Chapter 1
HTTP Protocol

Presented by: Dr. R. BENTRCIA

Department of Computer Science, M'sila University

Outline

- Objectives
- Web Application Architecture
- HTTP Overview
- Uniform Resource Locator (URL)
- HTTP Features
- HTTP Operation
- HTTP Parameters
- HTTP Messages
- Apache HTTP Server
- HTTP Specifications

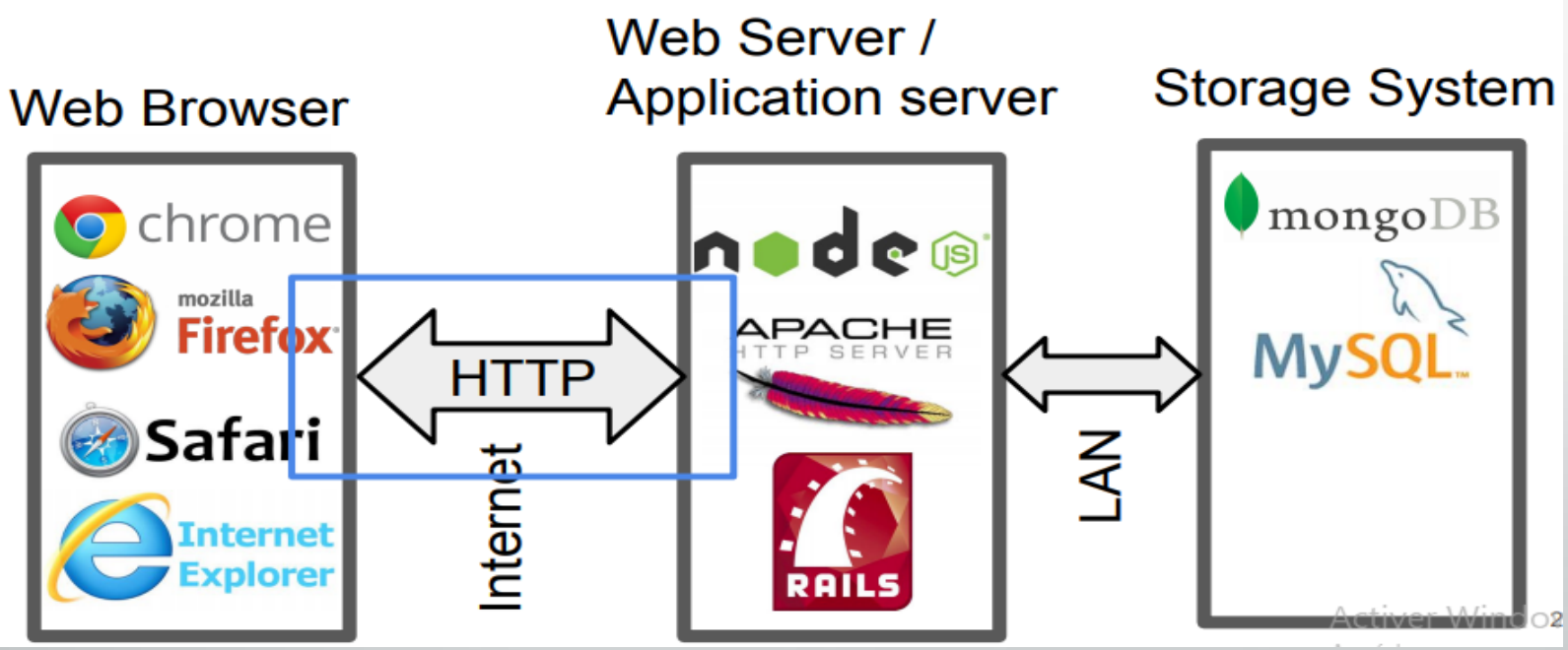
Objectives

- To gain an understanding of HTTP protocol, the various modes of operation of the HTTP protocol, and the different HTTP messages and elements used during HTTP requests and HTTP responses.

Web Application Architecture

- Many applications are running concurrently over the Web, such as web browsing/surfing, e-mail, file transfer, audio & video streaming, and so on.
- In order for proper communication to take place between the client and the server, these applications must agree on a specific application-level protocol such as HTTP, FTP, SMTP, POP, and etc.

