PES UNIVERSITY ONLINE

COMPUTER COMMUNICATION NETWORKS

Prajeesha

Department of Electronics and Communication Engineering

COMPUTER COMMUNICATION NETWORKS

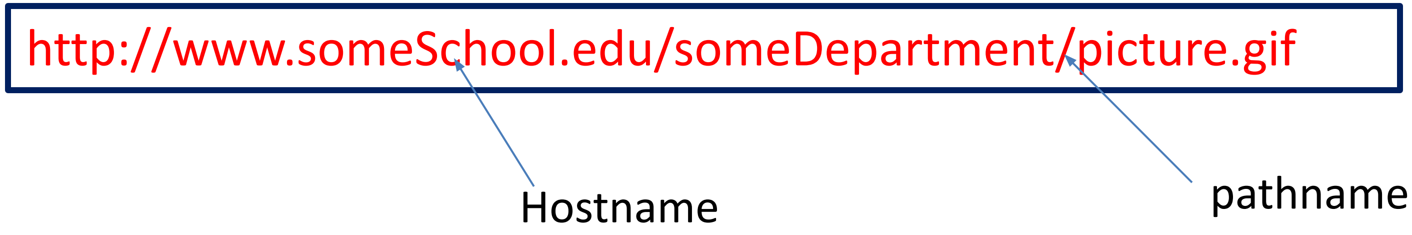
Application layer protocols: Web and HTTP

Prajeesha

Department of Electronics and Communication Engineering

Web Terminologies

- A **Web page** (also called a document) consists of **objects**.
- An **object** is simply **a file** (HTML file, a JPEG image, a Java applet, or a video clip) that is addressable by a single URL.
- Most Web pages consist of a **base HTML file** and several **referenced objects**.
- E.g., If a Web page contains HTML text and five JPEG images, then the Web page has six objects: the base HTML file plus the five images.
- **The base HTML file** references the other **objects in the page with the objects' URLs.**
- Each URL has two components: the **hostname of the server** that houses the object and the object's **path name**



http://www.someSchool.edu/someDepartment/picture.gif

Web application

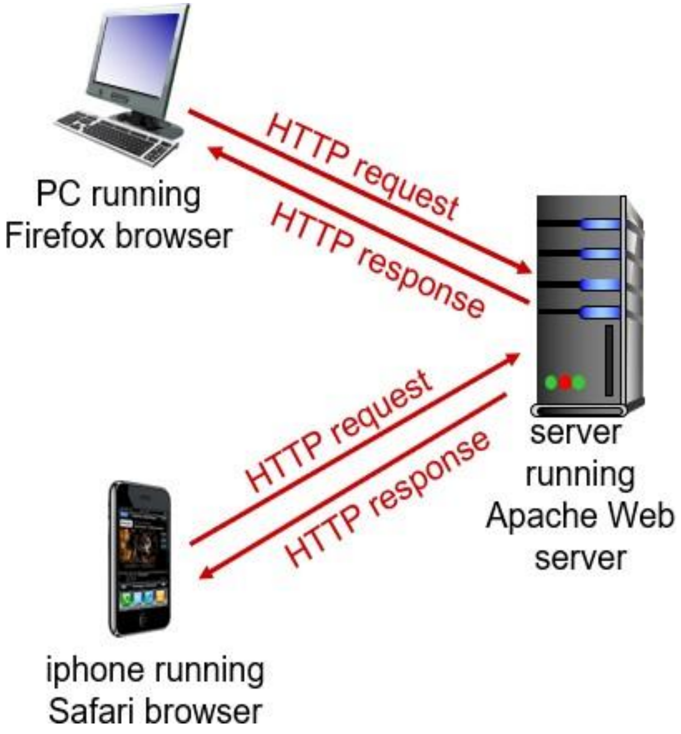
- When we **enter an URL** (e.g., www.someschool.edu) into the browser, the **client process** initiates the fetching of a **HTML file** from the web server.

- The **server process** running in the web server retrieves the file (if present) and replies back to the client process

- If the HTML-file includes several referenced objects (e.g., images), then each referenced object is individually fetched by the client process.

HTTP (HyperText Transfer Protocol)

