SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107
Accredited by NAAC-UGC with ‘A’ Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Information Technology

Course Name – 19IT401 Computer Networks

II Year / IV Semester

Unit 5 – Application Layer

Topic 2 –HTTP

HTTP

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

- It is a request / response protocol. 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. It uses services of TCP.

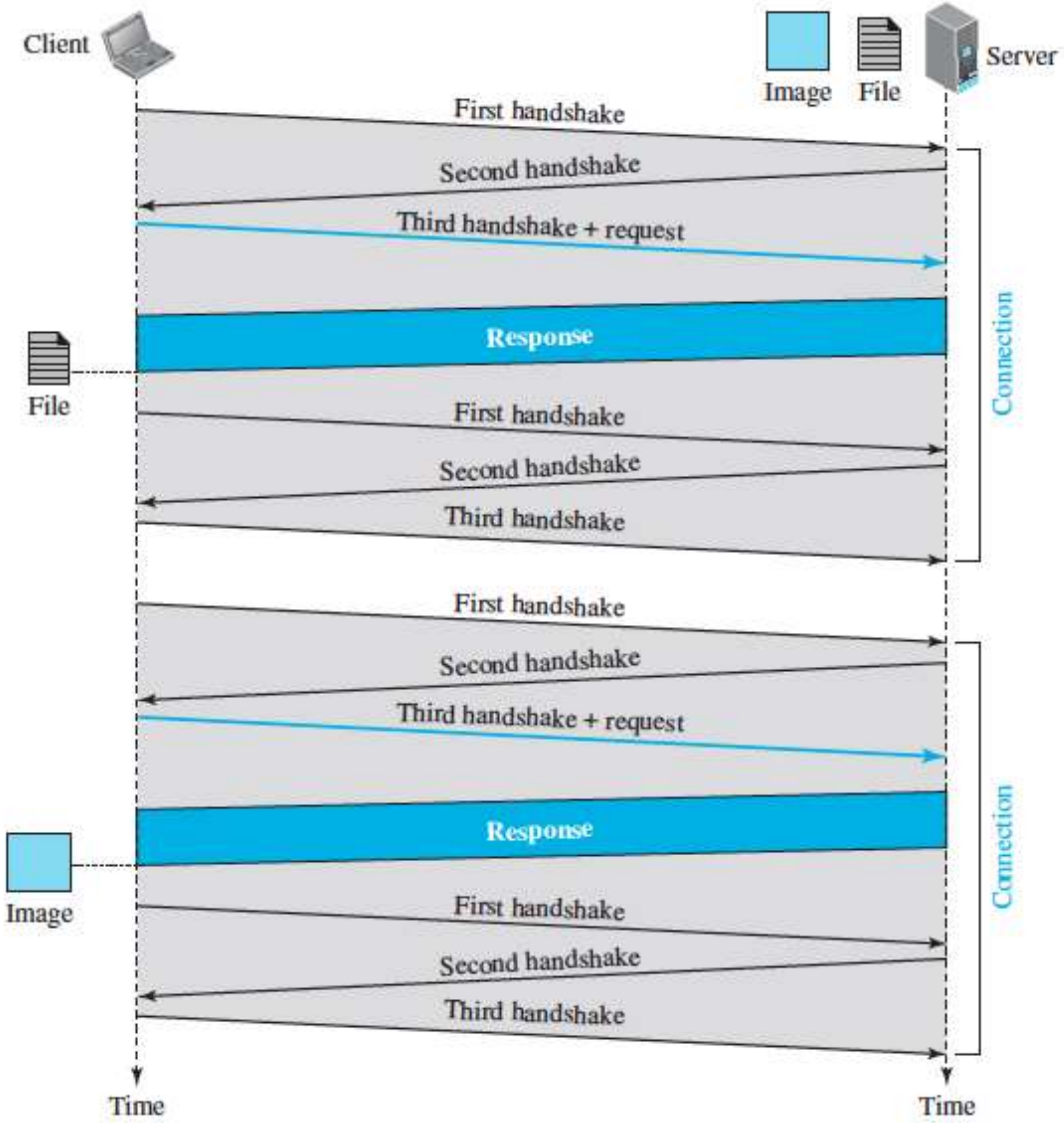
- Two Types of Connection - **Nonpersistent versus Persistent Connections**

