

HTTP1.1 vs HTTP2

The World Wide Web's primary protocol is HTTP (Hypertext Transfer Protocol). It outlines the format and transmission of messages between web servers and browsers. The protocol has existed for more than 20 years, with HTTP 1.1 being the most recent version. It is therefore beginning to look older.

HTTP 2 is the next generation of the HTTP protocol. It was created to alleviate some of the drawbacks of HTTP 1.1, including its inefficiency and the absence of support for some functionality.

✚ HTTP 1.1:- HTTP/1.1 is a stateless protocol. As a result, every request a client sends to a server is distinct from any others that have come before. As a result, the server must send the same headers and other data with each request, which might be inefficient. A connection-oriented method of data transfer is also used by HTTP 1.1. This indicates that each request generates a unique TCP connection. As a result, there may be bottlenecks because the server can only manage so many connections at once.

- **Connection-Oriented:** The HTTP 1.1 protocol is a connection-oriented protocol. This indicates that each request generates a unique TCP connection. Because the server can only manage a limited number of concurrent connections, this can lead to bottlenecks.
- **Header Compression:** HTTP 1.1 enables header compression, which can help to minimize the size of HTTP headers. The amount of data that needs to be sent across the network can be decreased, which can enhance performance.
- **Pipelining:** Multiple requests may be made to the server using the pipelining feature of HTTP 1.1 before the server must respond to any of them. This can increase performance by minimising the amount of time the client must wait for the server to respond to each request.

✚ HTTP 2:- HTTP 2 is a stateful protocol. As a result, the server is able to monitor the status of the client and server connections. Performance may be enhanced by allowing the server to provide resources that the client has previously requested. The multiplexing method of data transport is also used by HTTP 2. As a result, more than one request can be transmitted over a single TCP connection. By limiting the number of connections the server must manage, this can enhance performance.

- **Multiplexing:** Multiplexing is a feature of HTTP 2 that enables the transmission of several requests over a single TCP connection. Because there would be fewer connections for the server to manage, performance may be improved.
- **Header Compression:** In comparison to HTTP 1.1, HTTP 2 has a more effective header compression method. By lowering the volume of data that needs to be transferred over the network, this can further improve performance.
- **Server Push:** With the help of server push in HTTP 2, resources can be sent to clients before they are requested. By cutting down on the length of time the client has to wait for the server to respond to each request, this can enhance performance.

- Connection reuse: The client and server can use the same connection for a number of queries with HTTP 2. As a result, the time it takes to create a new connection will be less, which will increase performance.

Over HTTP/1.1, HTTP/2 has a variety of benefits, including:

Enhanced performance: Thanks to its support for multiplexing, header compression, server push, and connection efficiency, HTTP/2 is much quicker than HTTP/1.1.

Reduced latency: By transmitting several requests and responses over a single TCP connection, HTTP/2 can minimise latency. This can enhance the user experience by accelerating the loading of web pages.

Enhanced reliability: Thanks to its more advanced header compression mechanism, HTTP/2 is more dependable than HTTP/1.1. This may assist in lowering the frequency of data transfer errors.

Feature	HTTP/1.1	HTTP/2
Data transfer	Data is transferred in separate requests and responses.	Data is transferred in a single stream, which allows for multiple requests and responses to be sent simultaneously.
Header compression	Headers are compressed using a simple algorithm.	Headers are compressed using a more sophisticated algorithm, which results in smaller header sizes.
Server push	Not supported.	Supported, which allows the server to send resources to the client before the client requests them.
Connection efficiency	Each request and response requires a separate TCP connection.	Multiple requests and responses can be sent over a single TCP connection, which improves connection efficiency.
Overall performance	HTTP/1.1 is slower than HTTP/2.	HTTP/2 is significantly faster than HTTP/1.1.