|  |  |  |
| --- | --- | --- |
|  | **HTTP1.1** | **HTTP2** |
| **Multiplexing** | It uses a single, serialized connection per request, meaning that only one request can be outstanding at a time on a connection. | It supports multiplexing, allowing multiple requests and responses to be sent in parallel over a single connection. This improves efficiency and reduces latency. |
| **Header Compression** | Headers are not compressed, leading to redundant data being sent with each request and response. | It uses header compression, which significantly reduces the overhead associated with headers, resulting in faster page loading times. |
| **Binary Protocol** | It uses a text-based protocol, which is human-readable but less efficient in terms of parsing and transmission. | It employs a binary protocol, which is more efficient for machines to process and results in faster communication between the client and server. |
| **Resource Prioritization** | All resources are treated equally, and there is no built-in mechanism to prioritize critical resources. | It allows for the prioritization of resources, enabling the server to send more important resources first, improving overall page load times. |
| **Server Push** | The server has no mechanism to push content to the client without a specific request. | It supports server push, allowing the server to send additional resources to the client without waiting for a request, further optimizing page load times. |
| **Connection Reuse** | Requires multiple connections for parallelism, and the overhead of establishing and maintaining connections can impact performance. | Encourages the use of a single, long-lived connection, reducing the overhead associated with opening and closing multiple connections. |
| **Flow Control** | It relies on the sender to slow down if the receiver is overwhelmed, which can lead to suboptimal performance. | Implements flow control at the protocol level, allowing better management of data transmission between the client and server. |
| **Error Handling** | Errors can disrupt the entire connection and require retransmission of data. | It allows for more granular error handling, as errors in one stream do not necessarily affect others. |
| **Header Fields** | Each request and response has its set of headers, leading to redundant information being sent with each message. | Headers are compressed and sent in a binary format, reducing overhead. |
| **Security** | Does not have built-in support for encryption, relying on additional protocols (e.g., HTTPS) for secure communication. | Encourages the use of encryption by making it a requirement in many implementations, enhancing overall security. |