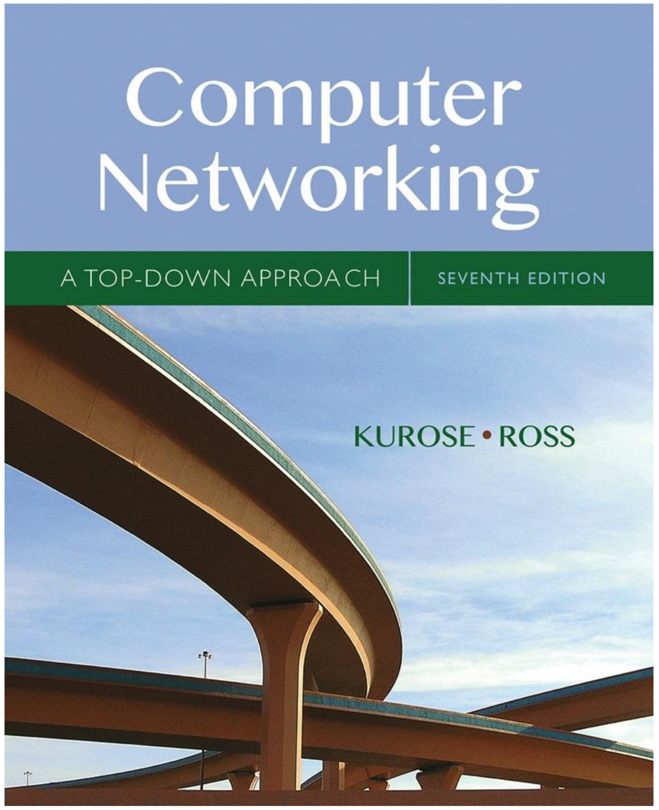
Chapter 2 Application Layer

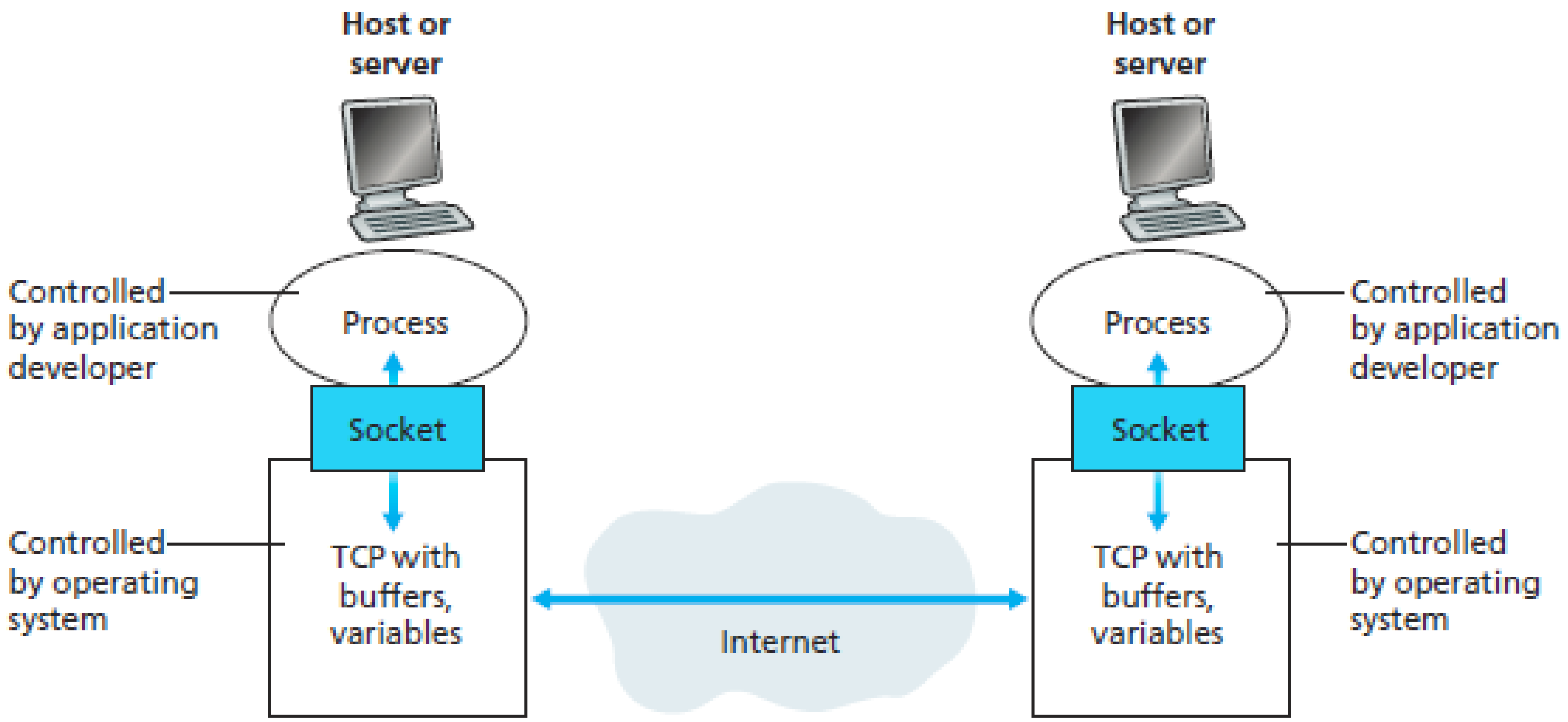
Supplementary materials



Computer Networking: A Top Down Approach

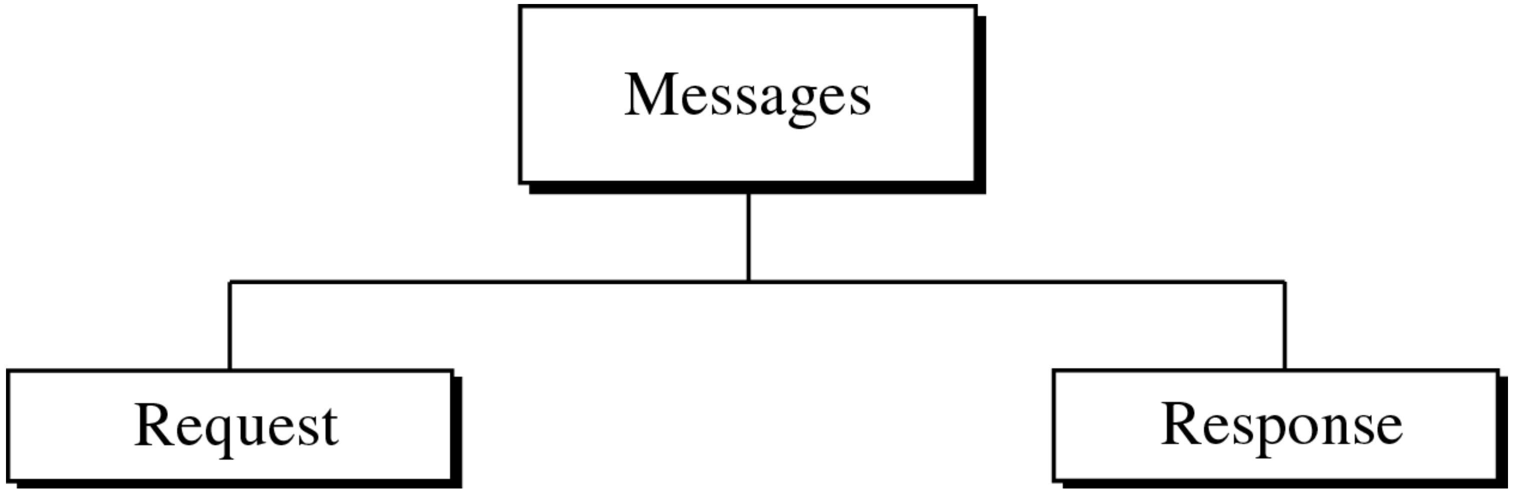
7th edition
Jim Kurose, Keith Ross
Pearson/Addison Wesley
April 2016

Application processes, sockets, and underlying transport protocol