Nonpersistent Connections

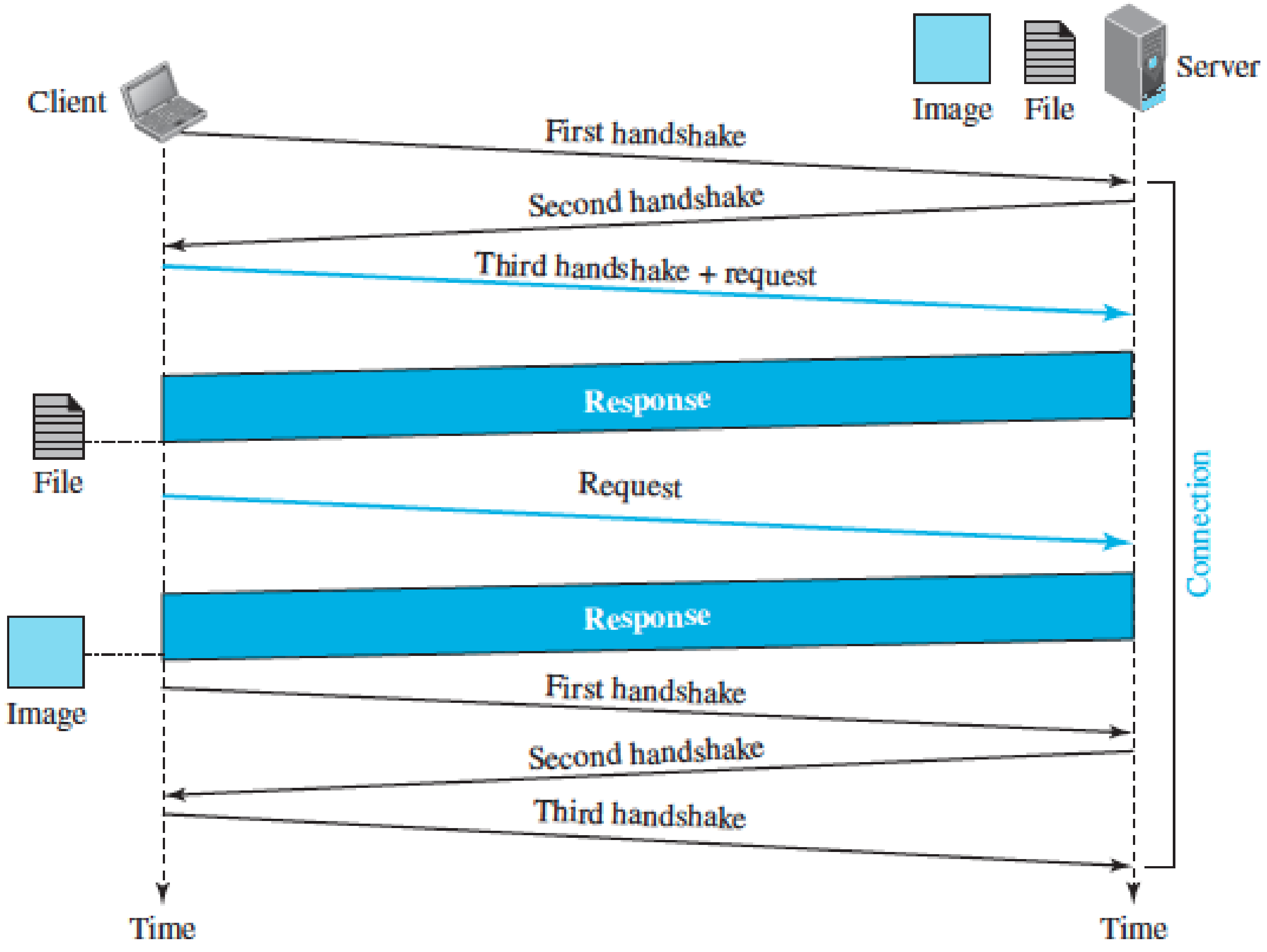
- In a nonpersistent connection, one TCP connection is made for each request/response.
- The following lists the steps in this strategy:
  1. The client opens a TCP connection and sends a request.
  2. The server sends the response and closes the connection.
  3. The client reads the data until it encounters an end-of-file marker; it then closes the connection.

HTTP

Nonpersistent Connection



Persistent Connection



HTTP

Persistent Connections

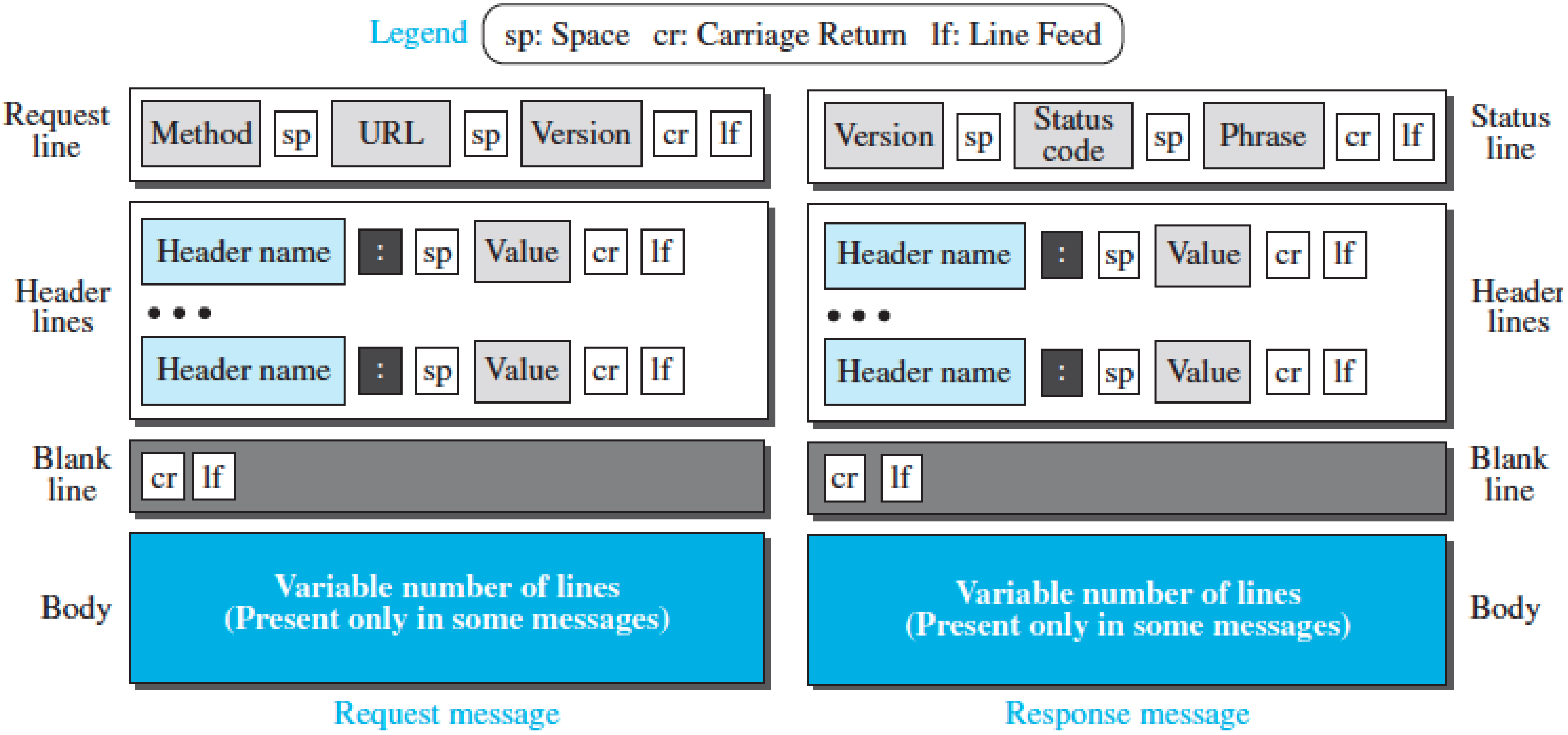
- HTTP version 1.1 specifies a persistent connection by default.
- In a persistent connection, the server leaves the connection open for more requests after sending a response.
- The server can close the connection at the request of a client or if a time-out has been reached.