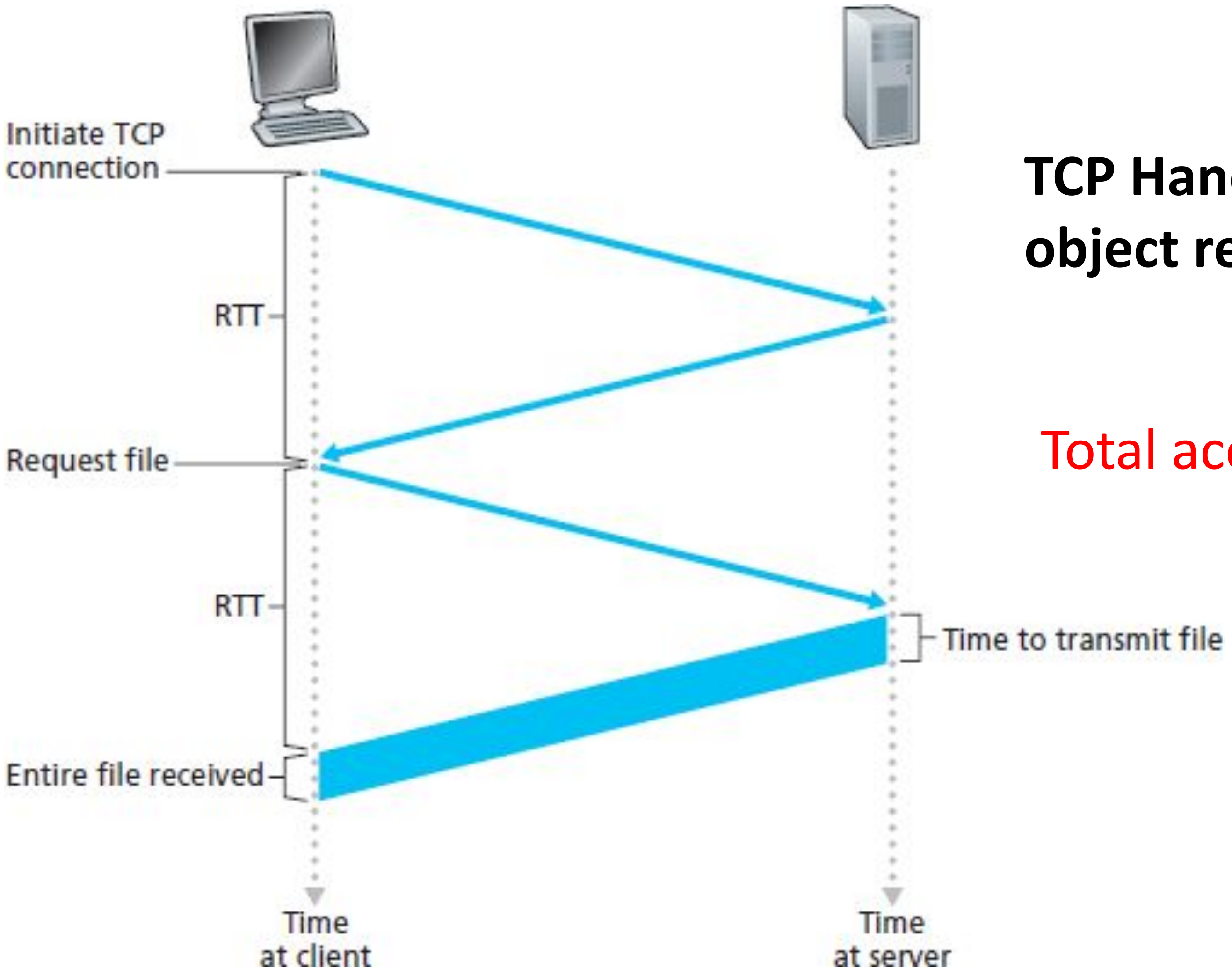
- **HyperText Transfer Protocol (HTTP)** defines the **message formats and rules** for the **client-server process communication** under the web application.
- **Messages** sent by a **client process** (i.e., browser program) are referred to as **HTTP request messages.**
- **Messages** sent by a **server process** (e.g., Apache server, Microsoft server) are referred to as **HTTP response.**
- **HTTP is a stateless protocol**, i.e., server only responds based on the information contained in the given HTTP request message and not on its history.



Requirements of HTTP based communication

- Web applications require reliable transfer of data
  - This is accomplished by the transport layer protocol -**TCP**
  - **Transmission control protocol (TCP)** performs a **3-way handshake** between the client-server processes to ensures every packet exchanged is tracked, corrected and reliably delivered to the corresponding processes
- The default socket (i.e., process ID) for the web server process is 80
- The client process can randomly choose the socket prior to sending the HTTP request message
  - The socket information is carried in the TCP header

Web and HTTP



TCP Handshake and object retrieval

Total access delay = 2×RTT+ L/R

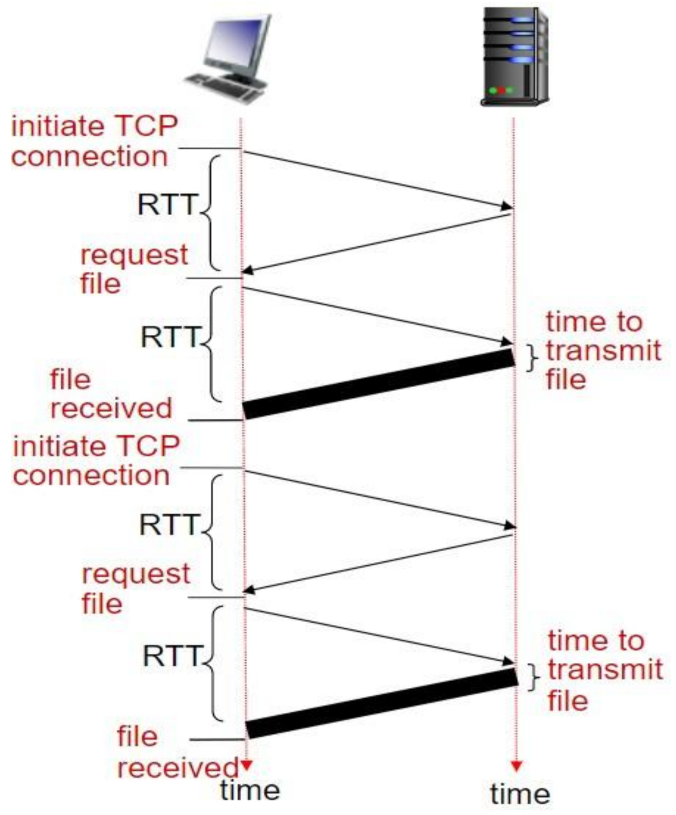
Requirements of HTTP based communication (contd.)