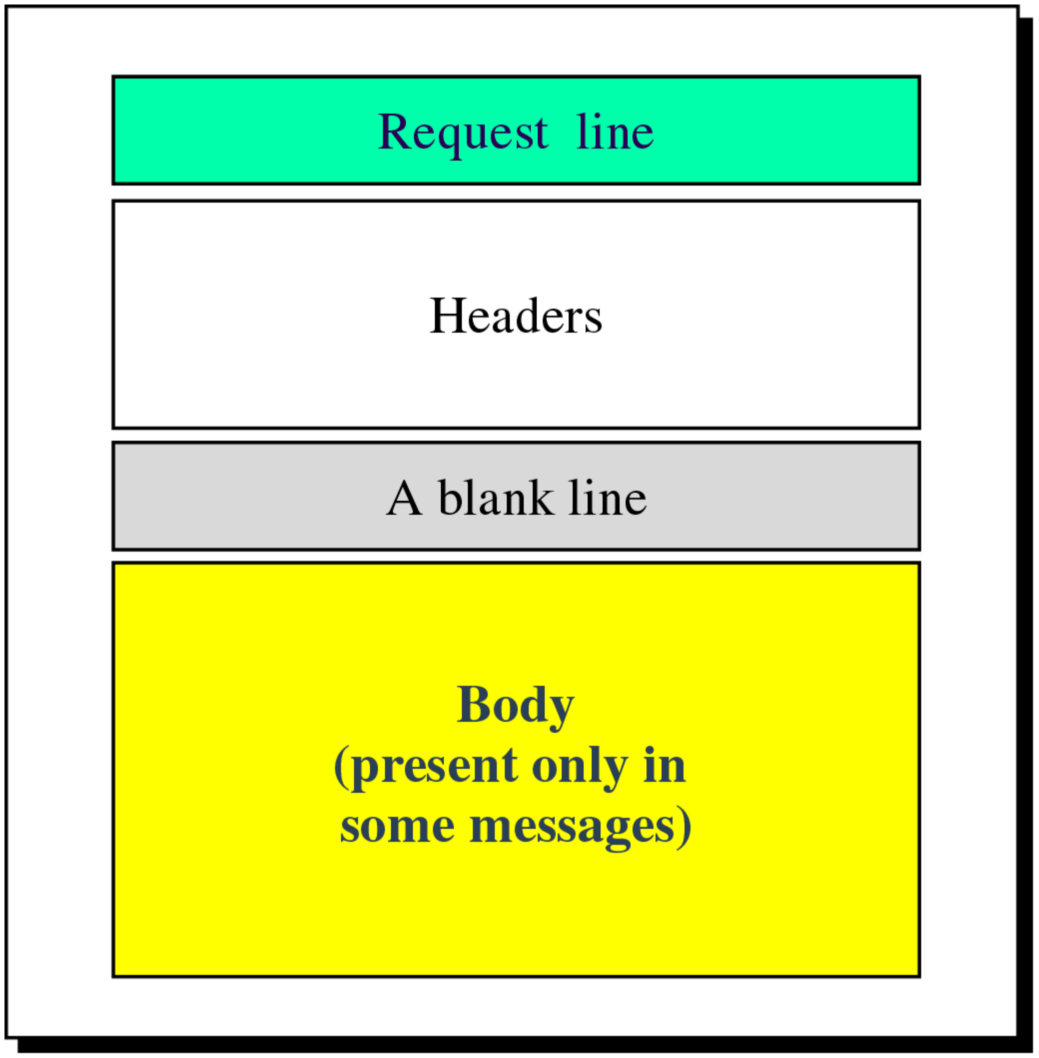

Host or server
Host or server
Controlled by application developer
Process
Socket
Controlled by operating system
TCP with buffers, variables
Internet
Process
Socket
TCP with buffers, variables
Controlled by application developer
Controlled by operating system

