

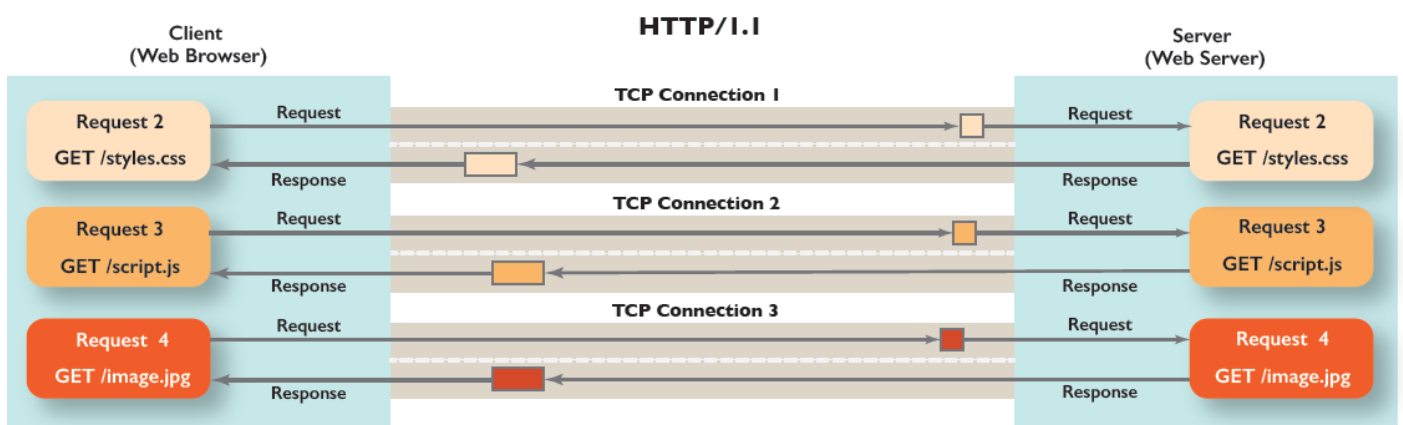
HTTP



Hyper Text Transfer Protocol

HTTP (Hypertext Transfer Protocol) is a set of rules that runs on top of the TCP/IP suite of protocols and defines how files are to be transferred between clients and servers on the world wide web.

HTTP/1.1



CREATION:

HTTP/1.1, the first standardized version of HTTP, was introduced in 1997. It presented significant performance optimizations (over HTTP/0.9 and HTTP/1.0) and transformed the way requests and responses were exchanged between clients and servers.

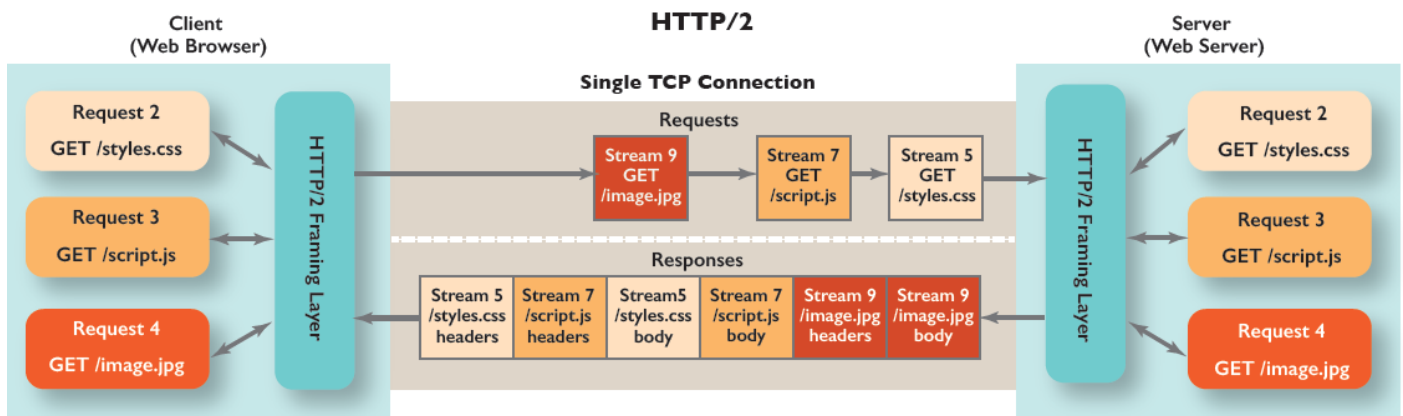
FEATURES:

