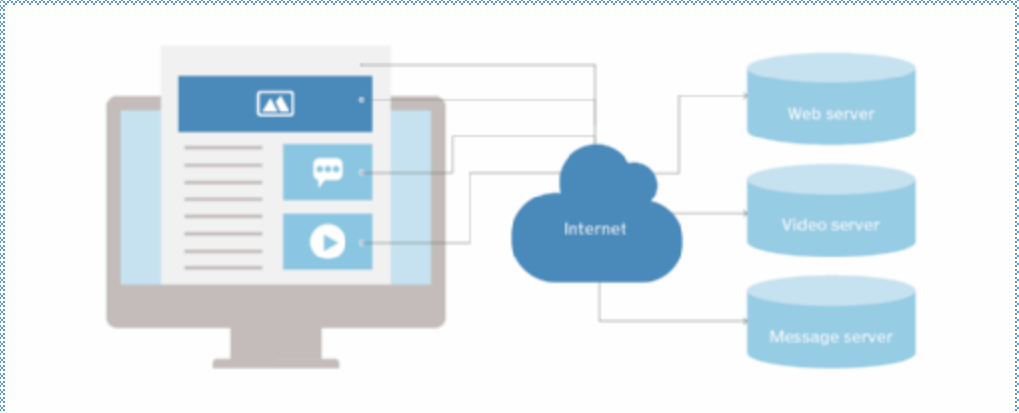
HyperText Transfer Protocol (HTTP)

- Meena M
CSE-A

What is HTTP and Protocol ?



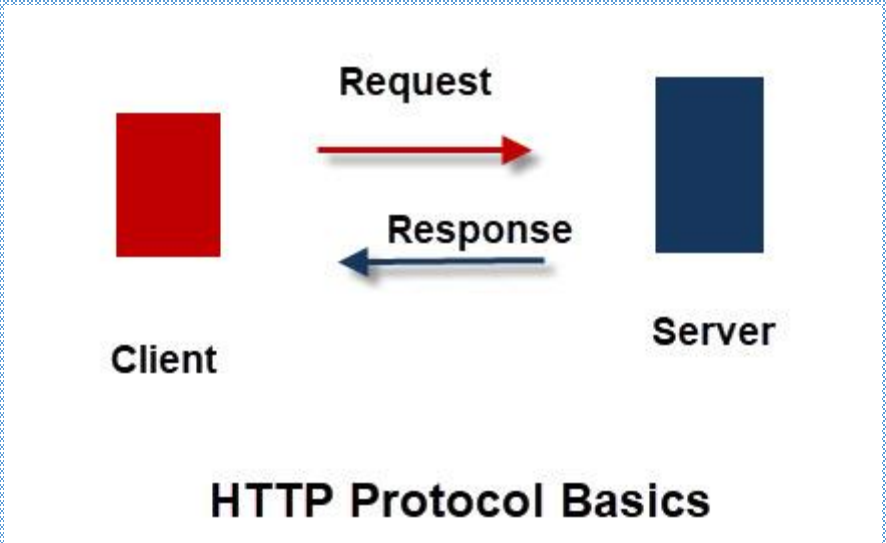
What is HTTP and Protocol ?

Protocol : It is a standard procedure for defining and regulating communication. Examples- TCP,UDP,HTTP etc.

HTTP is the foundation of data communication for the World Wide Web.

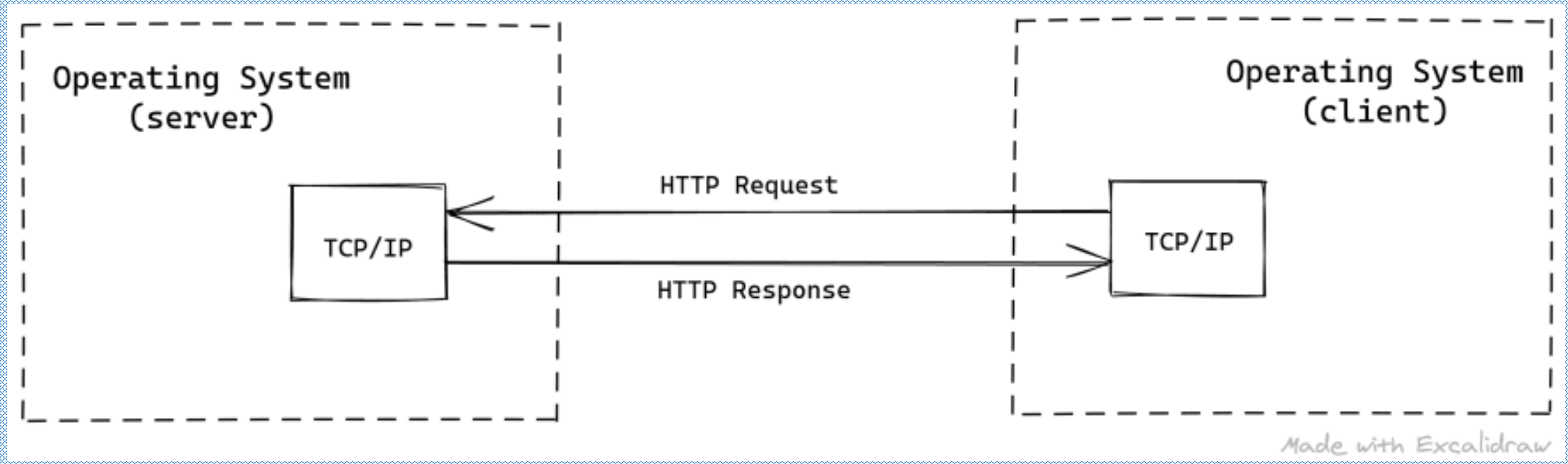
The HTTP is the Web's application layer protocol for transferring various forms of data between server and client like plaintext, hypertext, image, videos and sounds.