HTTP Message categories

- Two types of HTTP messages
  - request messages and response messages

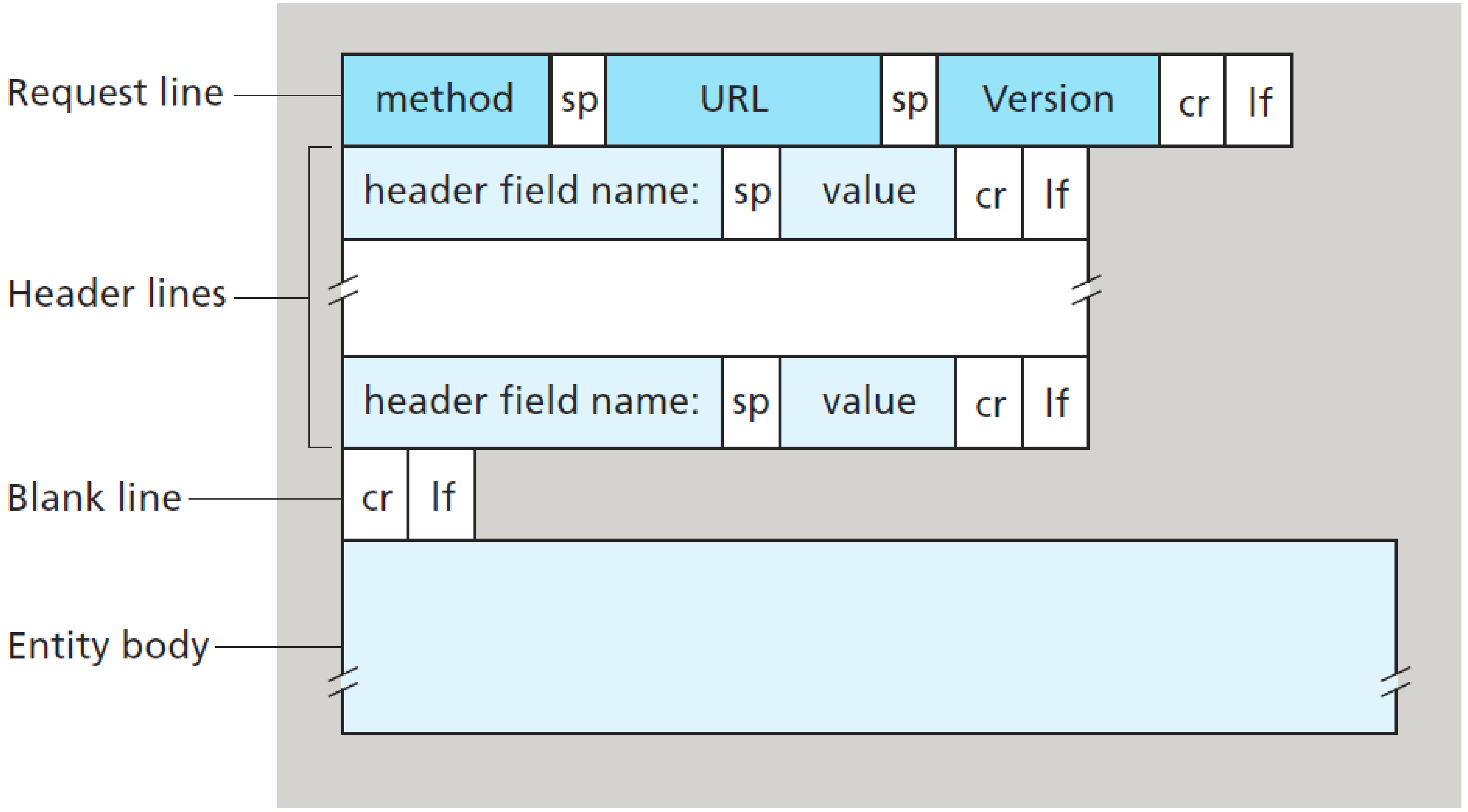


Request Message



Request message

Request Message (cont.)



- space (sp)
- carriage return (*cr*) – aka, return
- line feed (*lf*) – next line

Request Message (cont.)

- is written in ordinary ASCII text
- consists of lines, each followed by a carriage return (*cr*) and a line feed (*lf*)
- request line has three fields
  1. the method field: **GET**, POST, HEAD, PUT, and DELETE
  2. the Uniform Resource Locator (URL) field
     - The GET method is used when the browser requests an object, with the requested object identified in the URL field
  3. the HTTP version field
     - 1.0, 1.1, 2.0

