1. **Difference between HTTP 1.1 and HTTP 2**

The Hypertext transfer protocol (HTTP) is an application layer protocol standard. Since 1997 with the release of HTTP 1.1, there have been few versions to the protocol until 2015 when HTTP 2.0 was introduced with technical changes to improve the speeding up and reducing round trip time and become a more efficient web protocol.

HTTP 2 was derived from the Google’s experimental SPDY protocol and thus laid a foundation for standard requests such as GET, HEAD, PUT and POST.

The main goals of developing HTTP 2 were:

**Multiplexing:**

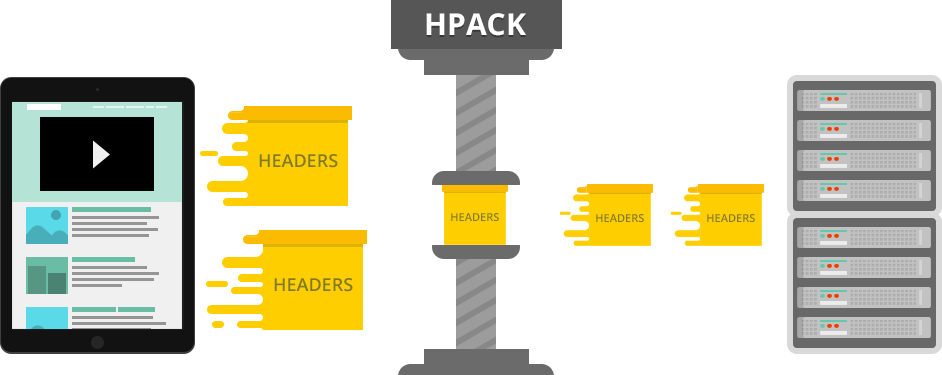
HTTP 2 is multiplexed, which is, it can initiate multiple requests in parallel over a single TCP connection. As a result, web pages containing several elements are delivered over one TCP connection. These capabilities solve the head-of-line blocking problem in HTTP 1.1, in which a packet at the front of the line blocks others from being transmitted. This reduces additional round trip time making the website load faster.



**Header compression**

In HTTP 1.1, header fields are not compressed. As web pages have grown to require dozens to hundreds of requests, the redundant header fields in these requests unnecessarily consume bandwidth, measurably increasing latency.

HTTP 2 uses header compression with the help of HPACK, to reduce the overhead caused by TCP’s slow-start mechanism. HPACK compresses the individual value of each header before it is transferred to the server, which then looks up the encoded information in a list of previously transferred header values to reconstruct the full header information.



**Binary Protocols**

Binary protocols consume less bandwidth, are more efficiently parsed and are less error-prone than the textual protocols used by HTTP 1.1. Additionally, they can better handle elements such as whitespace, capitalization and line endings.

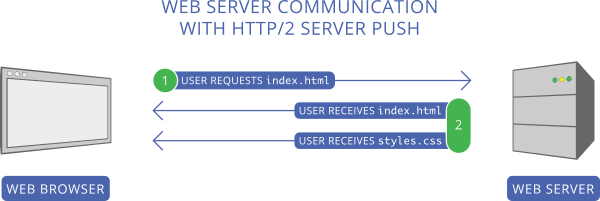
Browsers using HTTP 2 implementation will convert the same text commands into binary before transmitting it over the network.

HTTP 1.1 uses text commands to complete request response cycles while HTTP 2 uses binary commands in 1s and 0s to encode a text commands before transmitting over the network.



**Server Push**

HTTP 2 servers push likely-to-be-used resources into a browser’s cache, even before they’re requested. This allows browsers to display content without additional request cycles. It is for the most part, a performance technique that can be helpful in loading resources pre-emptively.



1. **Difference between GET and POST**

HTTP has set of standard requests such as GET, POST, PUT and HEAD and the two most important methods are GET and POST.

GET is used to request data from a specified source while POST is used to send data to the server to create a resource.

|  |  |
| --- | --- |
| GET | POST |
| GET requests can be cached | POST requests cannot be cached |
| GET requests remain in the browser history | POST requests do not remain in the browser history |
| They can be bookmarked | They cannot be bookmarked |
| They have length restrictions (max = 2048 characters) | They don’t have length restrictions |
| Only ASCII characters allowed | No restrictions. Binary data is allowed |
| Less secure as the data is sent through URL | Safer than GET as parameters are not stored in browser history. |
| Data is visible through URL | Data is not visible in the URL |
|  |  |