HTTP1.1

1)Introduced in 1997

2)It supports Connection reuse

3)TCP connection there could multiple requests and responses. It request several resources from the server.

4)Internet Landscape was constantly changing with websites becoming more dynamic and heavy

5)Features like CORS, Keep alive

6)Head of Line Blocking

7) ASCII /Plaintext, No compression, Header field names - case insensitive are Header Format

8) ASCII /Plaintext, Header Section, Message Body, Separated by empty line are Message Format.

9) HTTP1.1 is 1) Transport Layer Security(TLS)

TLS 1.2

Previously → SSL

2) Hypertext Transfer Protocol Secure (HTTPS)

3) Bi-directional encryption between client and server

10) Connection Establishment is Client initiates TCP connection, Multiple simultaneous TCP connections allowed

11)Example:SMS

Client 🡪 3 TCP Connection <--Server

Request: Hello

Response: Hello

Request: How are you?

Response: Fine..

HTTP2

--------------------------------------------------------------------------------------------------------------------------------------

1)Introduced in 2015

2)One secured TCP connection is setup in which HTTP Request are tranferred

3)Uses Multiplexing, single TCP connection resources.

4) HPACK compression of header into block, Breaks header block into frames for transmission, Huffman encoding + Static table of commonly used header fields + Dynamic table with fields specific to the session ,All field names lower case and request line is split into separate pseudo-header fields :method, :scheme, :authority, and :path are Header Format

5) Frames, Headers Frame, Data Frame – Payload are Message Format

6) HTTP2 is Same as in HTTP/1.1. The optionally runs over TLS for encrypted connection

7) Connection Establishment is Client initiates TCP connection, Single connection per host-port pair for each server, Multiple streams can be run.

8) Maximum connection limit per domain can be ignored. HTTP/2 compresses header data and sends it in a concise, binary format. Better than the plain text format used previously. Less need for popular HTTP 1.1 optimizations.

9) Example: SMS

Client 🡪 3 TCP Connection <--Server

Request: Hello

Request: How are you?

Response: Hello

Response: Fine..