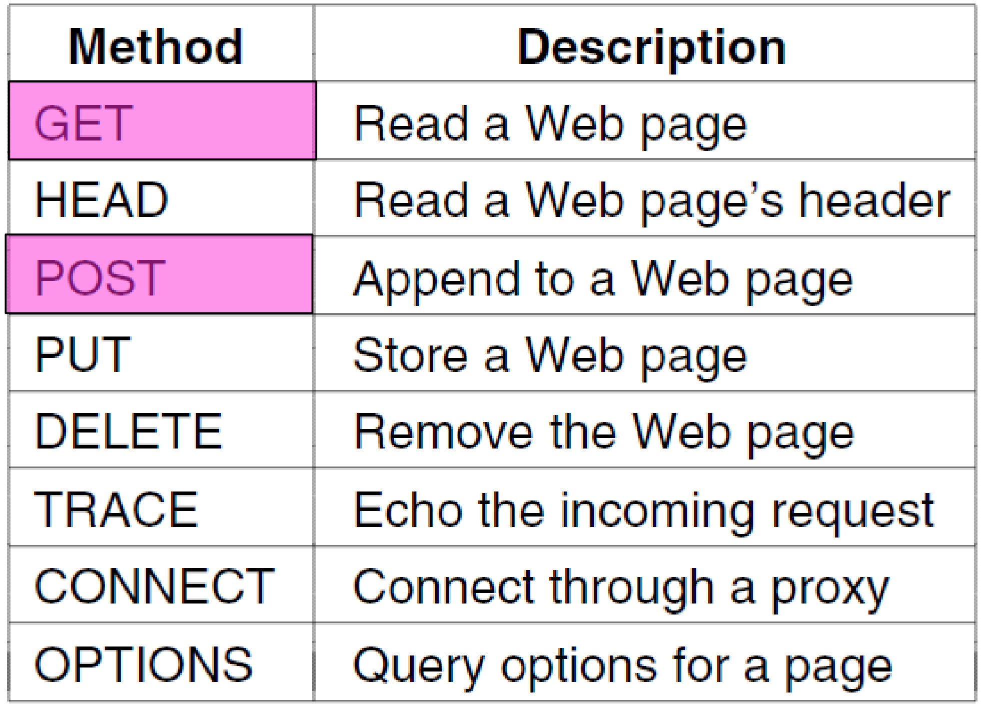
ASCII stands for American Standard Code for Information Interchange

Request Message (cont.)

- HTTP has several request methods.

Fetch a page →

Used to send input data to a server program →



Method	Description
GET	Read a Web page
HEAD	Read a Web page's header
POST	Append to a Web page
PUT	Store a Web page
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Connect through a proxy
OPTIONS	Query options for a page

Request Message (cont.)

- Pages are named with URLs (Uniform Resource Locators)
- Example: $\underbrace{\text{http://}}_{\text{Protocol}}\underbrace{\text{www.phdcomics.com}}_{\text{Server}}\underbrace{\text{/comics.php}}_{\text{Page on server}}$

Our focus $\rightarrow$ http

Name	Used for	Example
http	Hypertext (HTML)	http://www.ee.uwa.edu/~rob/
https	Hypertext with security	https://www.bank.com/accounts/
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	file:///usr/suzanne/prog.c
mailto	Sending email	mailto:JohnUser@acm.org
rtsp	Streaming media	rtsp://youtube.com/montypython.mpg
sip	Multimedia calls	sip:eve@adversary.com
about	Browser information	about:plugins

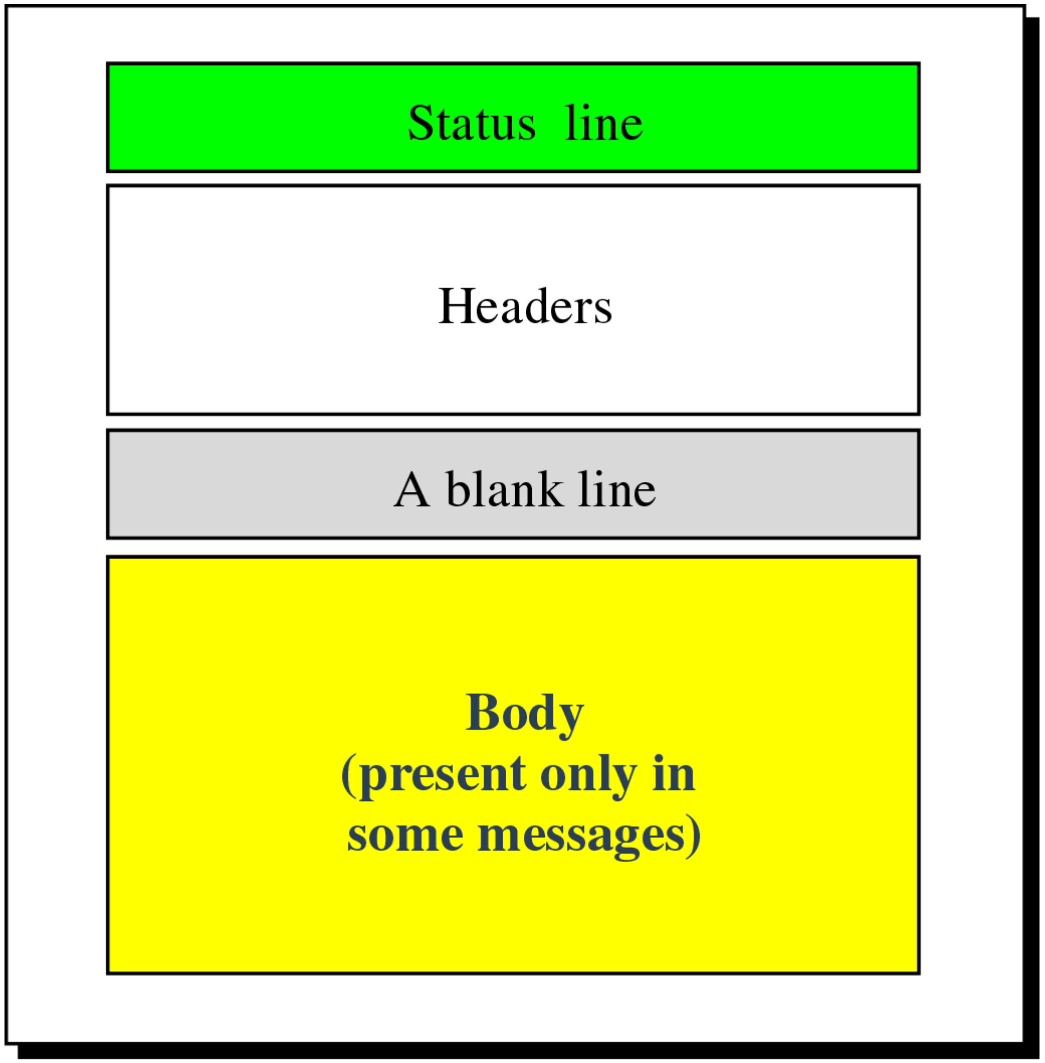
Common URL protocols

Request Message (cont.) - example

```
1  GET /somedir/page.html HTTP/1.1
2  Host: www.someschool.edu
3  Connection: close
4  User-agent: Mozilla/5.0
5  Accept-language: fr
```