Message Format

- The HTTP protocol defines the format of the **request and response messages**.
- Each message is made of four sections.
- 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 other three sections have the same names in the request and response messages. They are **Header Line, Blank line and Body.**

HTTP

Message Format


Legend
sp: Space cr: Carriage Return lf: Line Feed
Request line
Method sp URL sp Version cr lf
Header lines
Header name : sp Value cr lf
• • •
Header name : sp Value cr lf
Blank line
cr lf
Body
Variable number of lines (Present only in some messages)
Request message
Status line
Version sp Status code sp Phrase cr lf
Header lines
Header name : sp Value cr lf
• • •
Header name : sp Value cr lf
Blank line
cr lf
Body
Variable number of lines (Present only in some messages)
Response message

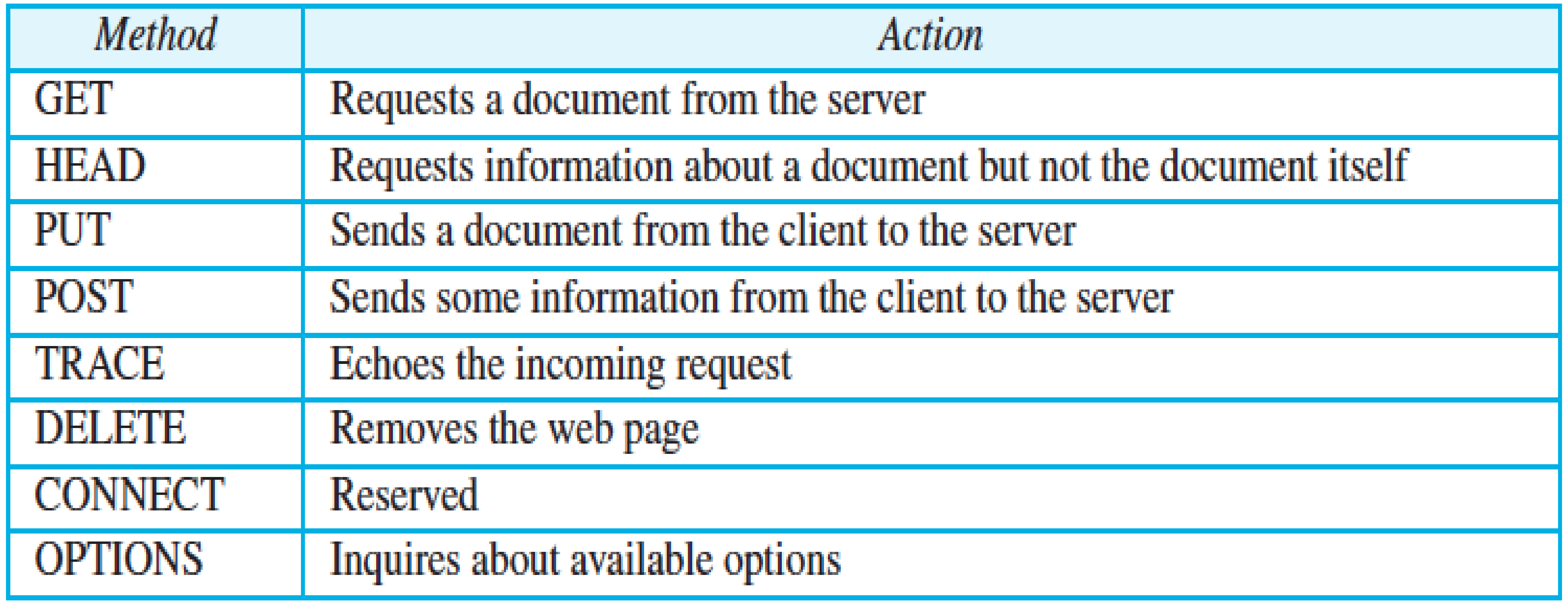
HTTP

Request Message

- There are three fields in request line. The fields are called **method, URL , and version**
- The method field defines **the request types**. In version 1.1 of HTTP, several methods are defined.
- Most of the time, the client uses the GET method to send a request

- The second field, **URL**, 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**.