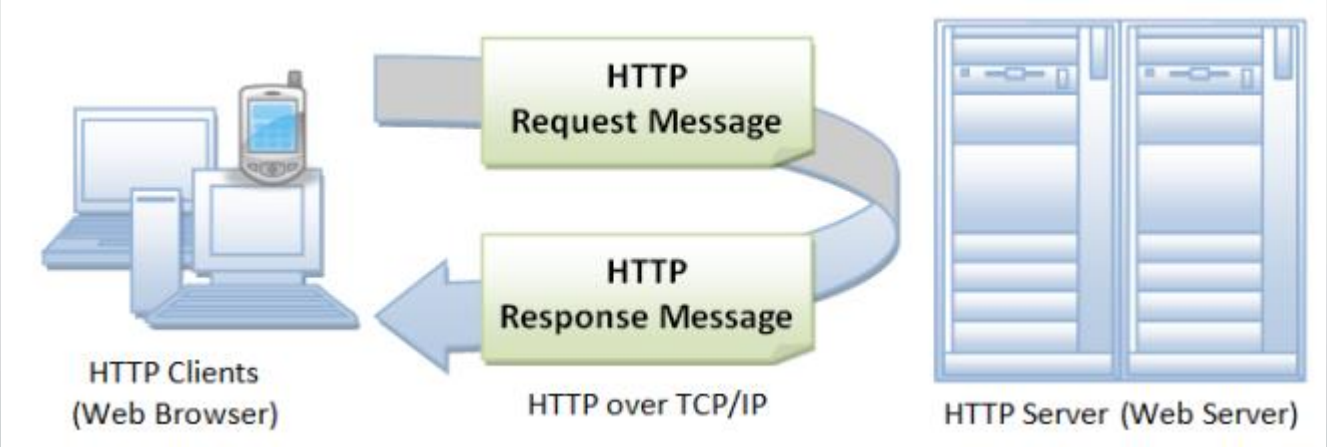
Web Application Architecture



HTTP Overview

- HTTP (HyperText Transport Protocol) is a simple request-response protocol layered on TCP/IP.
- Protocol for transfer of various data formats between server and client: Plaintext – Hypertext – Images – Video – Sound.

HTTP Overview



Uniform Resource Locator (URL)

- A URL (Uniform Resource Locator) is used to uniquely identify a resource over the web.
- URL has the following syntax:
  - protocol://hostname:port/path-and-file-name
- There are 4 parts in a URL:
  - Protocol: The application-level protocol used by the client and server, e.g., HTTP, FTP, and telnet.
  - Hostname: The DNS domain name (e.g., www.test101.com) or IP address (e.g., 192.128.1.2) of the server.
  - Port: The TCP port number that the server is listening for incoming requests from the clients.
  - Path-and-file-name: The name and location of the requested resource, under the server document base directory.
- For example, in the URL http://www.test101.com/docs/index.html, the communication protocol is HTTP; the hostname is www.test101.com. The port number was not specified in the URL, and takes on the default number, which is TCP port 80 for HTTP. The path and file name for the resource to be located is "/docs/index.html".

HTTP Features

- The most typical use of HTTP is between a Web browser and a Web server.
- In a typical scenario, a new TCP connection is created between client and server for each transaction and then terminated as soon as the transaction completes.
- Stateless:
  - Each transaction treated independently
  - Each new TCP connection for each transaction
  - Terminate connection when transaction complete