- Web developers choose between two possible ways of conducting the process communication based on HTTP
  - Persistent connection
  - Non-persistent connection
  - Within each type, the referenced objects may be fetched in series or in parallel after the HTML file is retrieved

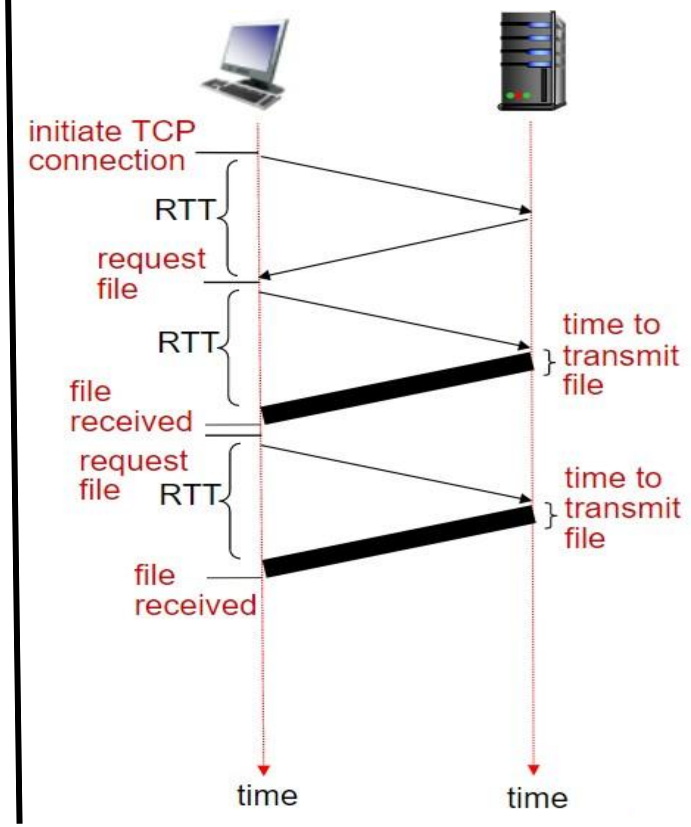
Persistent connection	**Non-persistent connection**
One 3-way handshake to fetch the HTML file and all referenced objects	Separate 3-way handshakes to fetch the HTML file and all referenced objects
Socket of client process is same during the connection	Socket of client process changes for each object during the connection

Web and HTTP

Non-persistent HTTP



Persistent HTTP



Numerical 1:

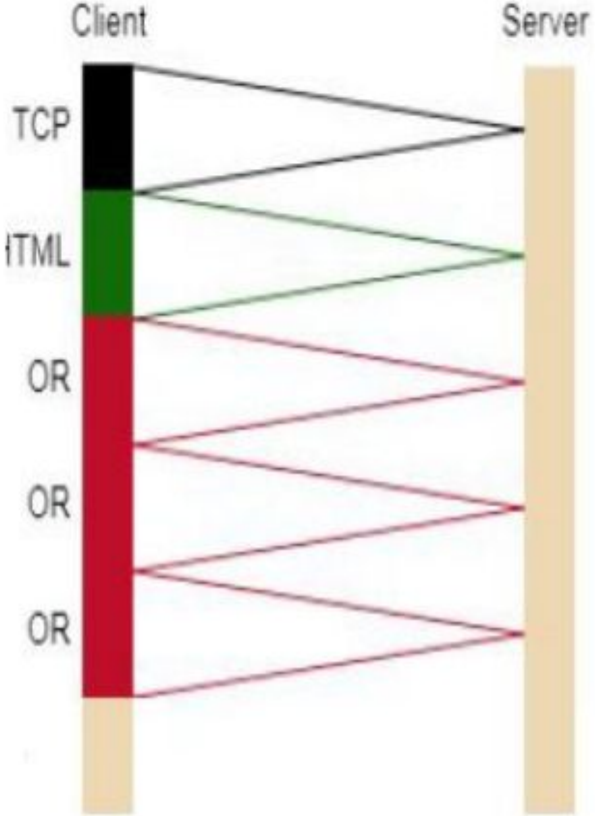
Draw a simple timing diagram indicating the delays involved in retrieving a web page containing the base HTML object and 3 additional images using

(i) Persistent HTTP
(ii) Non-persistent HTTP and
(iii) Persistent HTTP with three parallel connections.

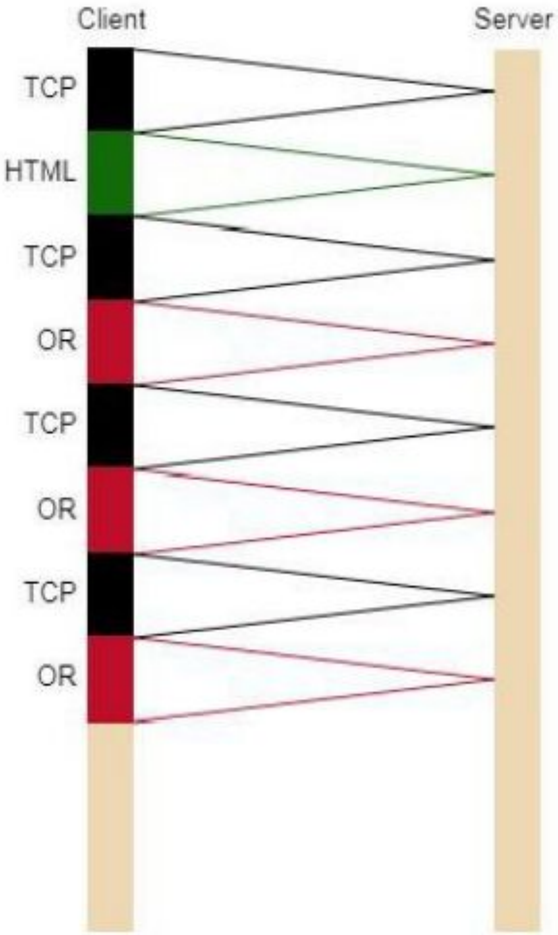
Assume size of all objects to be negligible. Express total delay in terms of RTT.