INTRODUCTION

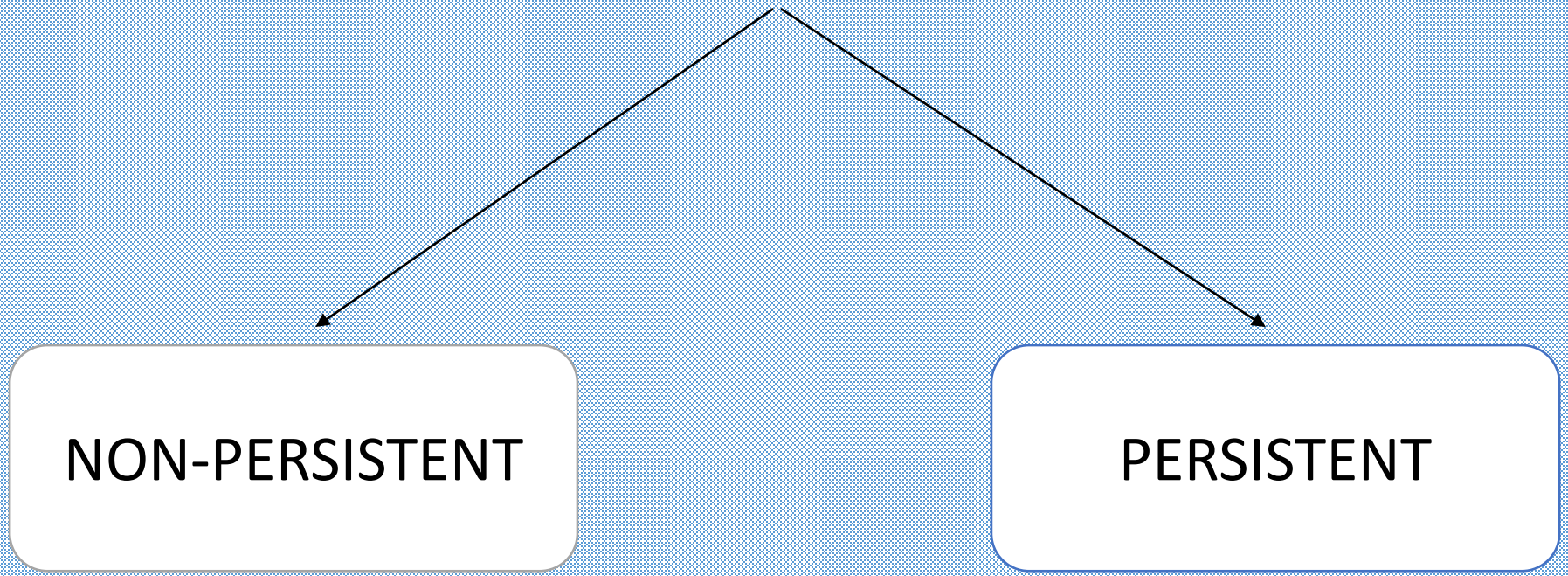


INTRODUCTION

- HTTP defines 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.
- HTTP uses the services of TCP.


Operating System
(server)
TCP/IP
HTTP Request
HTTP Response
Operating System
(client)
TCP/IP
Made with Excalidraw

