Hypertext Transfer PROTOCOL HTTP

Ανάπτυξη Διαδικτυακών Συστημάτων & Εφαρμογών

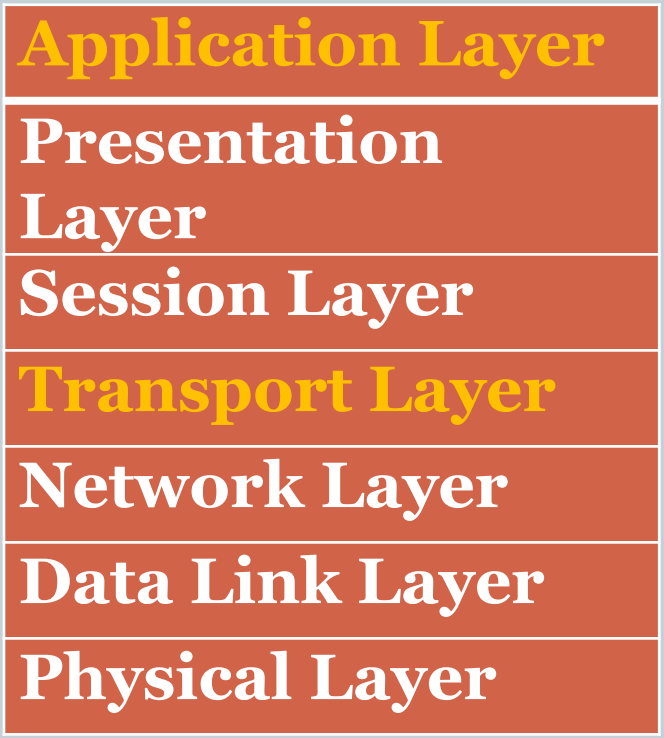
Τμ. Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων

ΔιΠαΕ

Αντώνης Σιδηρόπουλος

Overview

- HTTP is the abbreviation of **HyperText Transfer Protocol,** which perfumes a reliable Transport layer protocol for host-to-host data transfer. HTTP is located in the application-layer of the OSI model. Consider the different uses, HTTP can use TCP protocol for a reliable data transport as well as UDP for a unreliable transport.