Solution:

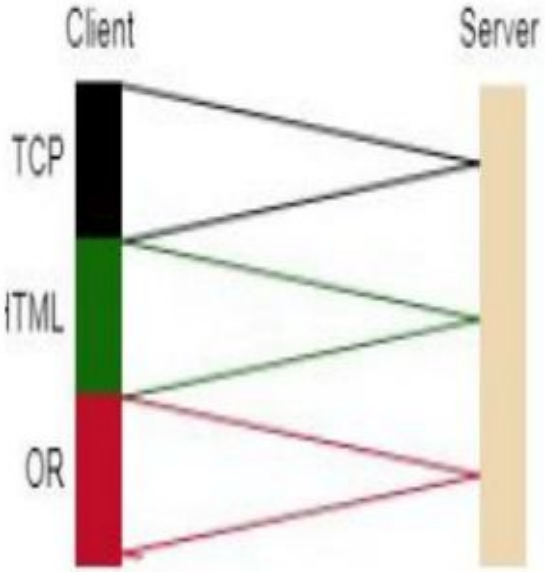
Persistent HTTP
5 RTT



Non-persistent HTTP
8 RTT



Persistent with 3 parallel
3 RTT



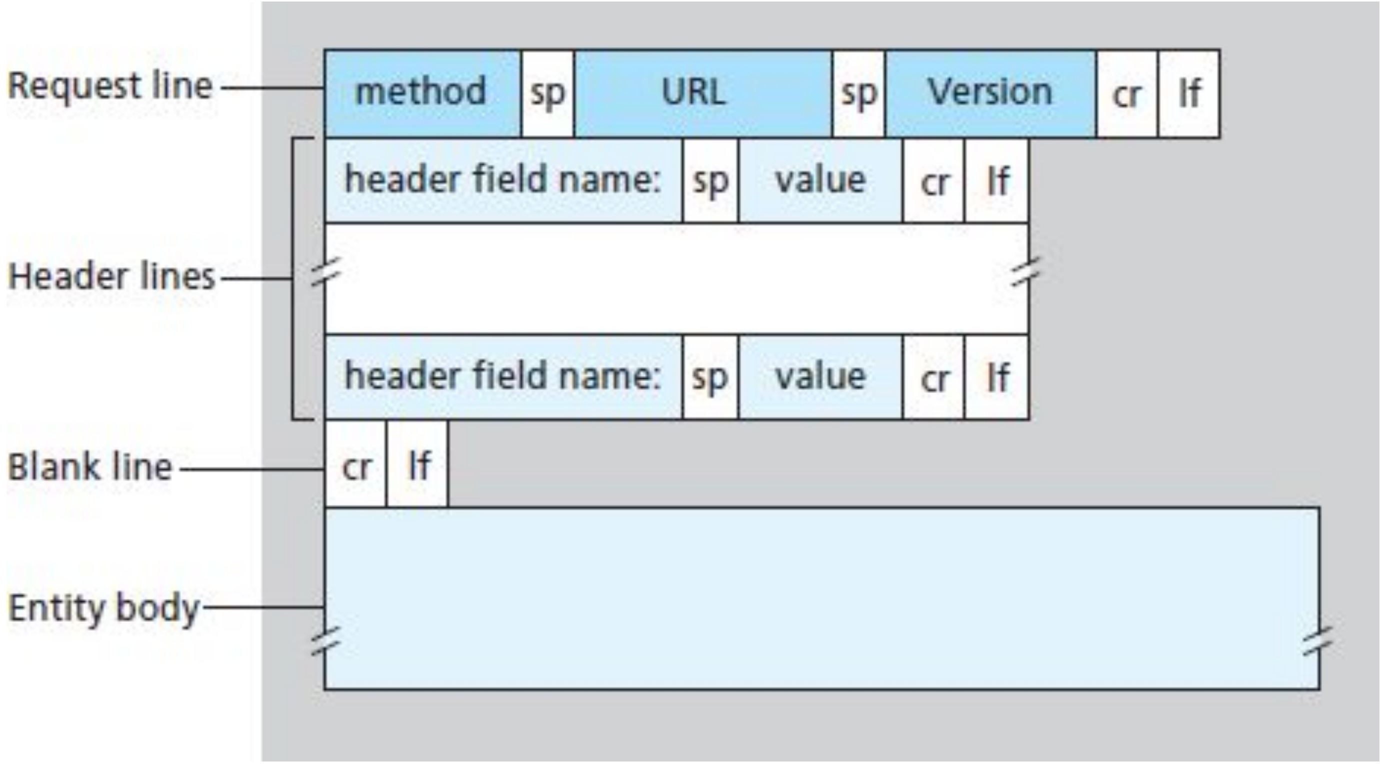
HTTP Message Format

There are two types of HTTP messages, **request** messages and **response** messages

