modification time of the information.
 - The proxy server can then use the information in this header to guess how long the information would be valid.

HTTP security

- The HTTP per se does not provide security.
- However, HTTP can be 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.

WWW and HTTP -SUMMARY

•Architecture

- Hypertext and Hypermedia
- Web Client (Browser)
- Web Server
- Uniform Resource Locator (URL)

•Web Documents

- Static Documents
- Dynamic Documents
 - CGI
 - Input, Output
 - Scripting technologies for dynamic documents
- Active Documents
 - Java Applets
 - JavaScripts

•HTTP

- HTTP Transaction
 - Request Message
 - Request Line
 - Header Lines In Request Message
 - Body In Request Message
 - Response Message
 - Status line
 - Header Lines In Response Message
 - Body
- Conditional Request
- Persistence
 - Nonpersistent Connection
 - Persistent Connection
- Cookies
 - Creating and Storing Cookies
 - Using cookies
- Web Caching: Proxy Server
 - Proxy Server Location and Cache Update
- HTTP Security