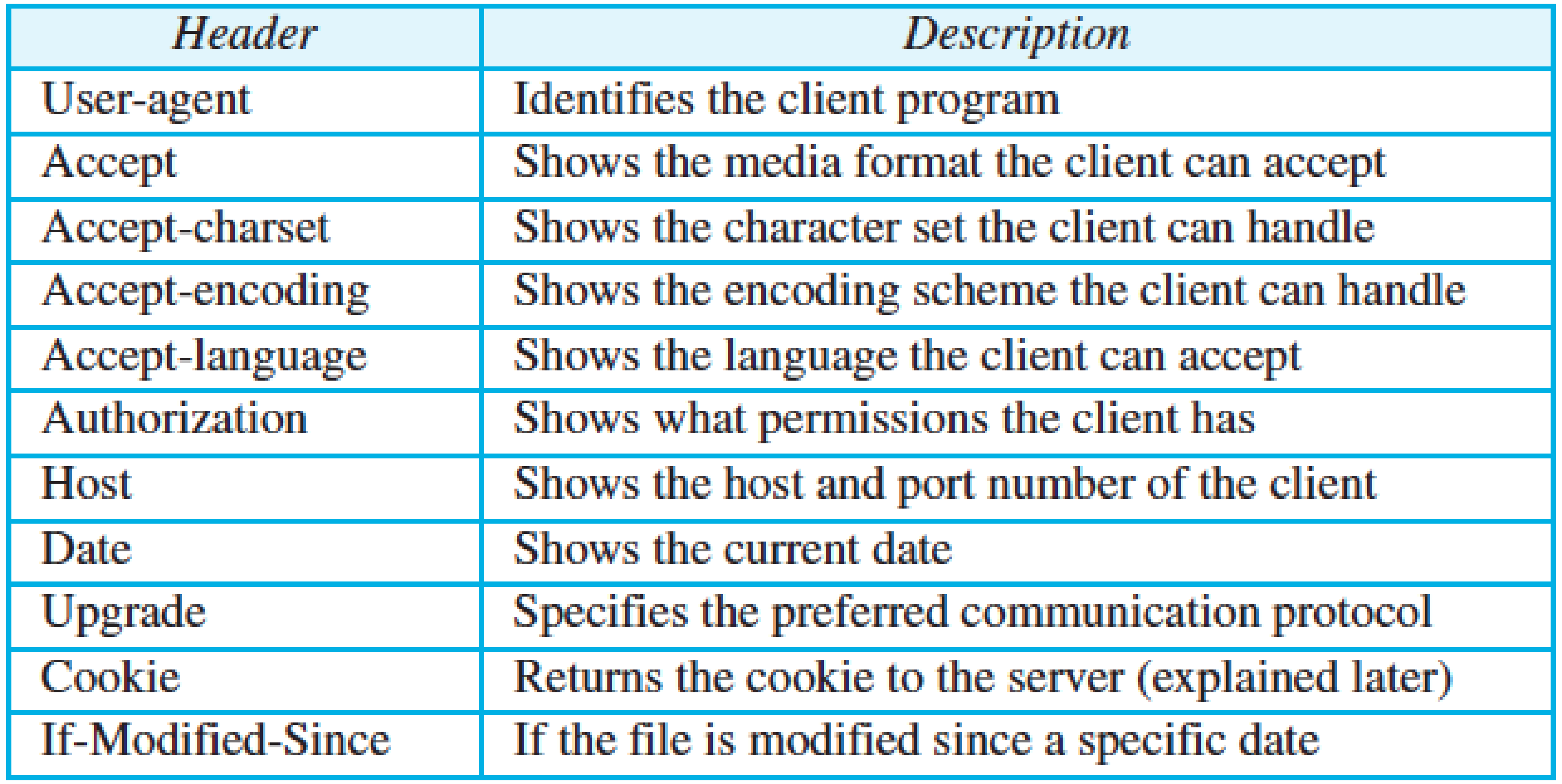


Method	*Action*
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
PUT	Sends a document from the client to the server
POST	Sends some information from the client to the server
TRACE	Echoes the incoming request
DELETE	Removes the web page
CONNECT	Reserved
OPTIONS	Inquires about available options

HTTP

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.
- Each header line has a header name, a colon, a space, and a header value.
- The **body** can be present in a request message. Usually, it contains the comment to be sent or the file to be published on the website when the method is PUT or POST.