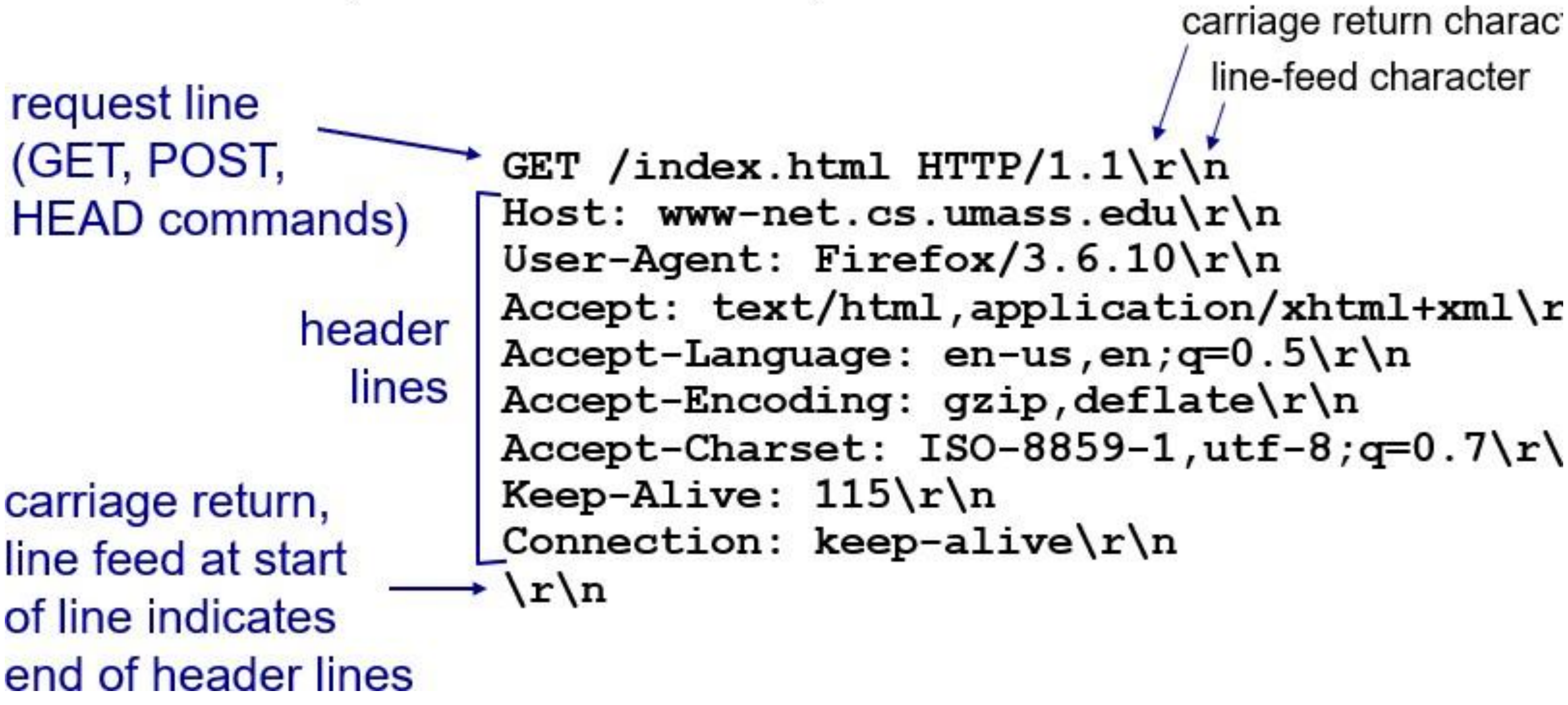
HTTP Request message

- The request message consists of five lines, each followed by a carriage return and a line feed.
- The last line is followed by an additional carriage return and line feed.
- The first line of an HTTP request message is called the **request line**; the subsequent lines are called the **header lines**.
- The request line has three fields: the **method** field, the **URL** field, and the **HTTP version** field.
- Method field values can be **GET**, **POST, HEAD, PUT**, and **DELETE.**

HTTP Request Message Format



Example of HTTP Request Message



```
GET /index.html HTTP/1.1\r\n
Host: www-net.cs.umass.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml\r
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
\r\n
```