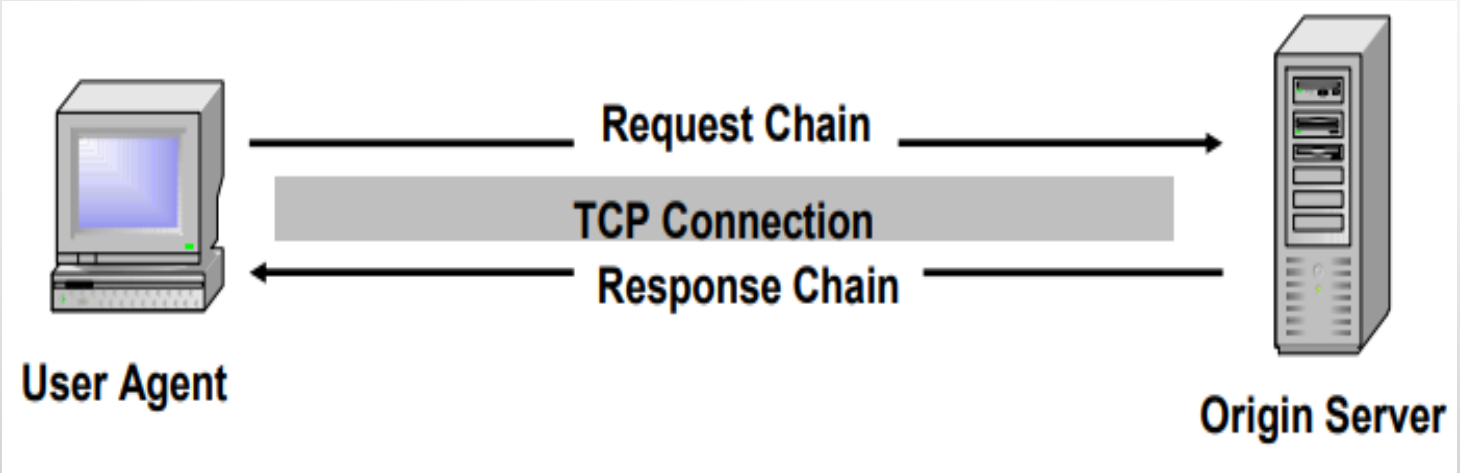
HTTP Features

- **HTTP is connectionless**: The HTTP client /browser initiates an HTTP request and after a request is made, the client disconnects from the server and waits for a response. The server process the request and re-establish the connection with the client to send response back.
- **HTTP is media independent**: This means, any type of data can be sent by HTTP as long as both the client and server know how to handle the data content. This is required for client as well as server to specify the content type using appropriate MIME-type.
- **HTTP is stateless**: As mentioned above, HTTP is a connectionless and this is a direct result that HTTP is a stateless protocol. The server and client are aware of each other only during a current request. Afterwards, both of them forget about each other. Due to this nature of the protocol, neither the client nor the browser can retain information between different request across the web pages.

HTTP Operation

- Typically there are three examples of HTTP operation:
  - Direct connection
  - Intermediate systems
  - A cache

HTTP Operation – Direct Connection



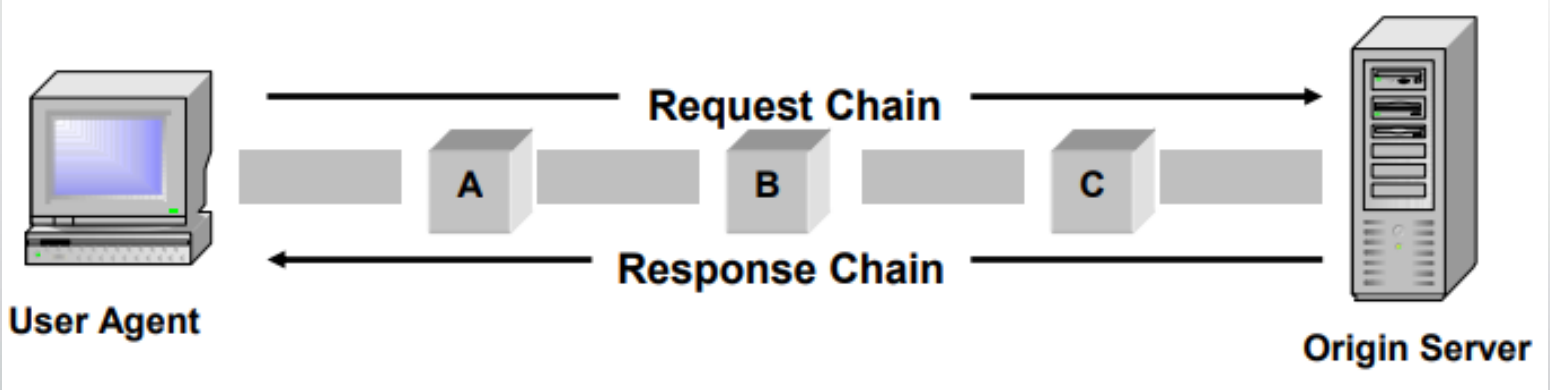
HTTP Operation – Direct Connection

- This is the simplest case, in which a user agent or client (e.g., a Web browser) establishes a direct connection with the origin server (e.g., Web server). First, the client opens an end-to-end TCP connection between the client and server.
- Then the client issues a request that consists of a URL and a MIME-like message containing request parameters, information about the client, and perhaps some additional content information.

HTTP Operation – Direct Connection

- The server uses a MIME configuration file (called "conf\mime.types") to map the file extension to a media type, so that it can ascertain the media type of the file by looking at its file extension. For example, file extensions ".htm", ".html" are associated with MIME media type "text/html", file extension of ".jpg", ".jpeg" are associated with "image/jpeg". When a file is returned to the client, the server has to put up a Content-Type response header to inform the client the media type of the data.
- When the server receives the request, it attempts to complete the request and returns an HTTP response containing status information, a success/error code, and a MIME-like message containing information about the server, information about the response itself, and possible body content. The TCP connection is then closed.

