**What is HTTP?**

HTTP stands for Hypertext Transfer Protocol. It is the foundation of the World Wide Web and is used by browsers to load web pages.

A typical example is when your browser sends an HTTP request to a web server after entering an URL. The HTTP command then provides an HTTP response to the web server with the webpage’s contents.

**HTTP 2:**

HTTP/2 is the next version of HTTP and is based on Google’s SPDY Protocol (originally designed to speed up the serving of web pages). It was released in 2015 by the Internet Engineering Task Force (IETF).

HTTP/1.1 and HTTP/2 are two different versions of the Hypertext Transfer Protocol, which is used for communication on the World Wide Web.

**Here are 15 key differences between HTTP/1.1 and HTTP/2:**

|  |  |  |
| --- | --- | --- |
|  | **HTTP/1.1** | **HTTP/2** |
| **Multiplexing** | Uses a single connection per request, which can lead to head-of-line blocking. | Supports multiplexing, allowing multiple requests and responses to be multiplexed over a single connection simultaneously, improving efficiency. |
| **Header Compression** | Headers are not compressed, leading to increased overhead. | Utilizes header compression, reducing the amount of data transmitted and improving performance. |
| **Binary Protocol** | Text-based protocol. | Binary protocol, which is more efficient for machines to parse and reduces the chance of errors. |
| **Prioritization** | Lacks explicit support for request prioritization. | Supports prioritization of requests, allowing more important resources to be fetched first. |
| **Server Push** | Doesn't support server push. | Introduces server push, allowing servers to push resources to the client before they are explicitly requested. |
| **Connection Reuse** | Requires multiple connections for parallelism. | Enables full multiplexing and can reuse a single connection for multiple transactions, reducing latency. |
| **Flow Control** | Lacks flow control mechanisms. | Implements flow control, preventing overwhelming the receiver with data. |
| **Header Size** | Headers are sent with each request and can be large. | Header size is reduced due to header compression and the binary format. |
| **Upgrade Mechanism** | Uses the "Upgrade" header for protocol upgrade. | Requires a different upgrade mechanism, using the "HTTP/2" protocol identifier. |
| **Security** | Doesn't inherently provide security features; relies on additional protocols like HTTPS. | Encourages the use of security (HTTPS) and is designed with security in mind. |
| **Round-Trips** | Requires multiple round-trips to complete a request. | Aims to reduce the number of round-trips needed for communication. |
| **Header Duplication** | Headers are often duplicated across requests. | Header fields are typically not retransmitted, reducing redundancy. |
| **Error Handling** | Errors are often handled in a way that disrupts the connection. | Supports more graceful error handling without disrupting the entire connection. |
| **Backward Compatibility** | Fully backward compatible with HTTP/1.0. | Designed to be backward compatible with HTTP/1.1, allowing for a smooth transition. |
| **Implementation Complexity** | Simpler to implement. | More complex due to features like multiplexing and header compression. |