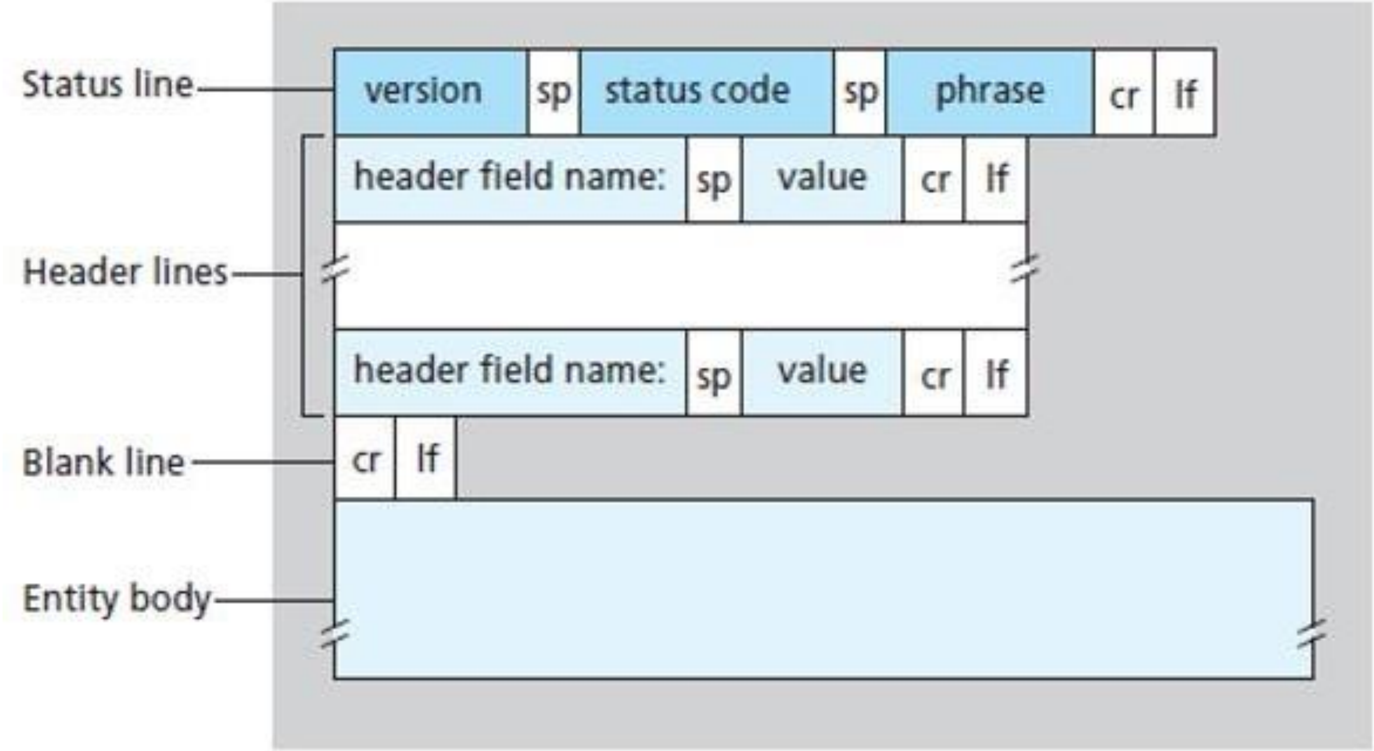
HTTP Response Message Format

- Response message has three sections: an **initial status line**, **six header lines**, and the **entity body**.
- The entity body is the meat of the message which contains the requested object itself (represented by data data data data data ...).
- The status line has **three** fields: the **protocol version** field, a **status code**, and a corresponding **status** message.
- **Connection: close** header line tells the client that it is going to close the TCP connection after sending the message.
- **Date:** header line indicates the time and date when the HTTP response was created and sent by the server.

HTTP Response Message Format

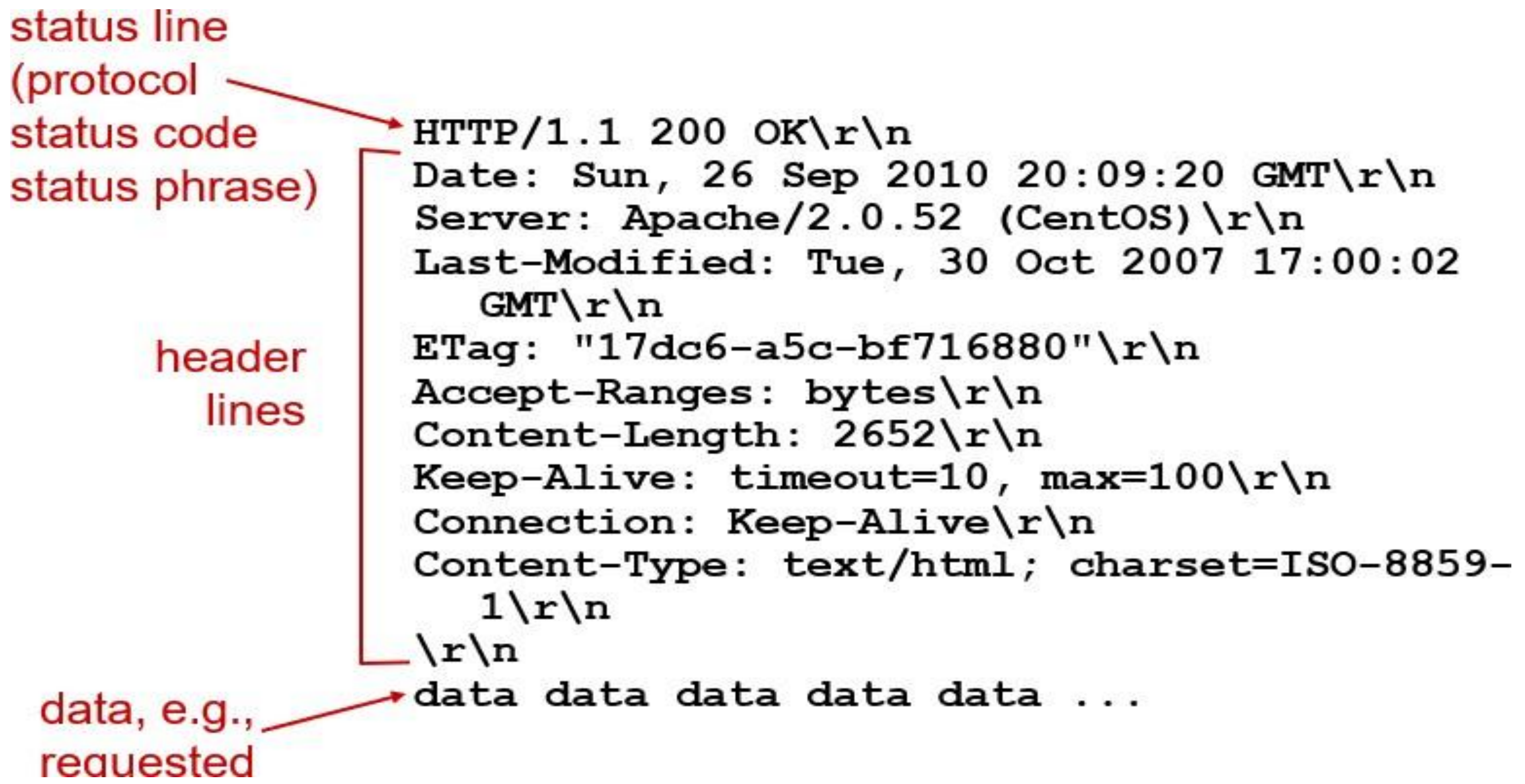
- **Server**: header line indicates that the message was generated by an Apache Web server
- **User-agent**: header line in the HTTP request message.
- **Last-Modified**: header line indicates the time and date when the object was created or last modified
- **Content-Length:** header line indicates the number of bytes in the object being sent.
- **Content-Type**: header line indicates that the object in the entity body is HTML text

