1. Difference between HTTP/1.1 and HTTP/2

|  |  |
| --- | --- |
| **HTTP/1.1** | **HTTP/2** |
| Developed by Timothy Berners-Lee in 1988 as a communication standard for www | Developed by the IETF’s HTTP working group in 2015 |
| A protocol with standardized more than 15 years of extension and this is the first version of HTTP | In particular, for greater performance, much faster and more efficient than HTTP1.1 |
| Which keeps all request and responses in plain text format | It uses the binary framing layer to encapsulate all messages in binary format |
| A plain text protocol, which is easy to readable, understandable for human and debug but can be less efficient due to parsing requirements | A binary protocol, which is more efficient to parse by machines. While it’s not as easy for human readable, however it’s faster to process |
| Simpler to implement due to its text- based nature | More complex to implement due to binary framing, multiplexing and others. |
| A single error in a response can potentially disrupt the entire connection | A binary farming that allows for better error handling |
| Flow controls is handled at the TCP level, which can be less efficient for controlling the rate at which data is transmitted | It includes its own flow control mechanism at the application layer, which can be enables more efficient control |
| Loads resources one after the other. It blocks all the other resources behind it. | In contrast, multiple streams of data once so that no one resource blocks any other. |
| If a browser wants to retrieve multiple resources from a sever, it needs to establish multiple connection, leading to potential latency and inefficiency | Supports multiplexing, which allows multiple requests and responses to be multiplexed over a single connection. This will be reduced latency |
| Here header is not compression, which can lead to overhead, especially for small request/responses | Here header is compression, which significantly reduces the size of headers, saving bandwidth and improving performance. |
| The server cannot push resources to the client without an explicit request. All resources must be requested by the client. So, it will take the timing for complete. | Introduces server push, which allows the server to proactively send resources to the client that it anticipates the client will need based on the initial request. |
| Here request is processed in the order they are received, without any inherent prioritization. | Here will be allow for stream prioritization, which means that more important resources can be higher priority. |
| Browser often open multiple connections to a sever to fetch parallel. | A single connection, which is multiplexed for concurrent request. |
| It compresses data by itself | It uses HPACK for data compression |
| Widely supported and is compatible with virtually all web severs and clients, including older browsers | It is compatible with modern browsers and servers, however old severs and clients may not support it. |