CONNECTIONS

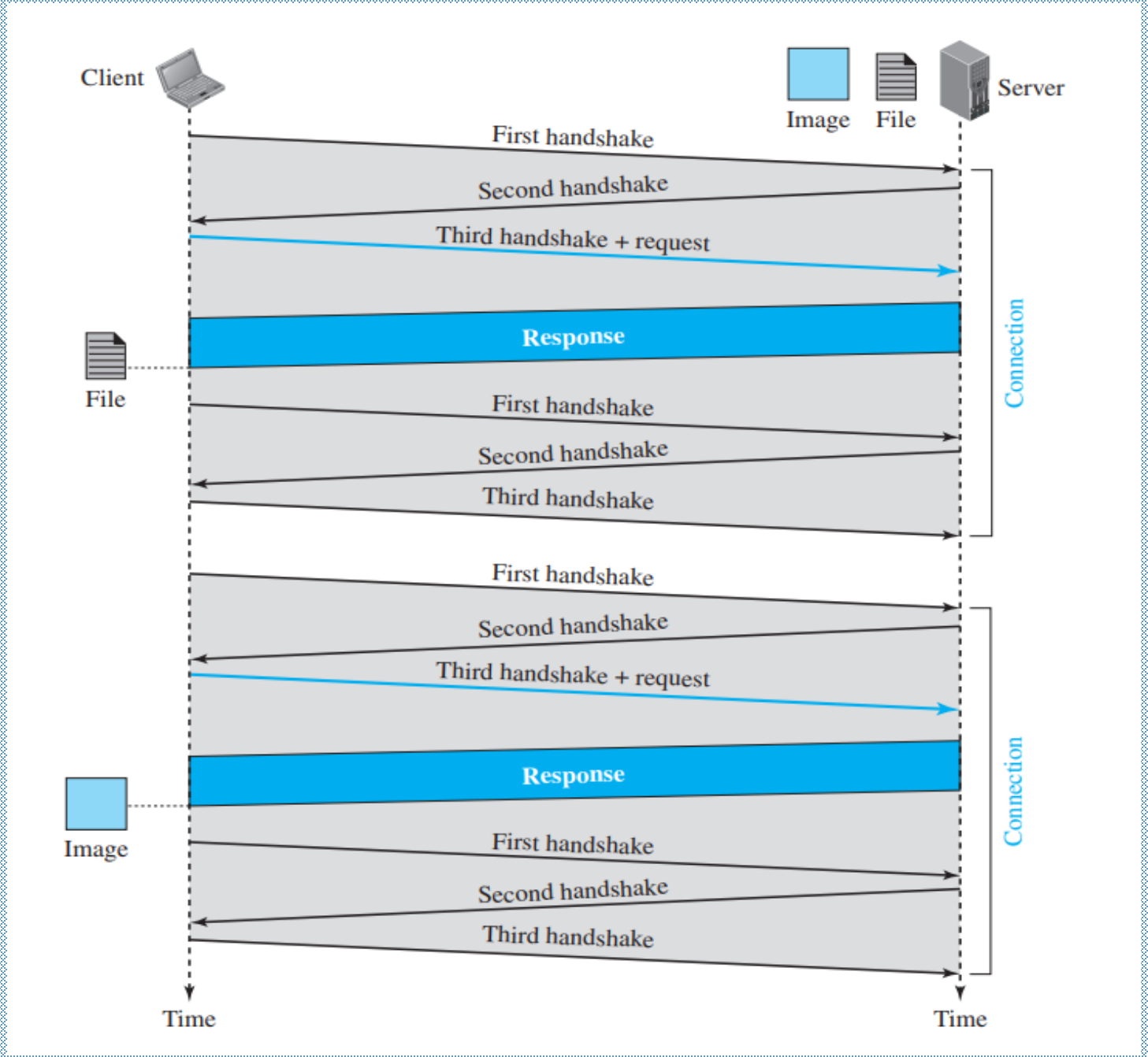


NON-PERSISTENT CONNECTION

- In a non-persistent connection, one TCP connection is made for each request/response. The following lists the steps in this strategy:

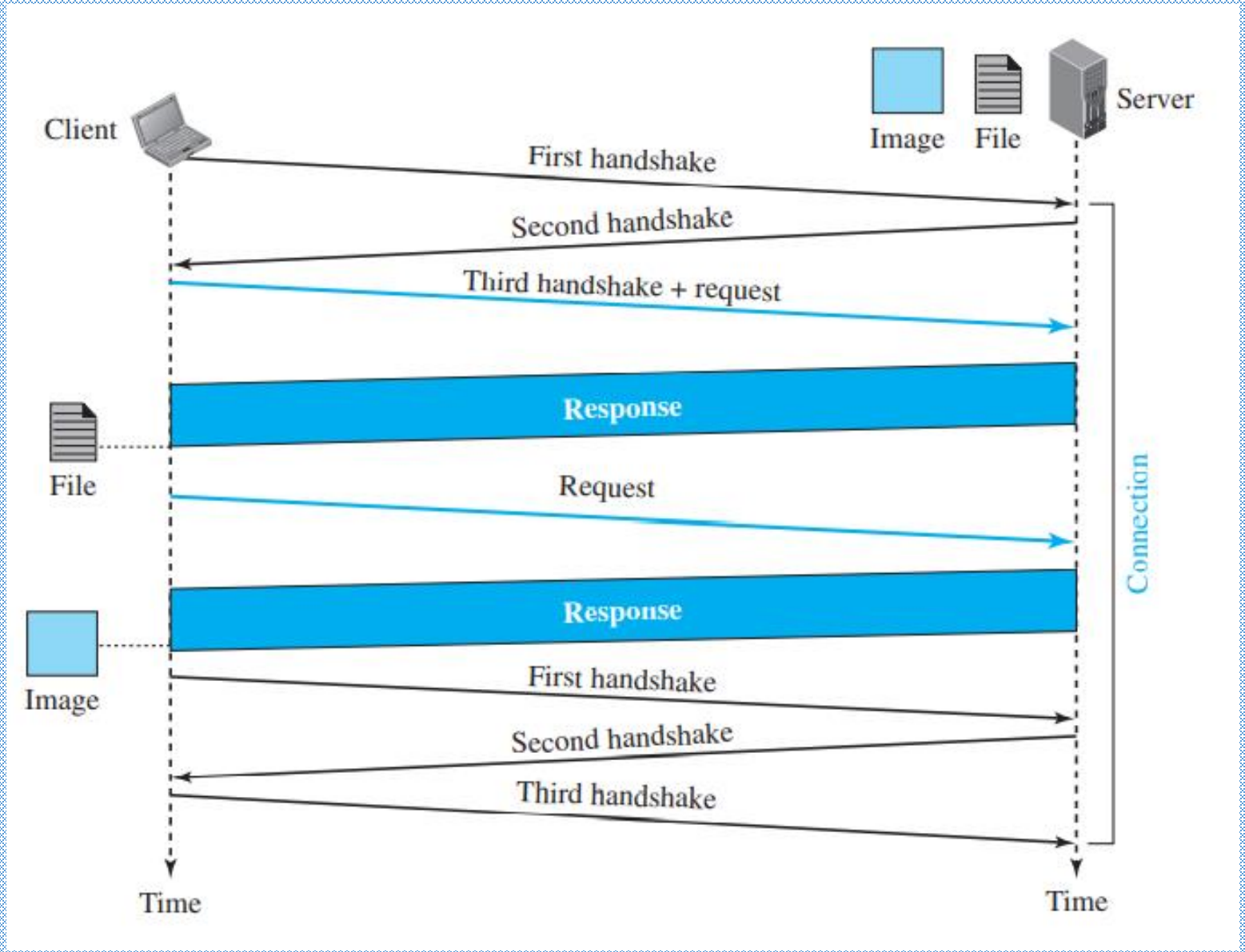
  1. The client opens a TCP connection and sends a request.
  2. The server sends the response and closes the connection.
  3. The client reads the data until it encounters an end-of-file marker; it then closes the connection.