HTTP Response Message Format



Status line
version sp status code sp phrase cr lf
Header lines
header field name: sp value cr lf
header field name: sp value cr lf
Blank line
cr lf
Entity body

Example: HTTP Response Message



```
HTTP/1.1 200 OK\r\n
Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Tue, 30 Oct 2007 17:00:02
   GMT\r\n
ETag: "17dc6-a5c-bf716880"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 2652\r\n
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=ISO-8859-
   1\r\n
\r\n
data data data data data ...
```

Examples of status codes and phrase in HTTP response

200 OK

- request succeeded, requested object later in this msg

301 Moved Permanently

- requested object moved, new location specified later in this msg
  (Location:)

400 Bad Request

- request msg not understood by server

404 Not Found

- requested document not found on this server

505 HTTP Version Not Supported

Conditional GET

- HTTP has a mechanism that allows a cache to verify that its objects are up to date. This mechanism is called the **conditional GET**.
- An HTTP request message is a so-called conditional GET message if
  (1) the request message uses the GET method and
  (2) the request message includes an If-Modified-Since: header line

  - **Conditional GET** tells the server to send the object only if the object has been modified since the specified date.

Conditional GET

On the behalf of a requesting browser, a proxy cache sends a request message to a Web server

```
GET /fruit/kiwi.gif HTTP/1.1
Host: www.exotiquecuisine.com
```

The Web server sends a response message with the requested object to the cache

```
HTTP/1.1 200 OK
Date: Sat, 3 Oct 2015 15:39:29
Server: Apache/1.3.0 (Unix)
Last-Modified: Wed, 9 Sep 2015 09:23:24
Content-Type: image/gif

(data data data data data ...)
```

Conditional GET

The cache forwards the object to the requesting browser but also caches the object locally.
The cache also stores the last-modified date along with the object
Another browser requests the same object via the cache.
The object is still in the cache, since this object may have been modified at the Web server in the past week, the cache performs an up-to-date check by issuing a conditional GET the cache sends:

```
GET /fruit/kiwi.gif HTTP/1.1
Host: www.exotiquecuisine.com
If-modified-since: Wed, 9 Sep 2015 09:23:24
```

Conditional GET

If-modified-since: header line is exactly equal to the value of the Last-Modified: header line that was sent by the server one week ago. This conditional GET is telling the server to send the object only if the object has been modified since the specified date.

Suppose the object has not been modified since 9 Sep 2015 09:23:24. Then, fourth, the Web server sends a response message to the cache

```
HTTP/1.1 304 Not Modified
Date: Sat, 10 Oct 2015 15:39:29
Server: Apache/1.3.0 (Unix)

(empty entity body)
```

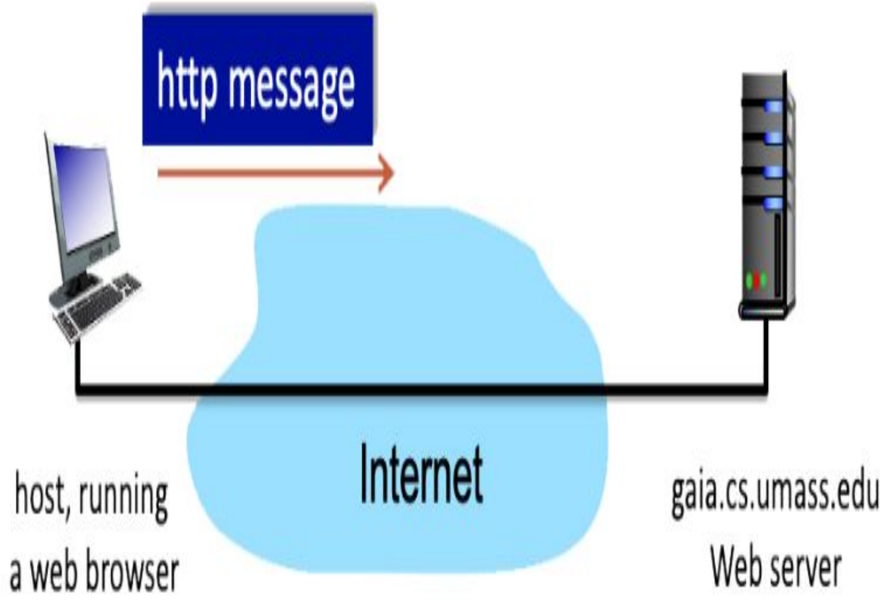
304 Not Modified in the status line, tells the cache that it can go ahead and forward its (the proxy cache’s) cached copy of the object to the requesting browser.