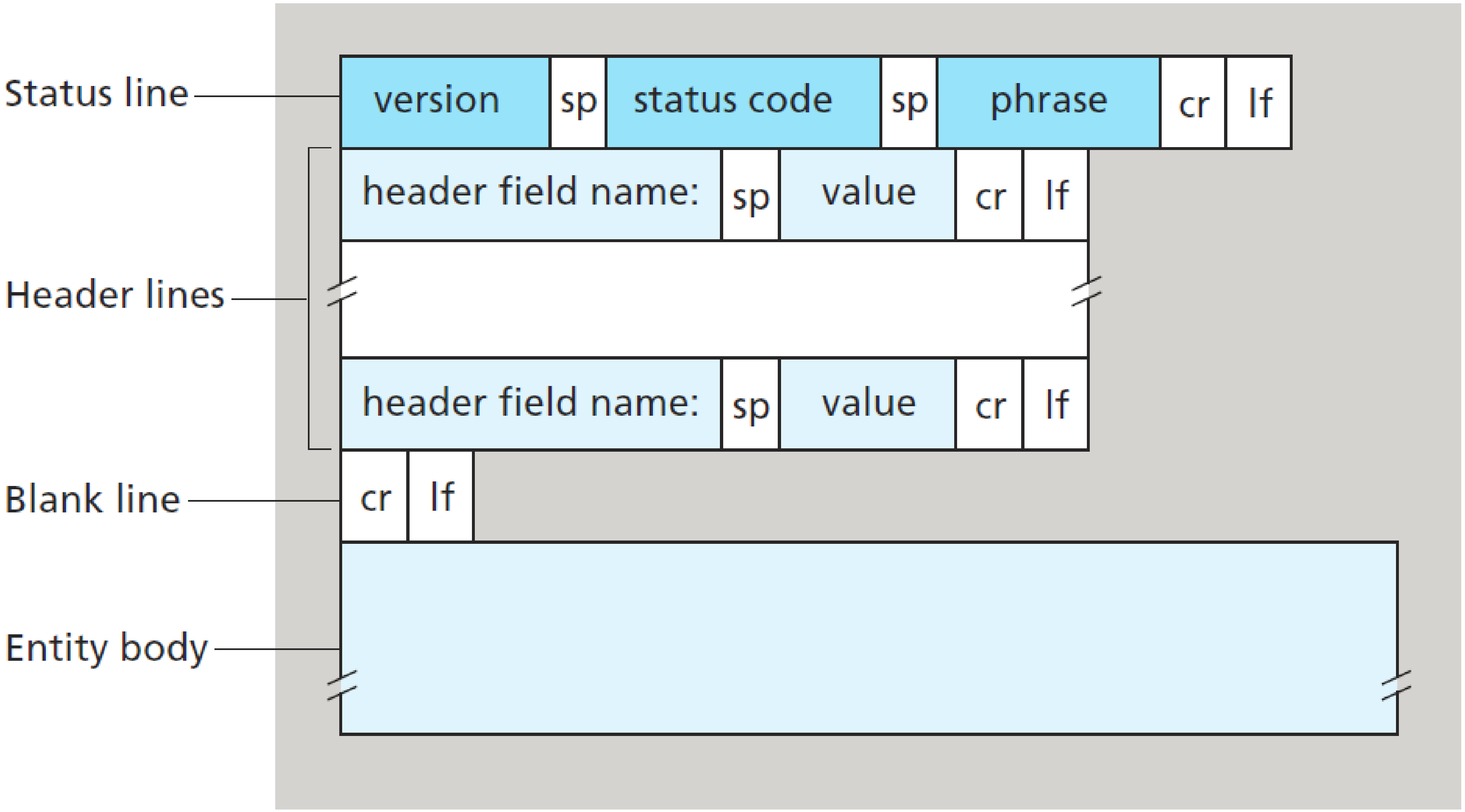
1. the browser is requesting the object /somedir/page.html; the browser implements version HTTP/1.1.
2. The header line Host: ww.someschool.edu specifies the host on which the object resides.
3. By including the Connection: close header line,
   - the browser is telling the server that it doesn't want to bother with persistent connections;
   - it wants the server to close the connection after sending the requested object.
4. The User-agent: header line specifies the user agent, that is, the browser type; Mozilla/5.0, a Firefox browser
5. Accept language: header indicates that the user prefers to receive a French (*fr*) version of the object as default
   - eg. Accept-Language: en-nz, en-gb;q=0.8, en;q=0.7
   - "I prefer New Zealand English, but will accept British English and other types of English."

Response message format



Response message

Response message format (cont.)