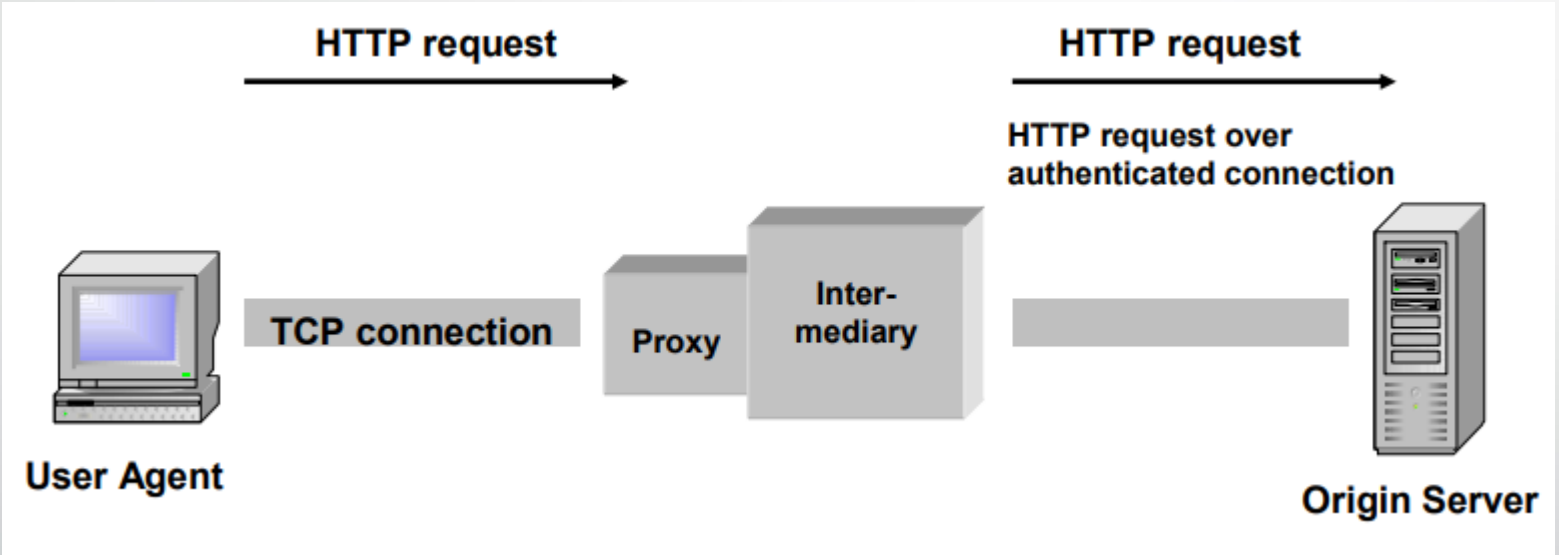
HTTP Operation – Intermediate Systems



HTTP Operation – Intermediate Systems

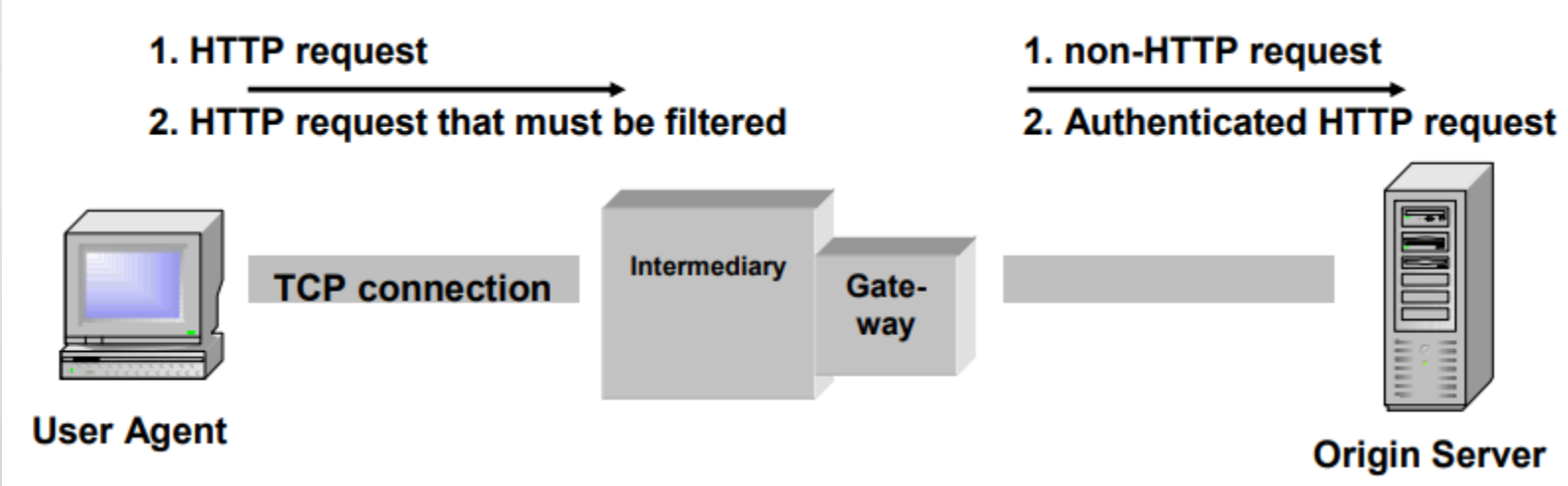
- In this scenario, there are one or more intermediate systems with TCP connections between logically adjacent systems.
- Each intermediate system acts as a relay, so that a request that is initiated by the client is relayed through the intermediate systems to the server, and the response from the server is relayed back to the client.
- There are three forms of intermediate systems defined in the HTTP specification:
  - Proxy
  - Gateway
  - Tunnel