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

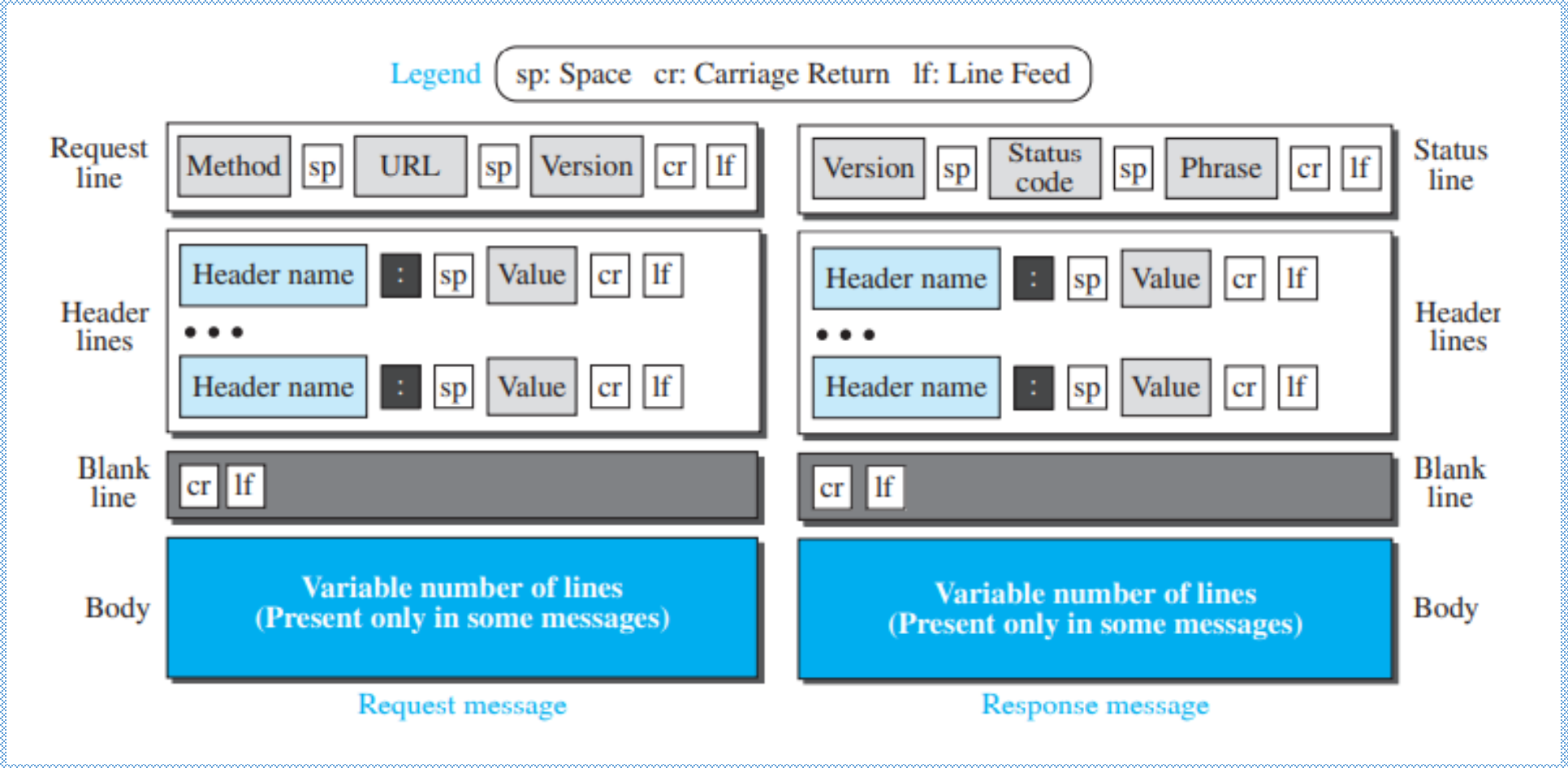

Client
Image
File
Server
First handshake
Second handshake
Third handshake + request
Response
File
First handshake
Second handshake
Third handshake
Connection
First handshake
Second handshake
Third handshake + request
Response
Image
First handshake
Second handshake
Third handshake
Connection
Time
Time

PERSISTENT CONNECTION

- HTTP version 1.1
- 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.
- Only one set of buffers and variables needs to be set for the connection at each site.


Client
Image
File
Server
First handshake
Second handshake
Third handshake + request
Response
File
Request
Response
Image
First handshake
Second handshake
Third handshake
Connection
Time
Time

MESSAGE FORMAT



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 and terminated by two characters (carriage return and line feed).
- The fields are called method, URL, and version.
- Methods:

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

- 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.
- 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 .
- The value field defines the values associated with each header name. The list of values can be found in the corresponding RFCs.
- 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 names commonly used are:

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

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.