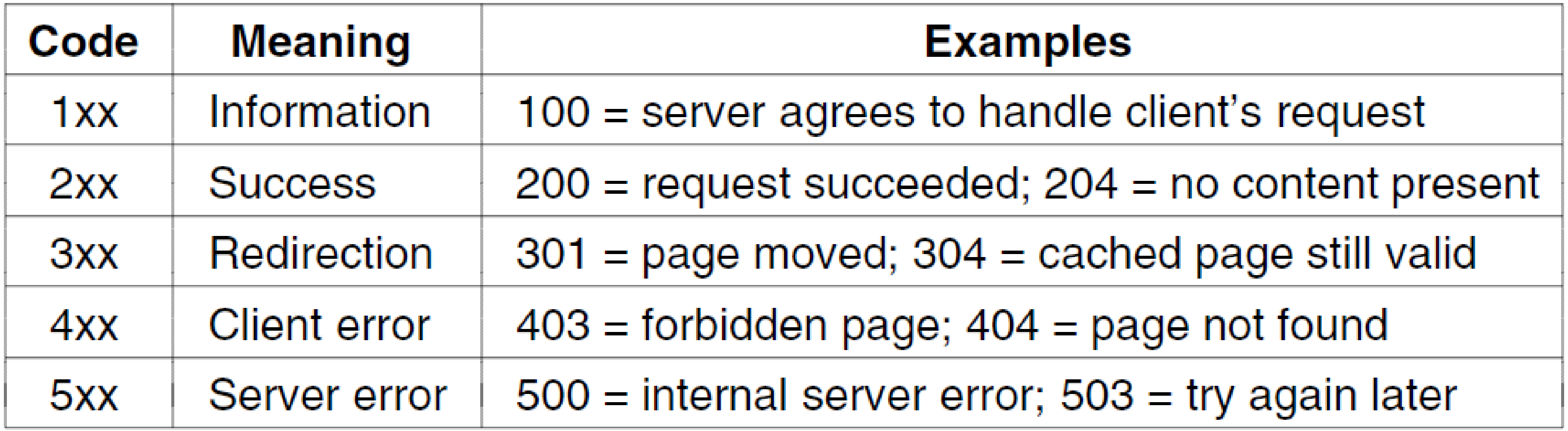
Response message (cont.)

- has three sections:

1. an initial status line
   - the protocol version field, **a status code**, and a corresponding status message
2. Six header lines
3. the entity body
   - contains the requested object itself (represented by data data data data ...)

Response message (cont.)

- Response codes tell the client how the request fared:



Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later

Response message (cont.)

- Status code examples
  - 200 OK: Request succeeded and the information is returned in the response.
  - 301 Moved Permanently: Requested object has been permanently moved; the new URL is specified in Location: header of the response message. The client software will automatically retrieve the new URL.
  - 400 Bad Request: This is a generic error code indicating that the request could not be understood by the server.
  - 404 Not Found: The requested document does not exist on this server.
  - 505 HTTP Version Not Supported: The requested HTTP protocol version is not supported by the server.

HTTP/2

- HTTP/2 is the next version of HTTP and is based on Google's SPDY Protocol (originally designed to speed up the serving of web pages). It was released in 2015 by the Internet Engineering Task Force (IETF).
- It is important to note that HTTP/2 is not a replacement for HTTP.
- It is merely an extension, with all the core concepts such as HTTP methods, Status Codes, URIs, and Header Fields remaining the same.

https://www.thewebmaster.com/hosting/2015/dec/14/what-is-http2-and-how-does-it-compare-to-http1-1/

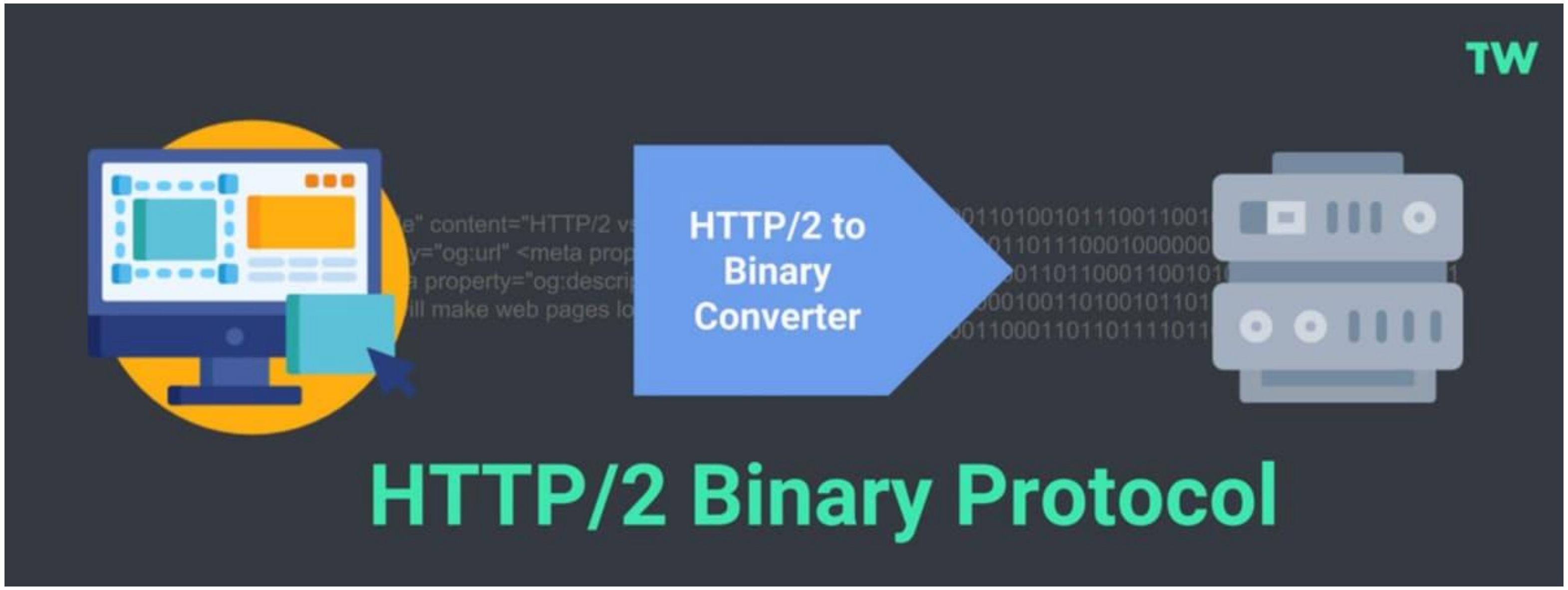
HTTP/2

- The key differences HTTP/2 has to HTTP/1.x are as follows:
  - It is binary, instead of textual
  - It is fully multiplexed, instead of ordered and blocking
  - It can use one connection for parallelism
  - It uses header compression to reduce overhead
  - It allows Server Pushing to add responses proactively into the Browser cache.