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

HTTP

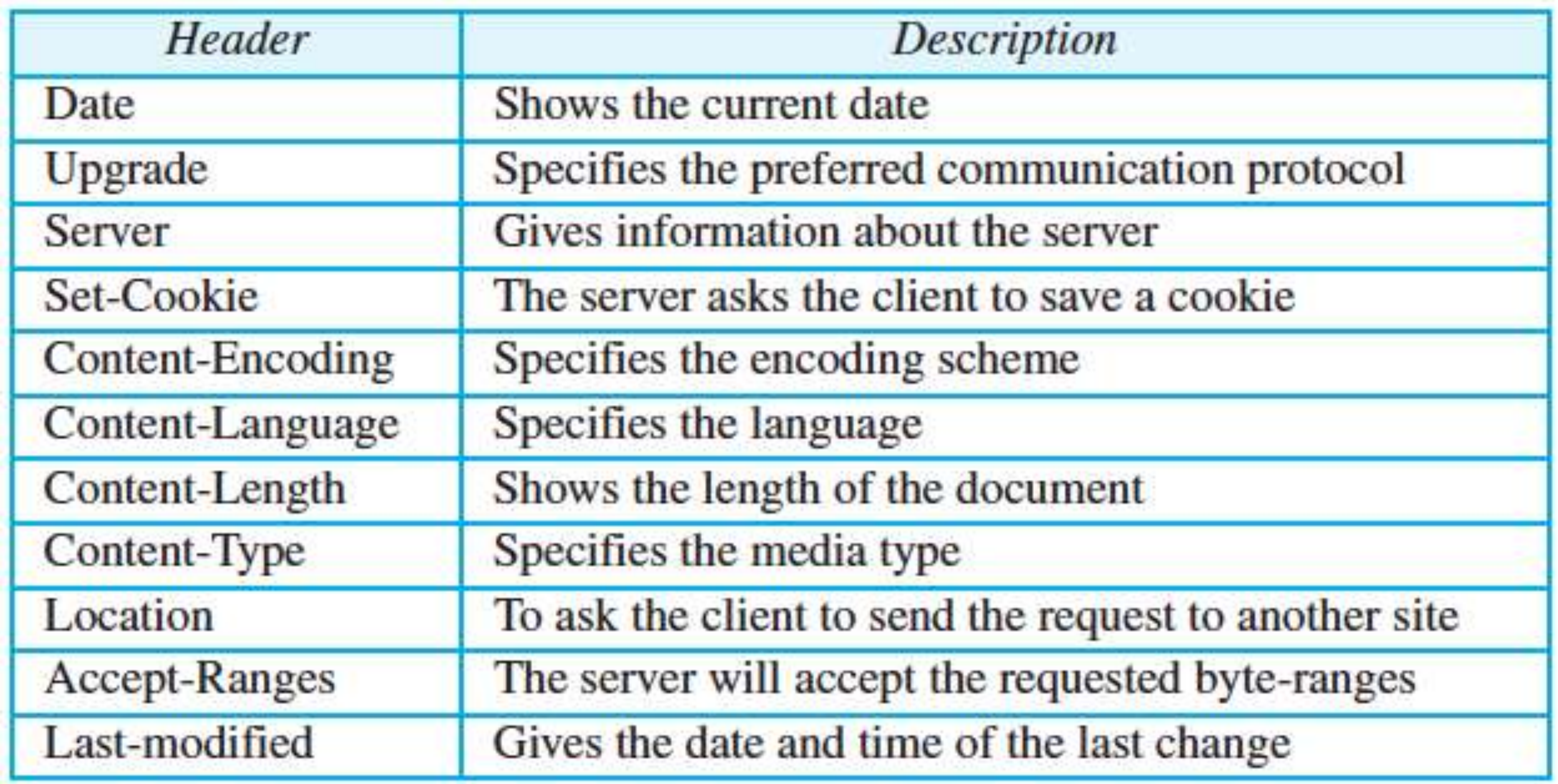
Response Message

- 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. They are **Version, Status code and Phrase**
- 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.
- Codes in the 100 range are only informational
- Codes in the 200 range indicate a successful request.
- Codes in the 300 range redirect the client to another URL.
- Codes in the 400 range indicate an error at the client site.
- Codes in the 500 range indicate an error at the server site.
- The **status phrase** explains the status code in text form.