  100 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,

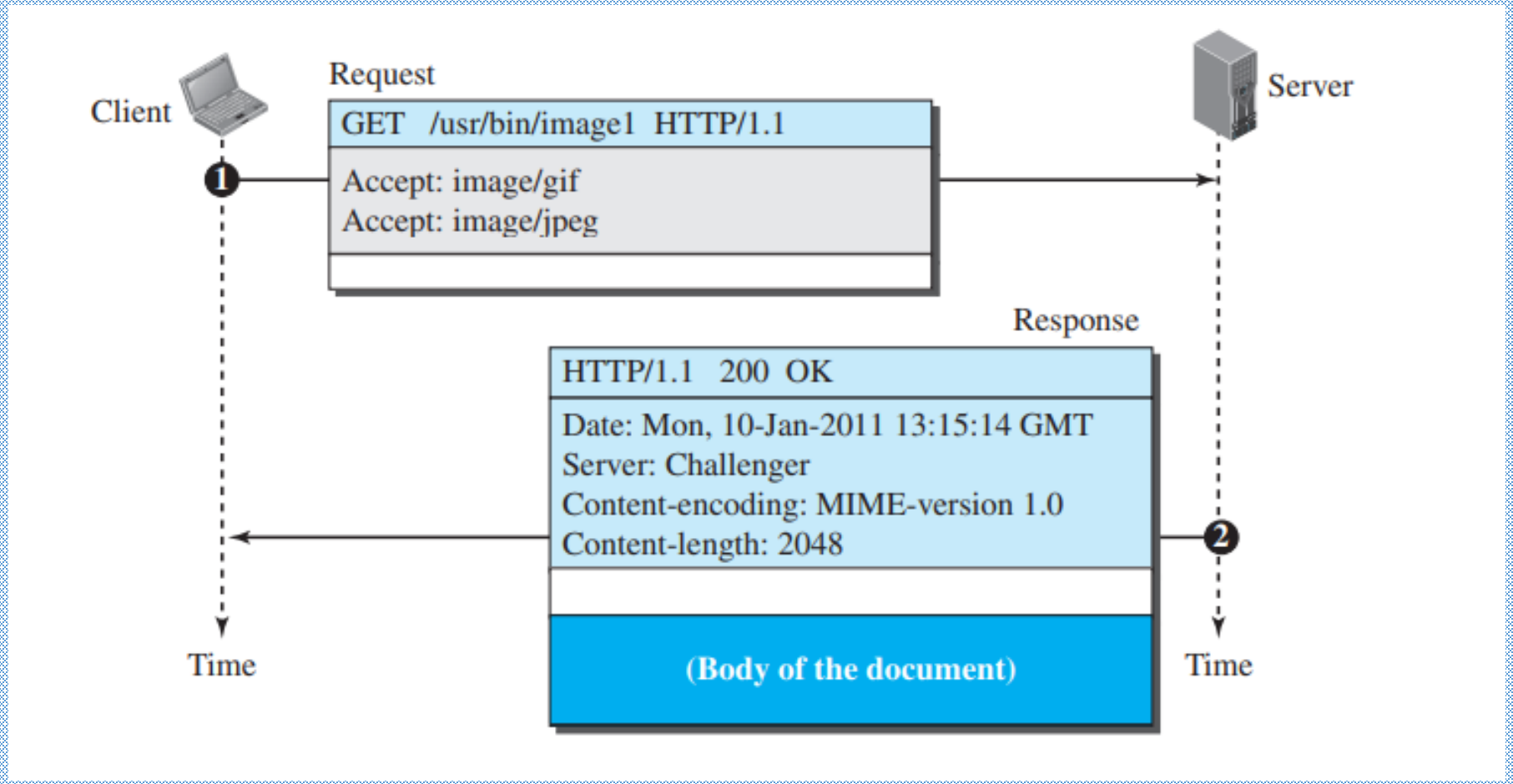
  and the codes in the 400 range indicate an error at the client site.