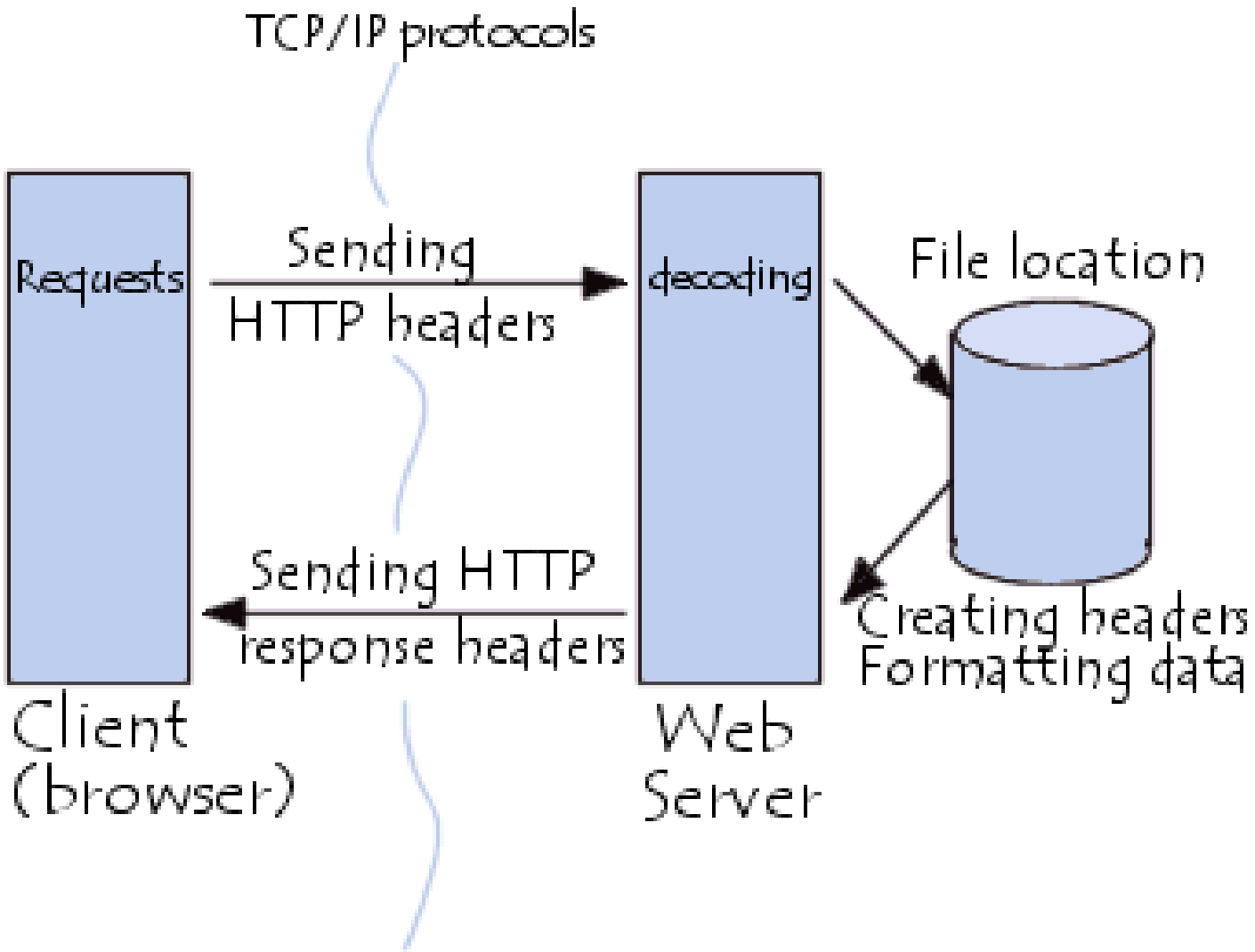
History

- HTTP/0.9 was the first one HTTP protocol and very simple, which only got one method, namely GET. The method could request a page from Server and the response always was an HTML page.
- HTTP/1.0 worked as the extension of HTTP/0.9. It expanded the protocol extended operations, extended negotiation, richer meta-information, tied with a security protocol and got more efficient by adding additional methods and header fields. But it uses a separate connection to the same server for every request-response transaction.
- While HTTP/1.1 can reuse a connection multiple times, for example, to download images or documents for a just delivered page. Hence HTTP/1.1 communications experience less latency as the establishment of TCP connections presents considerable overhead.
- HTTP/2 more efficient than http/1.1. HTTP/2 allows the server to "push" content, multiplexing of requests, header compression, and prioritization of requests.

HTTP Session

- Usually, HTTP functions as a Request-Response model, namely, a Client-Server model.

- Client asks for **URLs**(Uniform Resources Locator) or **URIs**(Uniform Resources Identifier).


TCP/IP protocols
Requests
Sending
HTTP headers
decoding
File location
Sending HTTP
response headers
Creating headers
Formatting data
Client
(browser)
Web
Server

HTTP Request

- A request Line: method + URL + protocol version
  - GET www.fit.edu HTTP/1.1
- Request Header Field
- Body of Request

Request Methods

Command	Description
GET	Request for the resource located at the specified URL.
HEAD	It like GET, but without the body of response.
POST	Submits data to be processed to the identified resource. The data is included in the body of the request.
PUT	Uploads a representation of the specified resource.
DELETE	Deletes the resource located at the specified URL.
TRACE	Echoes back the received request, so that a client can see what (if any) changes or additions have been made by intermediate servers.
OPTIONS	Returns the HTTP methods that the server supports for specified URL. This can be used to check the functionality of a web server by requesting '*' instead of a specific resource.
CONNECT	Converts the request connection to a transparent TCP/IP tunnel, usually to facilitate SSL-encrypted communication (HTTPS) through an unencrypted HTTP proxy.
PATCH	Is used to apply partial modifications to a resource.

Request Headers

Header name	Description
Accept	Type of content accepted by the browser (for example text/html).
Accept-Charset	Character set expected by the browser
Accept-Encoding	Data coding accepted by the browser
Accept-Language	Language expected by the browser (English by default)
Authorization	Identification of the browser to the server
More….	

HTTP Response

- A Response line: protocol version + status code + meaning of the code.
  - HTTP/1.1 200 OK
- Response Header Field
- Body of Response

Response status code

- 1xx: Informational
- 2xx: Success
- 200: OK
- 3xx: Redirection, not modified
- 301: Moved
- 4xx: Client Error
- 403: Forbidden
- 404: Not found
- 5xx: Server Error
- 502: Bad gateway

Response Headers

Header name	Description
Content-Encoding	Type of coding for the body of the response
Content-Language	Type of language in the body of the response
Content-Length	Length of the body of the response
Content-Type	Type of content of the body of the response (for example text/html).
Date	Date data transfer starts.
Expires	Data use by date
Forwarded	Used by intermediary machines between the browser and server
Location	Redirection to a new URL associated with the document
Server	Features of the server having sent the response