HTTP

Response Message

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

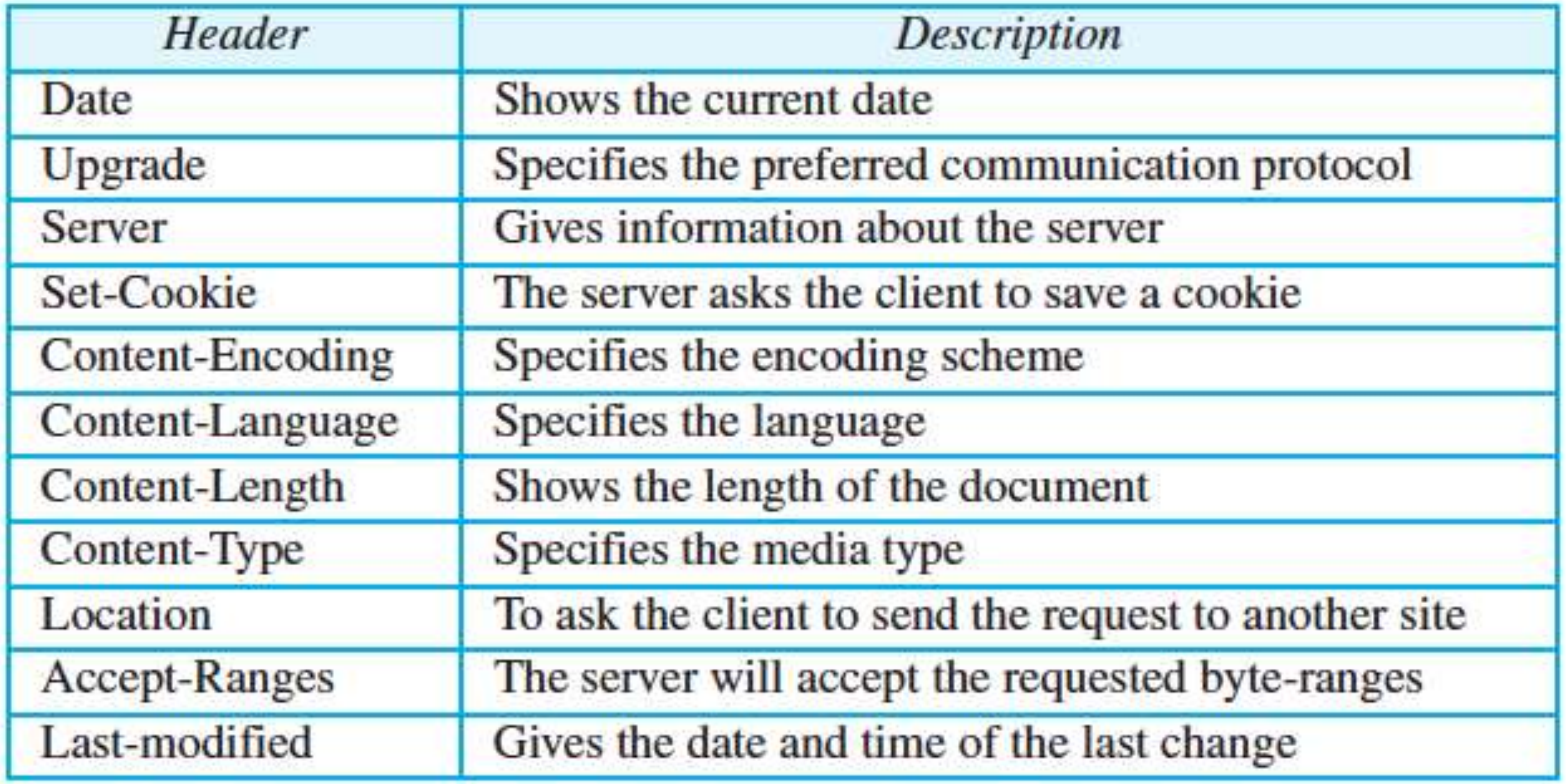


Header	*Description*
Date	Shows the current date
Upgrade	Specifies the preferred communication protocol
Server	Gives information about the server
Set-Cookie	The server asks the client to save a cookie
Content-Encoding	Specifies the encoding scheme
Content-Language	Specifies the language
Content-Length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

HTTP

Response Message

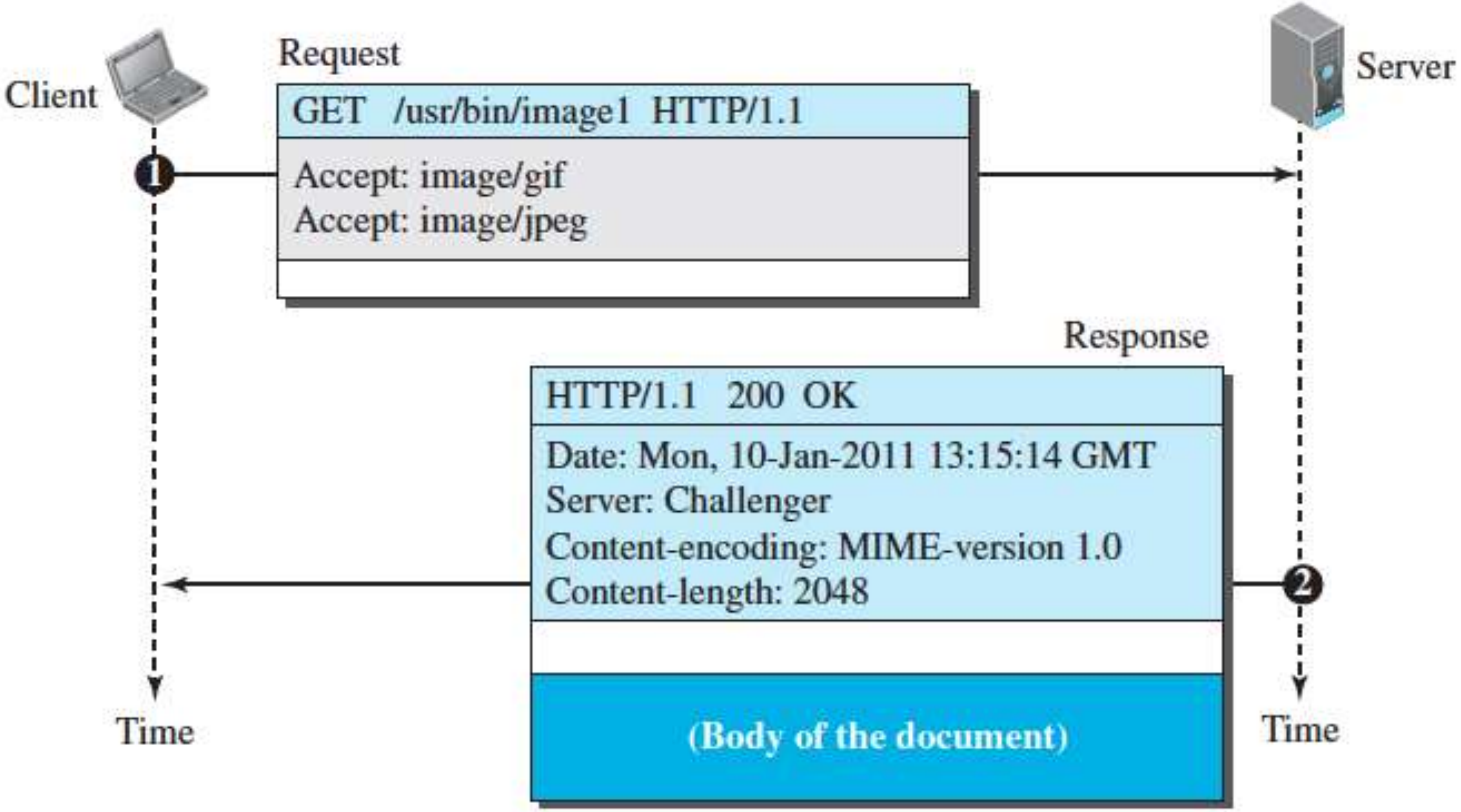
- After the status line, we can have zero or more **response header lines**. Each headerline sends additional information from the server to the client
- Each header line has a header name, a colon, a space, and a header value.
- The **body** contains the document to be sent from the server to the client. The body is present unless the response is an error message.