Intermediate Systems - Proxy



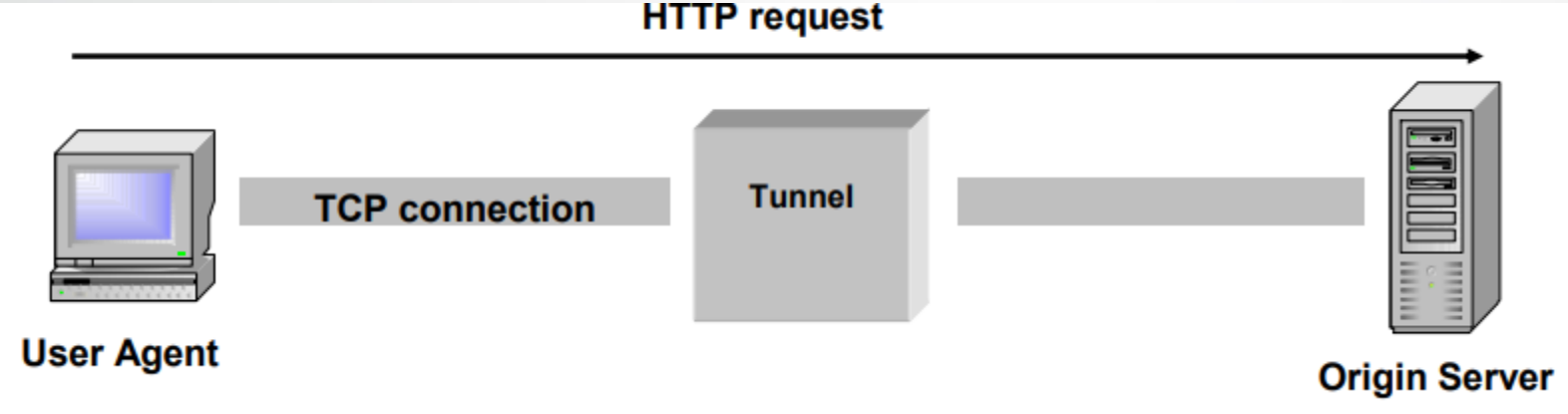
- Proxy: it acts on behalf of other clients and presents requests from other clients to a server. The proxy acts as a server in interacting with a client and as a client in interacting with a server.

Intermediate Systems - Gateway



- Gateway: it is a server that appears to the client as if it were an origin server. It acts on behalf of other servers that may not be able to communicate directly with a client.

Intermediate Systems - Tunnel



- The tunnel performs NO operations on HTTP requests and responses. It is a relay point between two TCP connections. Tunnels are used when there must be an intermediary between the client and server, e.g., a firewall in which a client or server external to a protected network can establish an authenticated connection and then maintain that connection for purposes of HTTP transactions.

HTTP Parameters

- A list of the important HTTP Protocol Parameters and their syntax in the way they are used in the communication:
- **HTTP Version**: The version of an HTTP message is indicated by an HTTP-Version field in the first line.
  - Example: HTTP/1.0 or HTTP/1.1

HTTP Parameters

- **Uniform Resource Identifiers URI**: it is simply formatted, case-insensitive string containing name, location etc to identify a resource, for example a website, a web service etc.
- Example: http://abc.com:80/~smith/home.html

HTTP Parameters

- **Date/Time Formats**: All HTTP date/time stamps MUST be represented in Greenwich Mean Time GMT, without exception. HTTP applications are allowed to use any of the following three representations of date/time stamps:
- example:
  - Sun, 06 Nov 1994 08:49:37 GMT ;
  - Sunday, 06-Nov-94 08:49:37 GMT ;
  - Sun Nov 6 08:49:37 1994

HTTP Parameters

- **Character Sets**: You use character set to specify the character sets that the client prefers. Multiple character sets can be listed separated by commas. If a value is not specified, the default is US-ASCII.
- Example:
  - US-ASCII or
  - ISO-8859-1 or
  - ISO-8859-7