https://www.thewebmaster.com/hosting/2015/dec/14/what-is-http2-and-how-does-it-compare-to-http1-1/

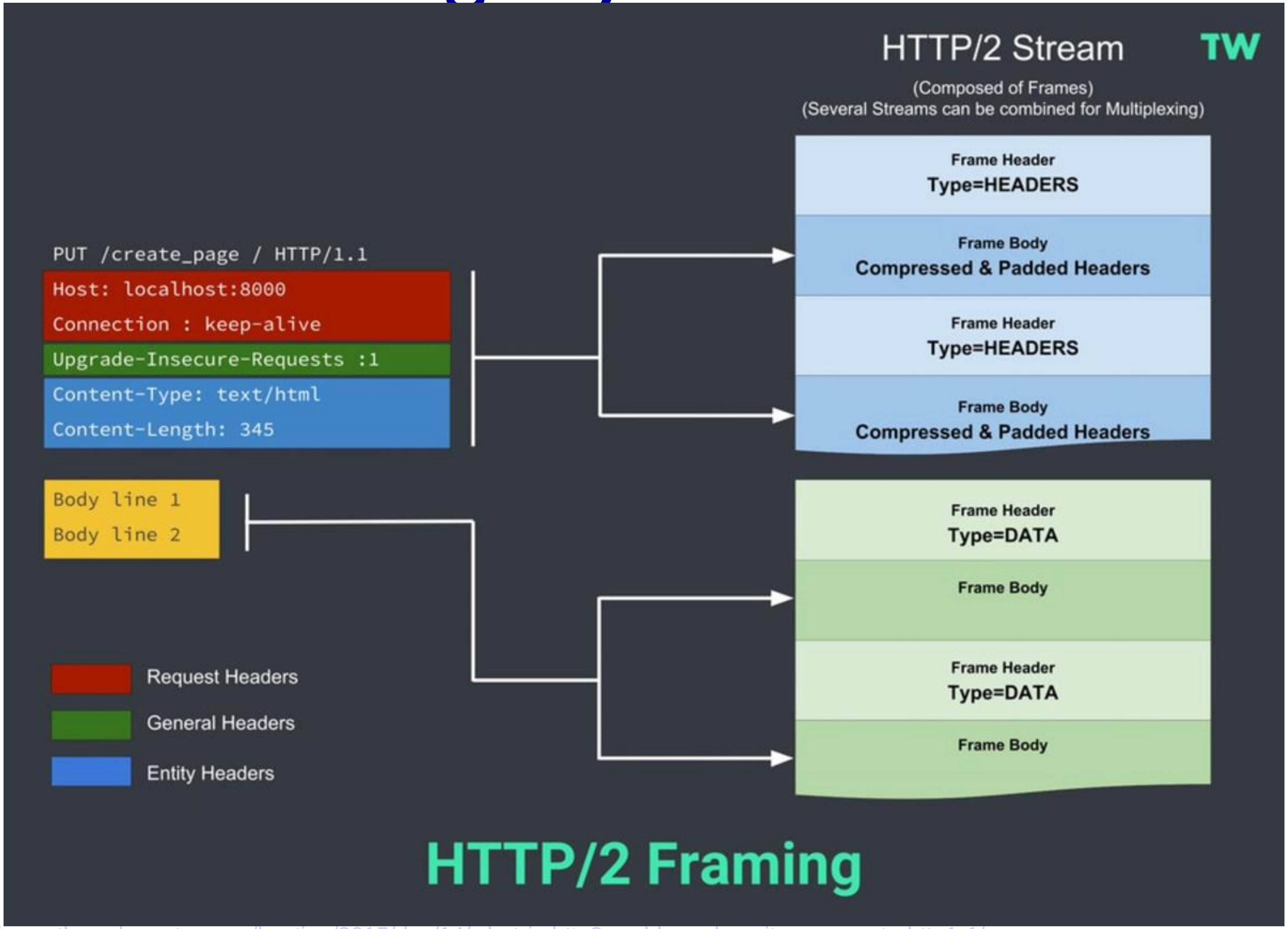
Textual vs. Binary

- HTTP1.x uses text-based commands to complete HTTP requests. If you were to view one of these requests they would be perfectly readable (to a system admin at least).
- HTTP2, on the other hand, uses binary commands (1s and 0s) to complete HTTP requests. It needs to be converted back from binary to read the request.



https://www.thewebmaster.com/hosting/2015/dec/14/what-is-http2-and-how-does-it-compare-to-http1-1/

The Framing Layer



```
PUT /create_page / HTTP/1.1
Host: localhost:8000
Connection : keep-alive
Upgrade-Insecure-Requests :1
Content-Type: text/html
Content-Length: 345

Body line 1
Body line 2
```

HTTP/2 Stream
TW
(Composed of Frames)
(Several Streams can be combined for Multiplexing)
Frame Header Type=HEADERS
Frame Body Compressed & Padded Headers
Frame Header Type=HEADERS
Frame Body Compressed & Padded Headers
Frame Header Type=DATA
Frame Body
Frame Header Type=DATA
Frame Body
🟥 Request Headers
🟩 General Headers
🟦 Entity Headers
HTTP/2 Framing

https://www.thewebmaster.com/hosting/2015/dec/14/what-is-http2-and-how-does-it-compare-to-http1-1/

Multiplexing and Concurrency

- HTTP/1.0 allowed just one request to be made at a time.
- HTTP/1.1 allowed multiple requests, but the number of requests was limited to around 6 or 8, depending on the browser.