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

HTTP

Example : This example retrieves a document

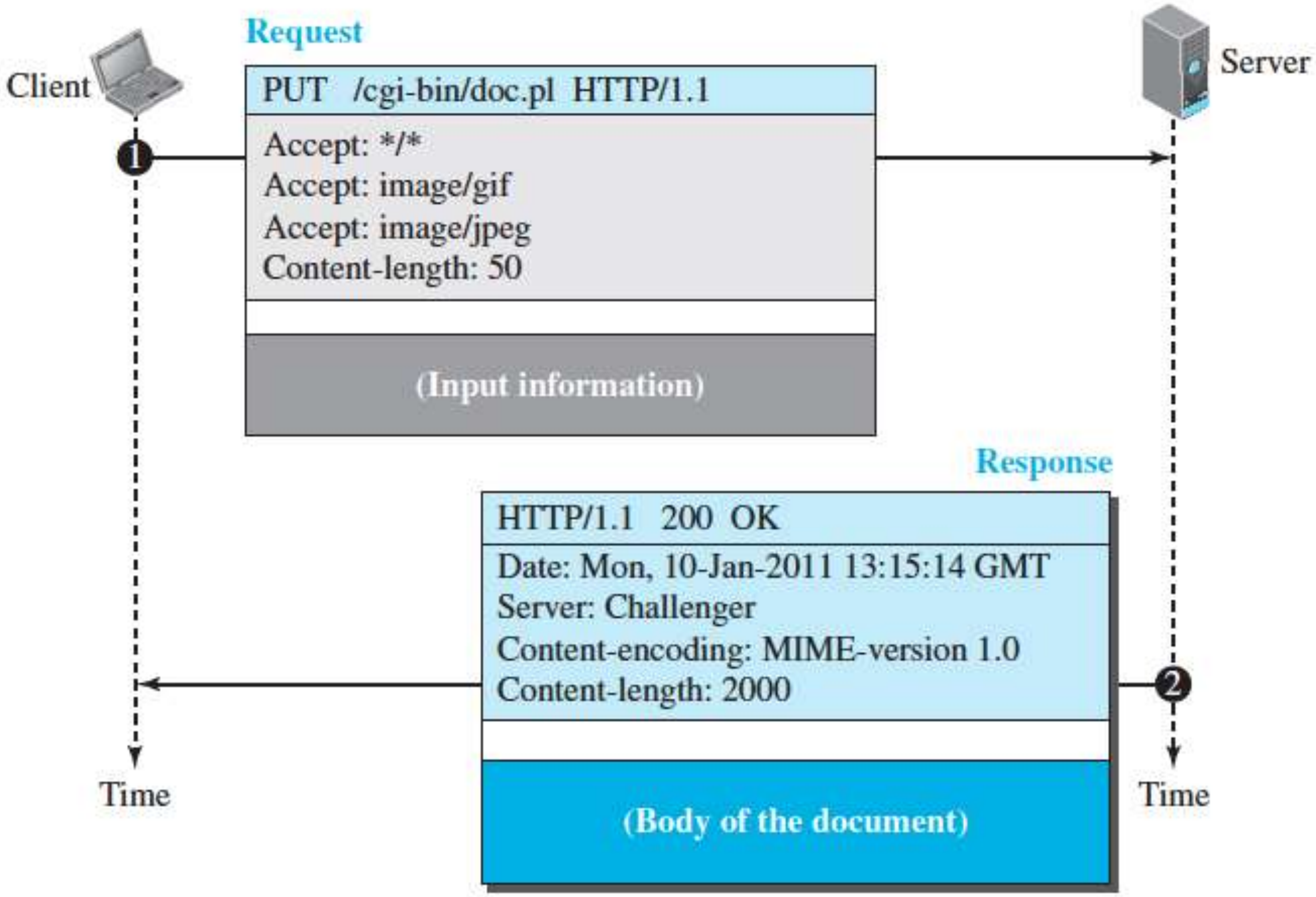


```
GET   /usr/bin/image1  HTTP/1.1
Accept: image/gif
Accept: image/jpeg
```

```
HTTP/1.1   200  OK
Date: Mon, 10-Jan-2011 13:15:14 GMT
Server: Challenger
Content-encoding: MIME-version 1.0
Content-length: 2048

(Body of the document)
```