HTTP Parameters

- **Content Encodings**: A content encoding values indicate an encoding algorithm has been used to encode the content before passing it over the network. Content codings are primarily used to allow a document to be compressed or otherwise usefully transformed without losing the identity.
- Example:
  - Accept-encoding: gzip or
  - Accept-encoding: compress or
  - Accept-encoding: deflate

HTTP Parameters

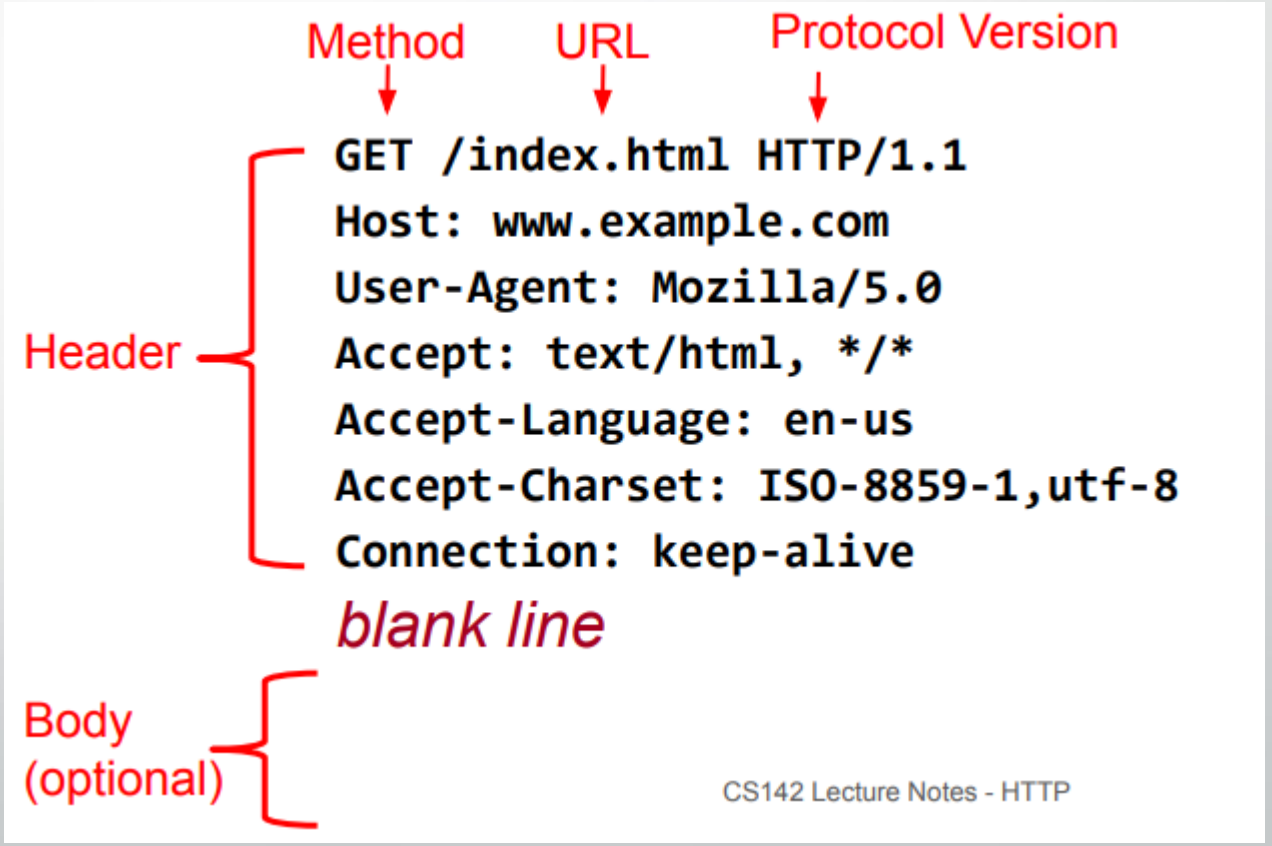
- **Media Types:** HTTP uses Internet Media Types in the Content-Type and Accept header fields in order to provide open and extensible data typing and type negotiation. All the Media-type values are registered with the Internet Assigned Number Authority IANA.
- Example:
  - Accept: image/gif

HTTP Parameters

- **Language Tags**: HTTP uses language tags within the Accept-Language and Content-Language fields. A language tag is composed of 1 or more parts: A primary language tag and a possibly empty series of sub tags:
- Example:
  - en, en-US, en-cockney, i-cherokee, x-pig-latin

HTTP Messages

- HTTP Requests:
  - whenever you enter a URL in the address box of the browser, the browser translates the URL into a request message according to the specified protocol; and sends the request message to the server.