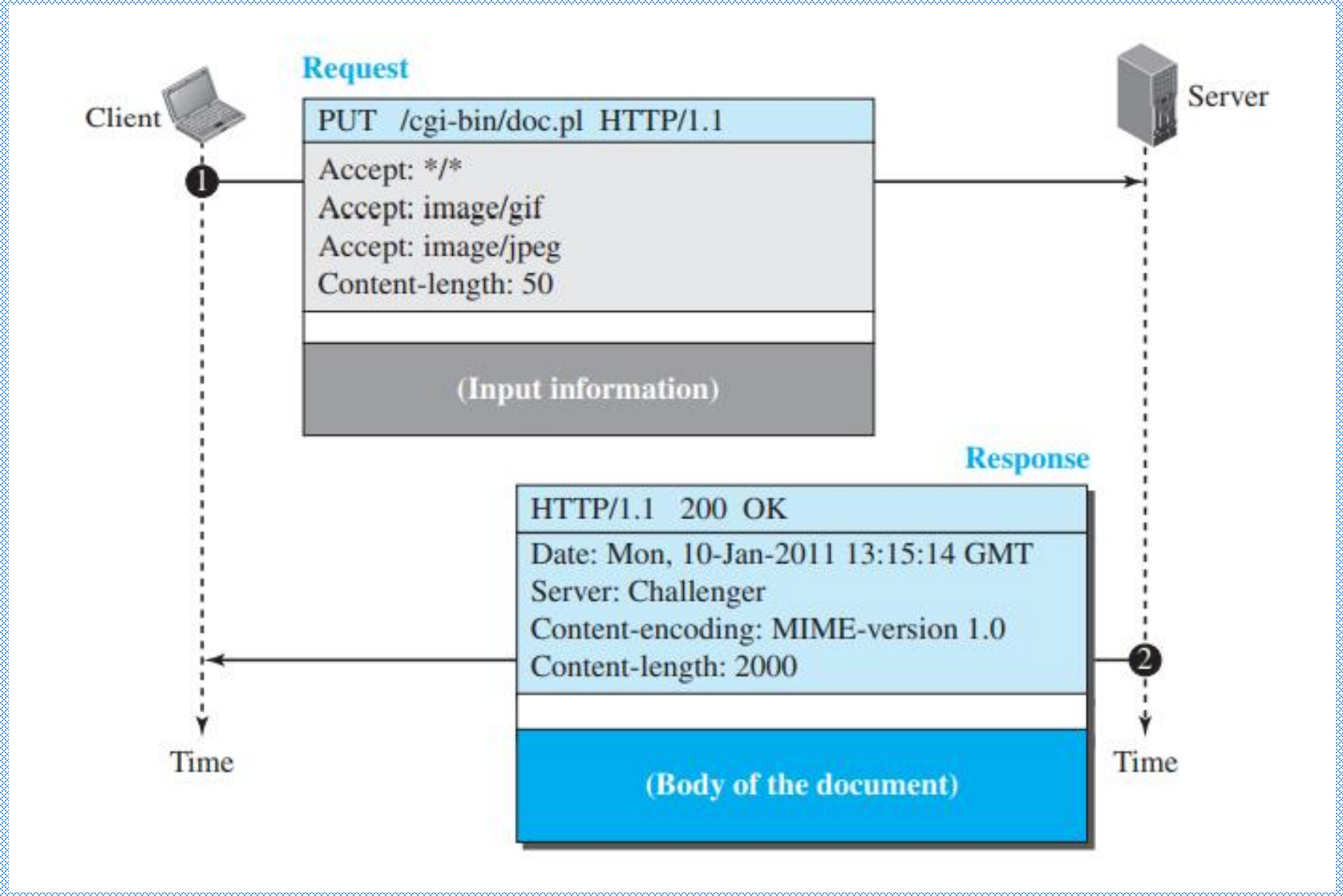
  Finally, the codes in the 500 range indicate an error at the server site.
- The status phrase explains the status code in text form. After the status line, we can have zero or more response header lines.
- Each header line sends additional information from the server to the client. For example, the sender can send extra information about the document.
- Each header line has a header name, a colon, a space, and a header value.

- Header names commonly used in a response message are as follows:

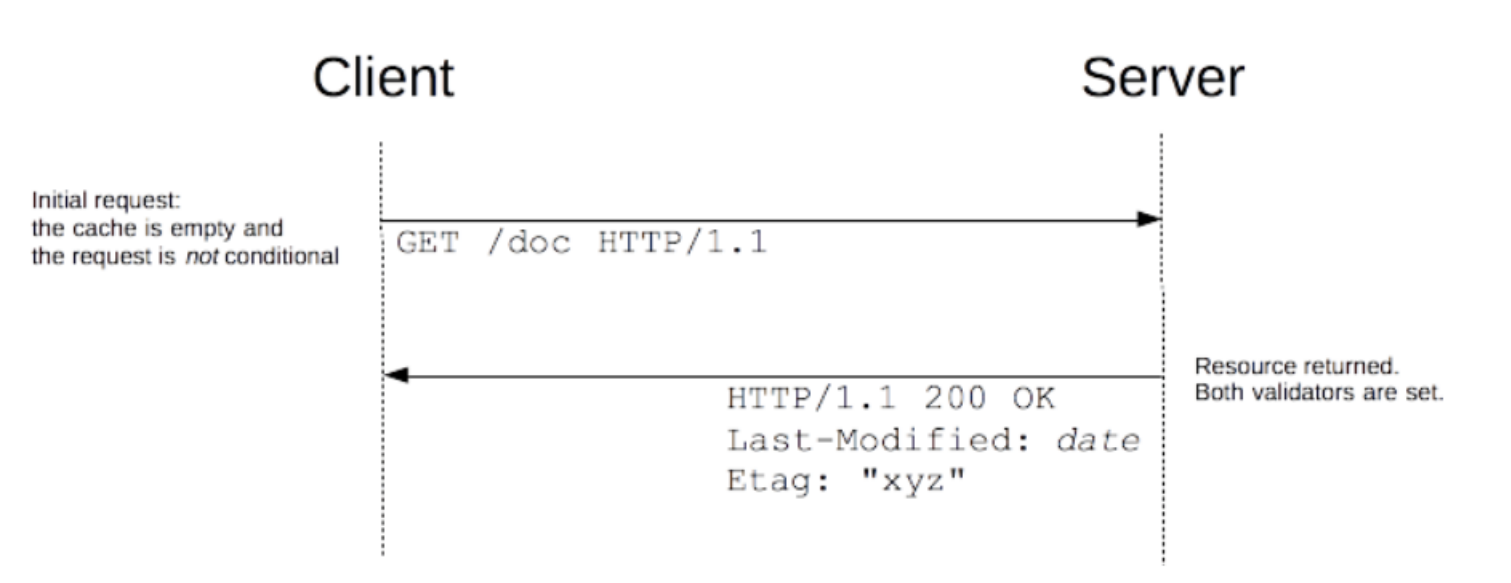
Header	*Description*
Date	Shows the current date
Upgrade	Specifies the preferred communication protocol
Server	Gives information about the server
Set-Cookie	The server asks the client to save a cookie
Content-Encoding	Specifies the encoding scheme
Content-Language	Specifies the language
Content-Length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

- 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.