HTTP Example

telnet www.in.gr 80

Request

```
GET / HTTP/1.1
Host: www.in.gr
<LF>
```

Response

```
HTTP/1.1 200 OK
Cache-Control: private,max-age=1,stale-while-revalidate=100,stale-if-error=360
Content-Type: text/html; charset=utf-8
Vary: Accept-Encoding
Access-Control-Allow-Origin: http://mobile.in.gr
X-Cache-OBJECT-TTL: 60.000
Transfer-Encoding: chunked
Date: Tue, 24 Oct 2017 10:42:49 GMT
Age: 29
Connection: keep-alive
Server: DOLServers

008000
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="el">
<head>
        <meta http-equiv='Content-Type' content='text/html; charset=utf-8' />
        <meta http-equiv='Content-Language' content='el' />
```

HTTP Example (2)

telnet www.in.gr 80

Request

```
GET http://www.in.gr/ HTTP/1.1
Host: www.in.gr
<LF>
```

Response

```
HTTP/1.1 200 OK
Cache-Control: private,max-age=1,stale-while-revalidate=100,stale-if-error=360
Content-Type: text/html; charset=utf-8
Vary: Accept-Encoding
Access-Control-Allow-Origin: http://mobile.in.gr
X-Cache-OBJECT-TTL: 60.000
Transfer-Encoding: chunked
Date: Tue, 24 Oct 2017 10:42:49 GMT
Age: 29
Connection: keep-alive
Server: DOLServers

008000
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="el">
<head>
        <meta http-equiv='Content-Type' content='text/html; charset=utf-8' />
        <meta http-equiv='Content-Language' content='el' />
```

HTTP GET Request – Example

- Example of HTTP **GET** request:

```
GET
/~asidirop/adise/theory/http_tests/?arg1=test_for_ar
g_1&second+arg=value+for+second_argument HTTP/1.1
Host: asidirop-srv.it.teithe.gr
Accept: */*
Accept-Language: bg
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0(compatible;MSIE 6.0;Windows
NT 5.0)
Connection: Keep-Alive
Cache-Control: no-cache
<CRLF>
```

HTTP request line

HTTP request headers

The request body is empty

HTTP POST Request – Example

- Example of HTTP **POST** request:

```
POST /~asidirop/adise/theory/http_tests/ HTTP/1.1
Host: asidirop-srv.it.teithe.gr
Accept: */*
Accept-Language: bg
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0(compatible;MSIE 6.0;Windows
NT 5.0)
Connection: Keep-Alive
Cache-Control: no-cache
<CRLF>
arg1=test_for_arg_1&second+arg=value+for+second_argu
ment<CRLF>
```

HTTP request line

HTTP request headers

The request body

Command line tools

- Wget (windows + linux)

```
asidirop@aetos:~$ wget -O test.html http://www.in.gr  -d
DEBUG output created by Wget 1.12 on linux-gnu.
---request begin---
GET / HTTP/1.0
User-Agent: Wget/1.12 (linux-gnu)
Accept: */*
Host: www.in.gr
Connection: Keep-Alive
---request end---
HTTP request sent, awaiting response...
---response begin---
HTTP/1.1 200 OK
Cache-Control: private,max-age=1,stale-while-revalidate=100,stale-if-error=3600
Content-Type: text/html; charset=utf-8
Vary: Accept-Encoding
Access-Control-Allow-Origin: http://mobile.in.gr
X-Cache-OBJECT-TTL: 60.000
Date: Tue, 24 Oct 2017 11:06:26 GMT
Age: 41
Connection: close
Server: DOLServers
---response end---
200 OK
Length: unspecified [text/html]
Saving to: “test.html”
```

Command line tools

- Curl (windows + linux)

```
asidirop@users:~$ curl http://www.in.gr/  -v > /tmp/p
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0*   Trying
213.133.127.245...
* TCP_NODELAY set
* Connected to www.in.gr (213.133.127.245) port 80 (#0)
> GET / HTTP/1.1
> Host: www.in.gr
> User-Agent: curl/7.52.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Cache-Control: private,max-age=1,stale-while-revalidate=100,stale-if-error=3600
< Content-Type: text/html; charset=utf-8
< Vary: Accept-Encoding
< Access-Control-Allow-Origin: http://mobile.in.gr
< X-Cache-OBJECT-TTL: 60.000
< Transfer-Encoding: chunked
< Date: Tue, 24 Oct 2017 11:10:37 GMT
< Age: 50
< Connection: keep-alive
< Server: DOLServers
<
{ [14121 bytes data]
* Curl http done: called premature == 0
```

Browser Debugger (Network)