HTTP Messages

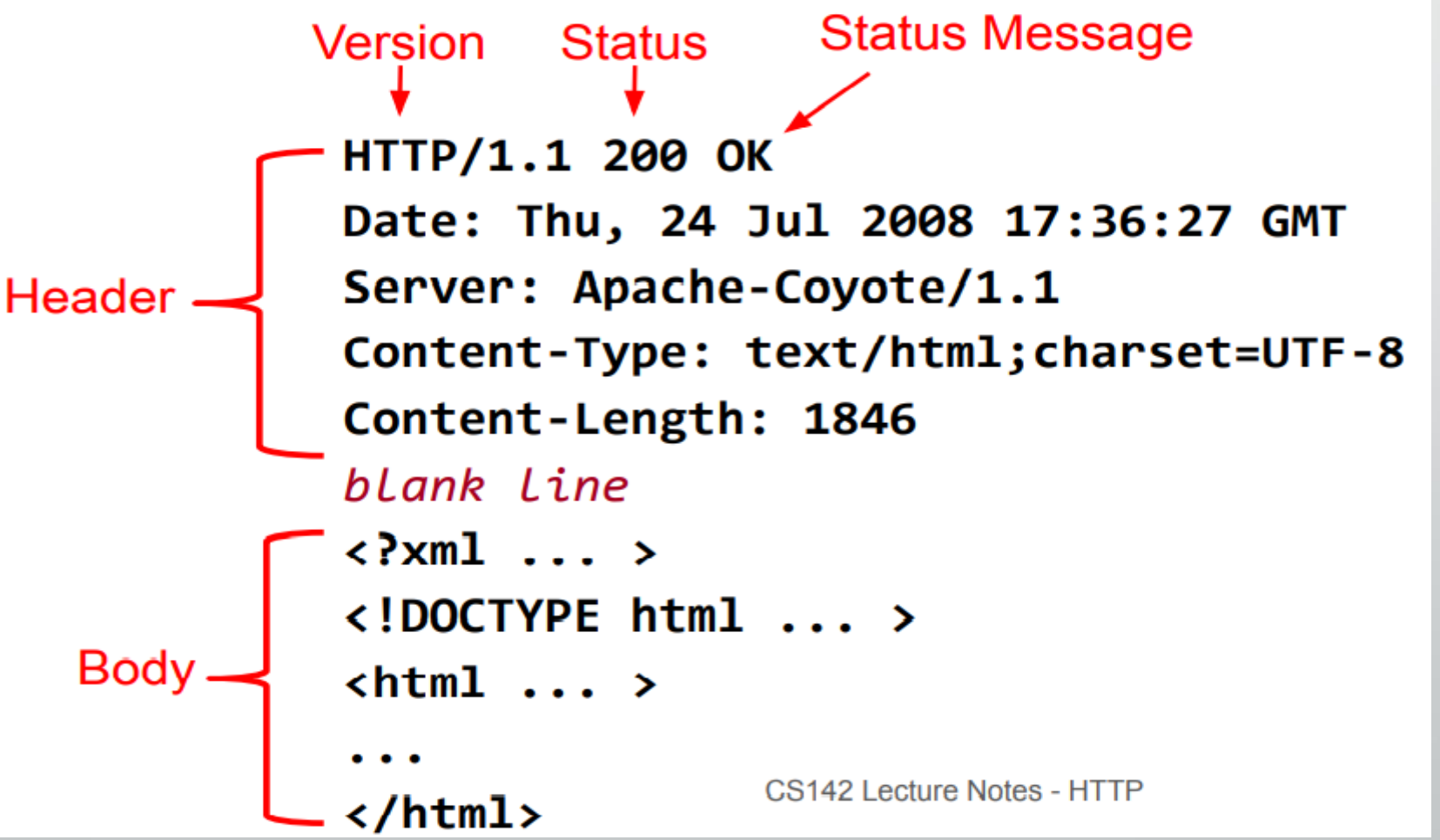
- HTTP Methods (Verbs):
  - GET - Fetch a URL
  - HEAD - Fetch information about a URL
  - PUT - Store to an URL
  - POST - Send form data to a URL and get a response back
  - DELETE - Delete a URL
- GET and POST (forms) are commonly used.

