|  |  |
| --- | --- |
| **HTTP 1.1(1997)** | **HTTP 2(2015)** |
| It uses text format | It uses binary format |
| **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. | **Multiplexing** – HTTP/2 is multiplexed, i.e., 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. |
| It is not using header compression | HTTP/2 uses header compression to reduce the overhead caused by TCP’s [slow-start](https://en.wikipedia.org/wiki/TCP_congestion_control#Slow_start) mechanism. |
| HTTP/1.1 requires one TCP stream for each concurrent request | HTTP/2 allows a single TCP stream to handle multiple concurrent requests. |
| It won’t support prioritization and server push. | provides support for functionality such as response prioritization, and server push. |
| It is mature, battle-hardened transport layer | It won’t support by default |
| HTTP/1.1 is the duplication of data across requests (cookies and other headers). Too many requests means too much redundant data, which would impact performance | Too many request will also process and it won’t impact the performance. |
| the image assets keep loading for a longer time one after another (typical to pipelining in HTTP/1.1) to complete the full image | Once the first few assets start loading over HTTP/2, the following assets are loaded very quickly |
| **Flow:** | **Flow:** |