|  |  |
| --- | --- |
| HTTP1.1 | HTTP2 |
| 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 it’s 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 |
| Developed by Timothy Berners-Lee in 1989 as a communication standard for the World Wide Web | Developed primarily at Google with the intention of reducing web page load |
| As it which keeps all requests and responses in plain text format | It uses the binary framing layer to encapsulate all messages in binary format |
| A contrast that is at the root of the practical differences between the two protocols. | To try new approaches to data delivery not available in HTTP/1.1, |
| Take care of this problem by introducing persistent connections and pipelining | The binary framing layer encodes requests/responses and cuts them up into smaller packets of information |
| It must make use of multiple TCP connections to lessen the effect of HOL blocking | Establishes a single connection object between the two machines |
| To accomplish this, HTTP/1.1 has a different technique called resource inlining, wherein the server includes the required source within the HTML page in response to the initial GET request. | As HTTP/2 supports multiple simultaneous responses to the client’s initial GET request, the server provides the required resource along with the requested HTML page |