- It was no longer required for each connection to be terminated immediately after every request was served with a response; instead, with the keep-alive header, it was possible to have persistent connections. It allowed multiple requests/responses per TCP connection.
- HTTP/1.1 provided support for chunk transfers that allowed streaming of content dynamically as chunks and for additional headers to be sent after the message body. This enhancement was particularly useful in cases where values of a field remained unknown until the content had been produced.

Other features that reinforced its stability were introduced such as:

1. pipelining (the second request is sent before the response to the first is adequately served)
2. content negotiation (an exchange between client and server to determine the media type, it also provides the provision to serve different versions of a resource at the same URI)
3. cache control (used to specify caching policies in both requests and responses)

HTTP/2



CREATION:

At the beginning of 2010, Google introduced an experimental protocol, SPDY, which supported multiplexing (multiple requests/responses sent and received asynchronously over a single TCP connection) but as it gained traction IETF's HTTP Working Group came up with HTTP/2 in 2015, which is based on the SPDY protocol.

FEATURES:

- It introduces the concept of a server push where the server anticipates the resources that will be required by the client and pushes them prior to the client making requests. The client retains the authority to deny the server push; however, in most cases, this feature adds a lot of efficiency to the process.
- Introduces the concept of multiplexing that interleaves the requests and responses without head-of-line blocking and does so over a single TCP connection.

- It is a binary protocol i.e., only binary commands in the form of 0s and 1s are transmitted over the wire. The binary framing layer divides the message into frames that are segregated based on their type – Data or Header. This feature greatly increases efficiency in terms of security, compression and multiplexing.

DIFFERENCE BETWEEN HTTP/1.1 AND HTTP/2.0

HTTP/1.1	HTTP/2
It works on the textual format.	It works on the binary protocol.
There is head of line blocking that blocks all the requests behind it until it doesn't get its all resources.	It allows multiplexing so one TCP connection is required for multiple requests.
It uses requests resource Inlining for use getting multiple pages	It uses PUSH frame by server that collects all multiple pages
It compresses data by itself.	It uses HPACK for data compression.

REQUEST:

HTTP/2 Inside: binary

HTTP/2.0 request: 00 00 9D 01 25 00 00 00 01 00 00 00 00 B6 41 8A ...% . . .A.
90 B4 9D 7A A6 35 5E 57 21 E9 82 00 84 B9 58 D3 ...z.S^W!...X.
3F 85 61 09 1A 6D 47 87 53 03 2A 2F 2A 50 8E 9B ?.a..mG.S.*/P..
D9 AB FA 52 42 CB 40 D2 5F A5 11 21 27 51 8B 2D ...RB.@...!'Q.-
4B 70 DD F4 5A BE FB 40 05 DE 7A DA D0 7F 66 A2 Kp..Z..@..z...f.
81 B0 DA E0 53 FA D0 32 1A A4 9D 13 FD A9 92 A4S..2.....
96 85 34 0C 8A 6A DC A7 E2 81 04 41 04 4D FF 6A ..4..j....A.M.j
43 5D 74 17 91 63 CC 64 B0 DB 2E AE CB 8A 7F 59 C]t..c.d.....Y
B1 EF D1 9F E9 4A 0D D4 AA 62 29 3A 9F FB 52 F4J...b):..R.
F6 1E 92 B0 D3 AB 81 71 36 17 97 02 9B 87 28 ECq6.....(.
33 0D B2 EA EC B9

HTTP/1.1 request:

```
GET / HTTP/1.1
Host: demo.nginx.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Chrome/47.0.2518.0
```

HTTP/2 uses HPACK header compression algorithm that is resilient to attacks like CRIME and utilizes static Huffman encoding.

CONCLUSION:

HTTP2 is much faster and more reliable than HTTP1. HTTP1 loads a single request for every TCP connection, while HTTP2 avoids network delay by using multiplexing.

HTTP is a network delay sensitive protocol in the sense that if there is less network delay, then the page loads faster.