HTTP

Example : client wants to send a web page to be posted on the server



```
PUT   /cgi-bin/doc.pl  HTTP/1.1
Accept: */*
Accept: image/gif
Accept: image/jpeg
Content-length: 50

(Input information)
```

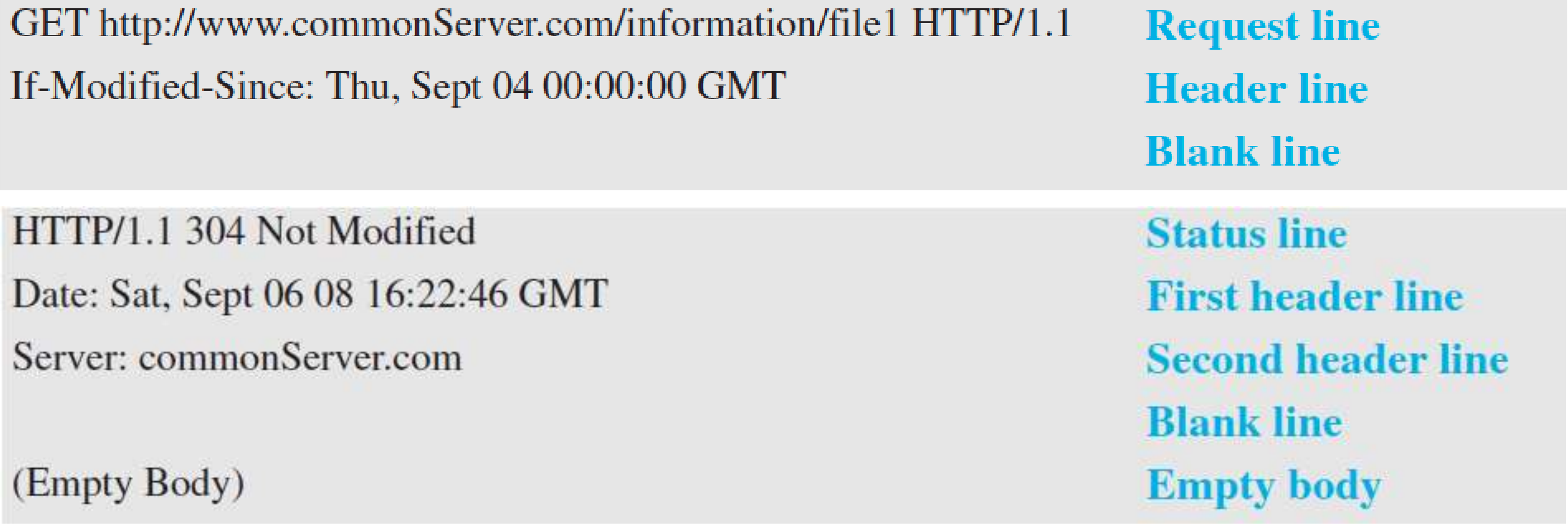
```
HTTP/1.1   200  OK
Date: Mon, 10-Jan-2011 13:15:14 GMT
Server: Challenger
Content-encoding: MIME-version 1.0
Content-length: 2000

(Body of the document)
```

HTTP

Conditional Request

- A client can add a condition in its request. In this case, the server will send the requested web page if the condition is met or inform the client otherwise.
- One of the most common conditions imposed by the client is the time and date the web page is modified.
- The client can send the header line If-Modified-Since with the request to tell the server that it needs the page only if it is modified after a certain point in time.

Example



```
GET http://www.commonServer.com/information/file1 HTTP/1.1
If-Modified-Since: Thu, Sept 04 00:00:00 GMT

HTTP/1.1 304 Not Modified
Date: Sat, Sept 06 08 16:22:46 GMT
Server: commonServer.com

(Empty Body)
```

Request line
Header line
Blank line
Status line
First header line
Second header line
Blank line
Empty body

HTTP

Cookies

- An HTTP cookie (web cookie, browser cookie) is a small piece of data that a server sends to a user's web browser. Web has functions that need to remember some information about the clients.
- The browser may store the cookie and send it back to the same server with later requests.
- Typically, an HTTP cookie is used to tell if two requests come from the same browser—keeping a user logged in, for example.
- It remembers stateful information for the stateless HTTP protocol.

Cookies are mainly used for three purposes:

- **Session management -** Logins, shopping carts, game scores, or anything else the server should remember
- **Personalization -** User preferences, themes, and other settings
- **Tracking** - Recording and analyzing user behavior

HTTP

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. However, to use the proxy server, the client must be configured to access the proxy instead of the target server.

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, that stores 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.

THANK YOU