HTTP Messages

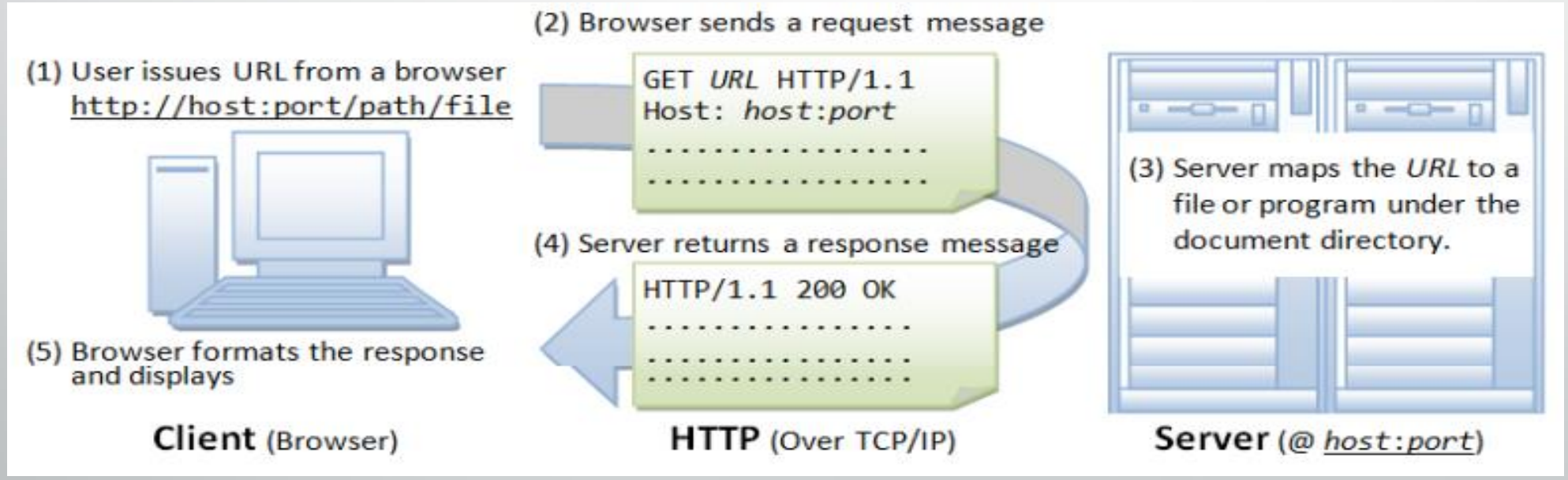
- When this request message reaches the server, the server can take either one of these actions:
  - The server interprets the request received, maps the request into a file under the server's document directory, and returns the file requested to the client.
  - The server interprets the request received, maps the request into a program kept in the server, executes the program, and returns the output of the program to the client.
  - The request cannot be satisfied, the server returns an error message.

HTTP Messages

- HTTP Response:



HTTP Messages



HTTP Messages

- Common HTTP Response Status Codes:
  - 200 OK — Success
  - 307 Temporary Redirect — Redirection - Browser retries using Location header
  - 404 Not Found — Famous one
  - 503 Service Unavailable — Something crashed on the server
  - 500 Internal Server Error — Something is messed up on the server
  - 501 Not Implemented — Coming
  - 400 Bad Request — Use if web app sends bogus request
  - 401 Unauthorized — Use if user isn't logged in
  - 403 Forbidden — Use if even logging in wouldn't help
  - 550 Permission denied — Not allow to perform request

Apache HTTP Server

- A HTTP server (such as Apache HTTP Server or Apache Tomcat Server) is needed to study the HTTP protocol.
- Apache HTTP server is a popular industrial-strength production server, produced by Apache Software Foundation (ASF).
  - www.apache.org.
- ASF is an open-source software foundation. That is to say, Apache HTTP server is free, with source code.

HTTP Specifications

- The HTTP specification is maintained by W3C (World-wide Web Consortium).
- The original version, HTTP/0.9 (1991), written by Tim Berners-Lee, is a simple protocol for transferring raw data across the Internet.
- HTTP/1.0 (1996), improved the protocol by allowing MIME-like messages.
- HTTP/1.0 does not address the issues of proxies, caching, persistent connection, virtual hosts, and range download. These features were provided in HTTP/1.1 (1999) (defined in RFC 2616).

HTTP Specifications

- Persistent (or Keep-alive) Connections:
  - In HTTP/1.0, the server closes the TCP connection after delivering the response by default (Connection: Close). That is, each TCP connection holds only one request. This is not efficient as many HTML pages contain hyperlinks.
  - The client can negotiate with the server and ask the server not to close the connection after delivering the response, so that another request can be sent through the same connection. This is known as persistent connection (or keep-alive connection).
  - For HTTP/1.1, the default connection is persistent. The client do not have to sent the "Connection: Keep-alive" header. Instead, the client may wish to send the header "Connection: Close" to ask the server to close the connection after delivering the response.

References

- HTTP Protocol, Professor Richard Harris, School of Engineering and Advanced Technology (SEAT),
- http://www.tutorialspoint.com/http/http_quick_guide.htm
- W3C HTTP specifications at http://www.w3.org/standards/techs/http.