HTTP GET Message- Numerical 2

Consider the figure below, where a client is sending an HTTP GET message to a web server, gaia.cs.umass.edu

Suppose the client-to-server HTTP GET message is the following:

GET /kurose_ross_sandbox/interactive/quotation5.htm HTTP/1.1
Host: gaia.cs.umass.edu
Accept: text/plain, text/html, image/png, image/gif, audio/vnf.wave, audio/mpeg, video/mp4, video/mpeg,
Accept-Language: en-us, en-gb;q=0.4, en;q=0.6, fr, fr-ch, de, ar, cs
If-Modified-Since: Sun, 31 Jul 2022 07:45:04 -0700
User Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)



HTTP GET Message- Numerical 2

1. What is the name of the file that is being retrieved in this GET message?
2. What version of HTTP is the client running?
3. True or False: The client will accept html files
4. True or False: The client will accept jpeg images
5. What is the client's preferred version of English?
6. What is the client's least preferred version of English?
7. True or False: The client will accept the German language
8. True or False: The client already has a cached copy of the file

Solution

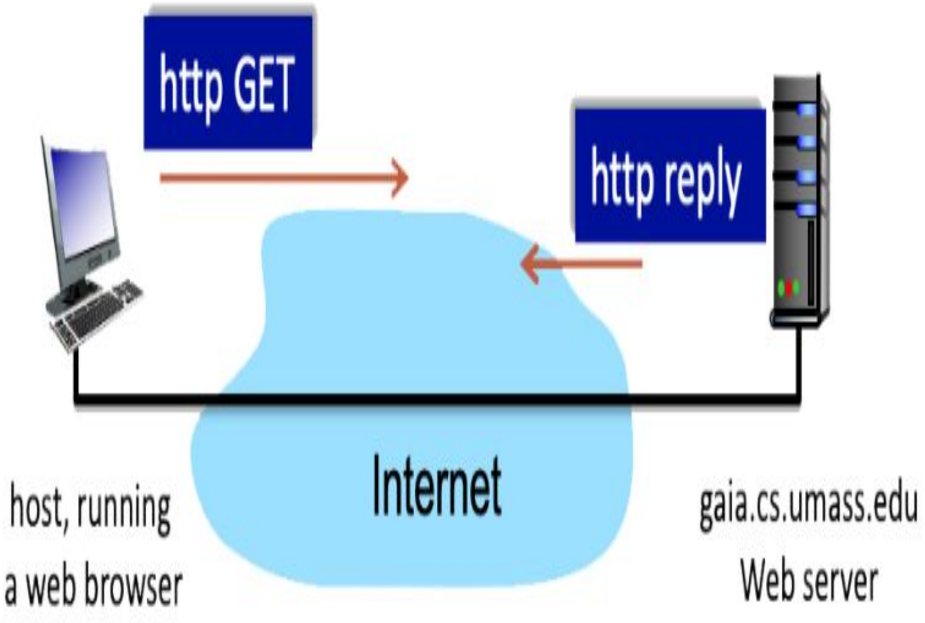
1. The name of the file is quotation5.htm.
2. The client is running on HTTP/1.1
3. True. In the 'Accept' field the client includes 'text/html' files.
4. False. The client does NOT include 'image/jpeg' in its 'Accept' field.
5. The client's preferred version of English is American English. Any language without a defined q value has a default value of 1
6. The client's least preferred version of English is British English because it has the lowest q value.
7. True. The client does include German in its 'Accepted-Language' field.
8. True. The client has a cached copy of the file that was updated on: Sun, 31 Jul 2022 07:45:04 -0700

HTTP Response Message- Numerical 3

Consider the figure below, where the server is sending a HTTP RESPONSE message back the client.

Suppose the server-to-client HTTP RESPONSE message is the following:

HTTP/1.1 404 Not Found
Date: Sun, 31 Jul 2022 14:49:56 +0000
Server: Apache/2.2.3 (CentOS)
Content-Length: 65979
Keep-Alive: timeout=43, max=51
Connection: Keep-alive
Content-type: text/html



HTTP Response Message- Numerical 3

1. Is the response message using HTTP 1.0 or HTTP 1.1?

2. Was the server able to send the document successfully? Yes or No

3. How big is the document in bytes?

4. Is the connection persistent or nonpersistent?

5. What is the type of file being sent by the server in response?

6. What is the name of the server and its version? Write your answer as server/x.y.z

7. Will the ETag change if the resource content at this particular resource location changes? Yes or No

Solution

1. The response is using HTTP/1.1
2. Since the response code is 404 Not Found, the document was NOT received successfully.
3. The document is 65979 bytes.
4. The connection is persistent.
5. The file type the server is sending is text/html.
6. The name and version of the server is Apache/2.2.3
7. Yes. The Etag is a string that uniquely identifies a resource. If a resource is updated, the Etag will change.

COMPUTER COMMUNICATION NETWORKS

Kurose links

https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/content/interactiveanimations/http-delay-estimation/index.html

http://gaia.cs.umass.edu/kurose_ross/interactive/http-get.php

PES
UNIVERSITY
ONLINE

THANK YOU

Prajeesha

Department of Electronics and Communication Engineering