Client
Request
GET /usr/bin/image1 HTTP/1.1
Accept: image/gif
Accept: image/jpeg
Server
1
Response
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)
2
Time
Time


Request
Client
Server
PUT /cgi-bin/doc.pl HTTP/1.1
Accept: */*
Accept: image/gif
Accept: image/jpeg
Content-length: 50
(Input information)
Response
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)
1
2
Time
Time

CONDITIONAL REQUEST



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.

The following shows how a client imposes the modification data and time condition on a request.

GET http://www.commonServer.com/information/file1 HTTP/1.1	**Request line**
If-Modified-Since: Thu, Sept 04 00:00:00 GMT	**Header line**
	Blank line

The status line in the response shows the file was not modified after the defined point in time. The body of the response message is also empty.

HTTP/1.1 304 Not Modified	**Status line**
Date: Sat, Sept 06 08 16:22:46 GMT	**First header line**
Server: commonServer.com	**Second header line**
	Blank line
(Empty Body)	**Empty body**

FUNCTIONS OF WEB

Functions of Web:

- 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 agencies.

For these purposes ….. We require?

COOKIES

COOKIES

- Creating and Storing Cookies
- Using Cookies
- Web Caching: Proxy Servers
- Cache Update

CREATING AND STORING COOKIES

- The creation and storing of cookies depend on the implementation; however, the principle is the same.

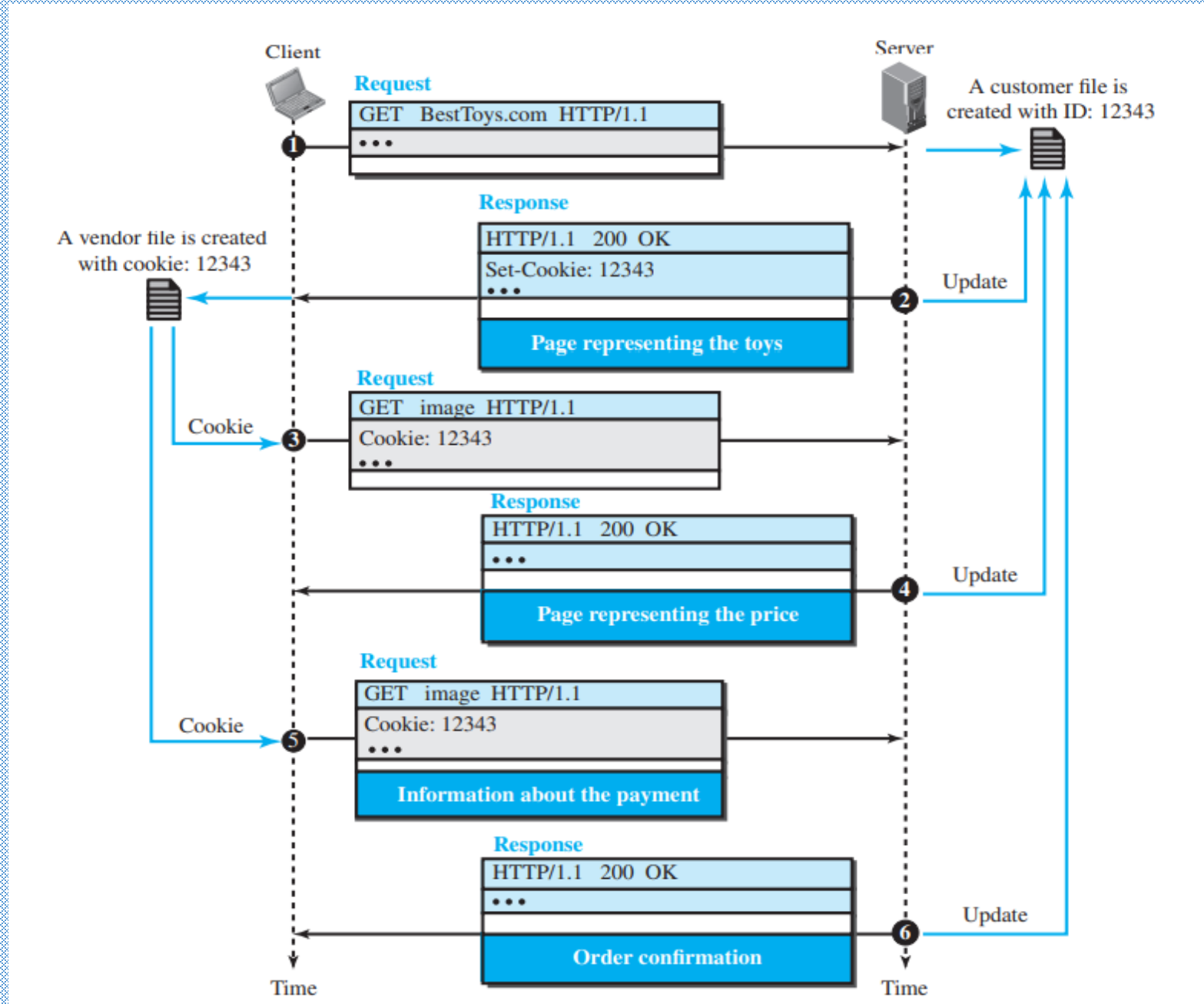
1. When a server receives a request from a client, it stores information about the client in a file or a string. The information may include the domain name of the client, the contents of the cookie (information the server has gathered about the client such as name, registration number, and so on), a 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 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.
- Now let us see how a cookie is used for the four previously mentioned purposes:

- 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 client when the client registers for the first time. For any repeated access, only those clients that 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 banner ads on some main website that is often visited by users. The advertising agency supplies only a URL that gives the advertising agency's address instead of the banner itself. When a user visits the main website and clicks the icon of a corporation, a request is sent to the advertising agency. The advertising agency sends the requested banner, but it also includes a cookie with the ID of the user. Any future use of the banners adds to the database that profiles the Web behavior of the user. The advertising agency has compiled the interests of the user and can sell this information to other parties. This use of cookies has made them very controversial. Hopefully, some new regulations will be devised to preserve the privacy of users.