Browser	Max Parallel Connections Per Host
IE 9	6
IE 10	8
Firefox 4+	6
Opera 11+	6
Chrome 4+	6
Safari	4

https://www.thewebmaster.com/hosting/2015/dec/14/what-is-http2-and-how-does-it-compare-to-http1-1/

Multiplexing and Concurrency

- **Domain Sharding**
  - With HTTP/1.x if a user wanted to make multiple parallel requests to improve performance, they would need to use a technique such as Domain Sharding.
  - This is where a user would use a subdomain (or multiple subdomains) for assets such as images, CSS files, and JavaScript files so that they could make two or three times the number of connections to speed up the download of files.
- **Head-of-line Blocking**
  - It allows full request and response multiplexing, by allowing the client and server to break down an HTTP message into independent frames, interleave them, and then reassemble them on the other end.
  - Furthermore, it only uses a single TCP connection to do all this.

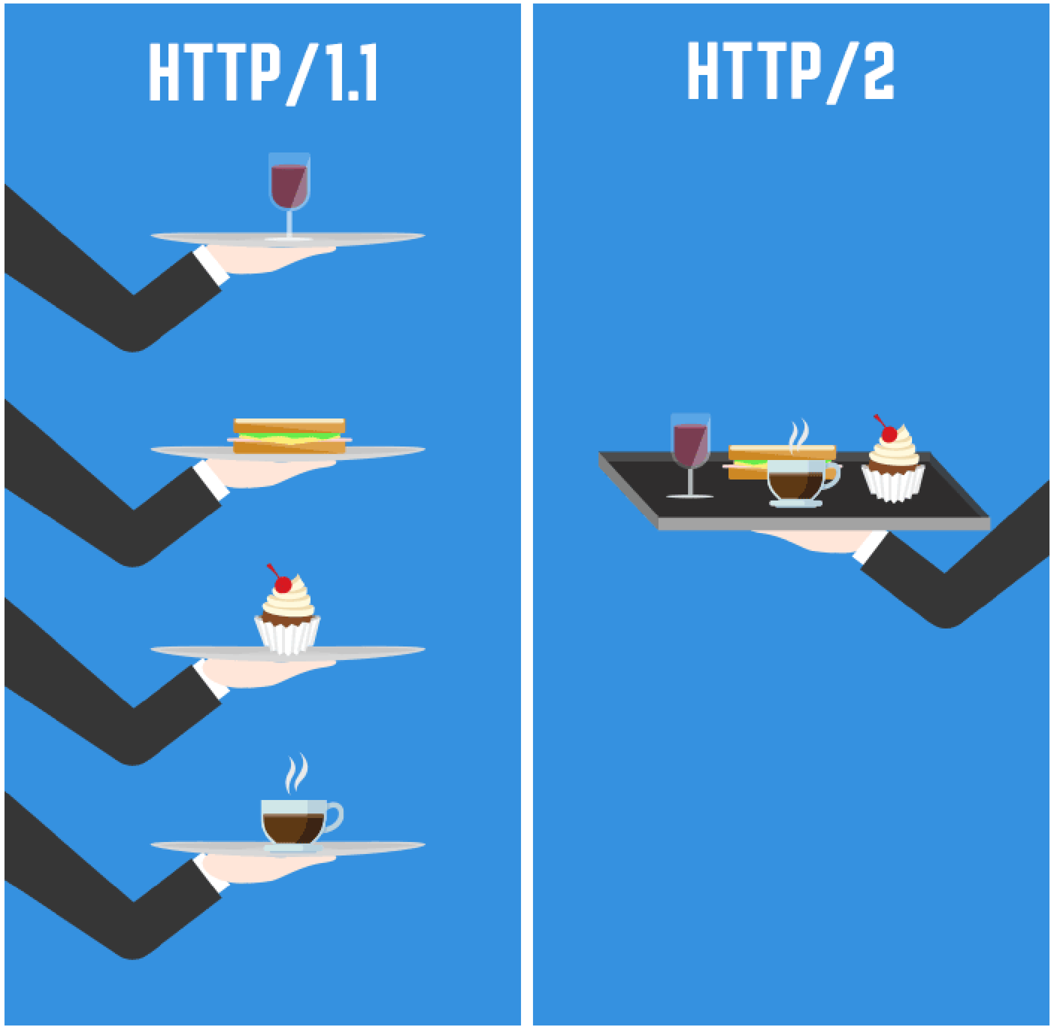
https://www.thewebmaster.com/hosting/2015/dec/14/what-is-http2-and-how-does-it-compare-to-http1-1/

HTTP methods: 1.0 vs 1.1

Method	1.0	1.1
GET	O	O
POST	O	O
HEAD	O	O
OPTIONS		O
PUT	Δ	O
DELETE	Δ	O
TRACE		O
CONNECT		O

HTTP1.0: RFC1945
HTTP1.1: RFC2626
Δ: Optional.

HTTP/1.1 vs 2.0



css-tricks.com

References

- Kurose, Ross, Computer Networking: A Top Down Approach, 6th Ed, Chap. 2. (Application Layer).
- Stallings, Data and Computer Communications, 10th Ed. Chap. 24.
- Forouzan, Data Communications and Networking, 5th Ed. Chap. 25.
- Tanenbaum & Wetherall, Computer Networks, 5th Ed., Chap. Chap. 7.