**Difference between HTTP1.1 vs HTTP2**

|  |  |  |
| --- | --- | --- |
| **Features** | **HTTP/1** | **HTTP/2** |
| **Multiplexing** | Uses a single connection per request, leading to potential head-of-line blocking issues. | Supports multiplexing, allowing multiple requests and responses to be sent in parallel over a single connection. |
| **Header Compression** | Headers are not compressed, leading to higher overhead | Headers are compressed using a more efficient algorithm (HPACK), reducing the amount of data transmitted |
| **Binary Protocol** | Text-based protocol, which can be inefficient for parsing and transmitting | Text-based protocol, which can be inefficient for parsing and transmitting |
| **Prioritization** | No built-in support for request prioritization | Supports stream prioritization, allowing more important resources to be fetched first |
| **Server Push** | No support for server push. Clients must explicitly request each resource | Allows servers to push resources to the client proactively, improving page load times. |
| **Connection Reuse** | Requires multiple connections for parallelism, leading to higher latency | Enables multiple streams within a single connection, reducing latency |
| **TLS/SSL Usage** | TLS/SSL is optional | Encourages the use of TLS/SSL for enhanced security. Many implementations require it. |
| **Header Overhead** | Each request carries redundant headers, leading to increased overhead. | Header compression reduces redundant information, decreasing overhead. |
| **Round-Trip Optimization** | Requires multiple round trips to complete a single request. | Aims to reduce latency by allowing multiple requests to be handled in a single round trip |
| **Flow Control** | Lacks flow control mechanisms. | Implements flow control to prevent overwhelming the receiver with data. |
| **Error Handling** | Errors can disrupt the entire connection, affecting all requests. | Errors are stream-specific, allowing unaffected streams to continue. |
| **Backward Compatibility** | Designed for backward compatibility with HTTP/0.9 and 1.0. | Not as backward compatible due to the significant protocol changes. |
| **Header Size Limit** | No limit on header size, leading to potential security vulnerabilities (HTTP Smuggling). | Implements a header size limit to mitigate security risks. |
| **Implementation Complexity** | Simpler to implement | More complex due to features like multiplexing and header compression. |
| **Resource Prioritization** | Relies on the order of requests to determine priority. | Supports explicit prioritization of resources. |