Client
Server
Request
GET BestToys.com HTTP/1.1
• • •
A customer file is created with ID: 12343
Response
HTTP/1.1 200 OK
Set-Cookie: 12343
• • •
Page representing the toys
A vendor file is created with cookie: 12343
Update
Request
GET image HTTP/1.1
Cookie: 12343
• • •
Cookie
Response
HTTP/1.1 200 OK
• • •
Page representing the price
Update
Request
GET image HTTP/1.1
Cookie: 12343
• • •
Cookie
Information about the payment
Response
HTTP/1.1 200 OK
• • •
Order confirmation
Update
Time
Time

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.
- Note that the 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

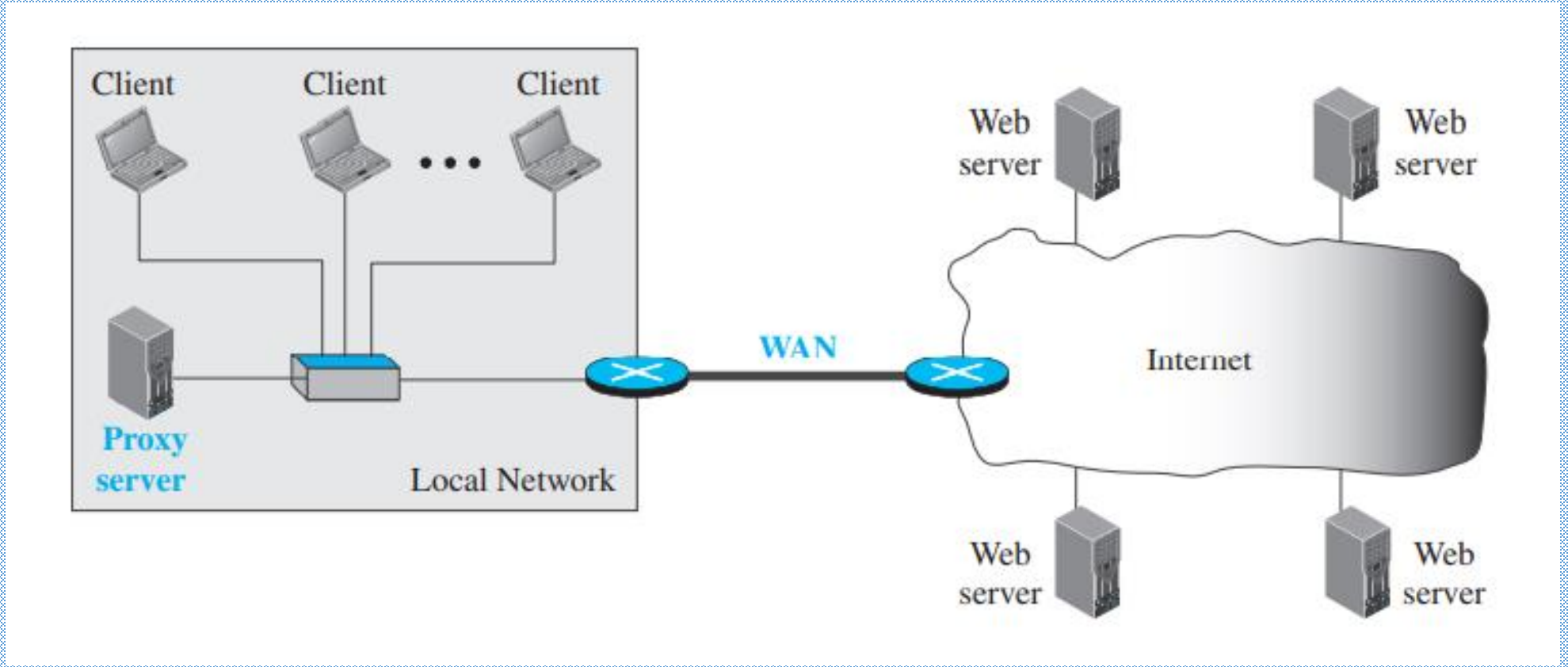
- The proxy servers are normally located at the client site. This means that we can have a hierarchy of proxy servers, as shown below:

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.

Example:


Client
Client
Client
Proxy server
Local Network
WAN
Internet
Web server
Web server
Web server
Web server

For how long should the response remain in the proxy server before being deleted and replaced ?

CACHE UPDATE

- 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

- 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.

HTTP VERSUS HTTPS

HTTP	HTTPS
An application protocol for distributed, collaborative, and hypermedia information systems	An extension of HTTP protocol for secure communication over a computer network
Stands for Hyper Text Transfer Protocol	Stands for Hyper Text Transfer Protocol Secure
Less secure	More secure
Works in the application layer of the OSI model	Works in the transport layer of the OSI model
Does not use a certificate	Uses SSL certificate
There is no encryption and decryption	There is encryption and decryption
Helps to transfer text, audio, video, images through web pages	Helps to transfer data securely via the network


thank
you