**Benefits of HTTP 1.1**

There are several benefits of HTTP 1.1 compared to the earlier version HTTP 1.0.

* **Persistent Connections**: One of the most significant benefits of HTTP 1.1 is the support for persistent connections, allowing multiple requests to be sent over the same connection. This reduces the overhead of establishing new connections for each request and results in improved performance.
* **Improved Caching**: With several features that improve caching, such as the ability to send conditional GET requests and support for caching negotiation, clients can cache responses and avoid unnecessary network traffic.
* **More Efficient Request and Response Handling**: With PUT, DELETE, and OPTIONS, fine-grained control over resources on the server is facilitated. It also supports HTTP pipelining and chunked transfer encoding, thus reducing the latency of sending large payloads.

As technology became more evolved, HTTP 2 was developed. It addressed some of the limitations and inefficiencies of HTTP 1.1.

Moving forward with **HTTP 1.1 vs 2.0**, let’s now discuss HTTP 2.

**What is HTTP 2?**

HTTP 2 is the second major version of the HTTP network protocol used for transmitting data over the Internet. It was developed to improve the performance and efficiency of web applications by reducing the amount of data sent over the wire. This reduces the number of round trips required to load a webpage and allows for a more efficient application of resources.

But why is http 2.0 vs. 1.1 a debated topic, and what makes the former more efficient than its previous version?

**Why HTTP 2 Replaced HTTP 1.1?**

As the technology developed and the Internet evolved, HTTP 1.1 soon became outdated. This was the reason why [HTTP 2 was developed](https://en.wikipedia.org/wiki/HTTP/2_Server_Push). However, some other reasons include the following:

**Request Multiplexing**

In HTTP 1.1, each request and response was sent over a separate connection. For each request, a new connection had to be established, resulting in a high number of round trips required to load a webpage. HTTP 2 came with multiplexing of requests.

**Header Compression**

HTTP 1.1 did not have built-in support for header compression, so headers were sent in plain text, which meant that they took up a significant amount of bandwidth. As a result, while sending large numbers of headers or when sending headers over slow connections, problems were faced.

These are the major reasons why HTTP 1.1 was replaced. However, there are other differences between the two versions. So first, let us compare both of them.

**HTTP 2.0 vs. 1.1**

These are some of the parameters based on which we will compare both the HTTP versions.

**Predicting Resource Requests**

In HTTP/1.1, the client-server initiates all requests for resources such as images, stylesheets, and scripts. The server can only respond to requests that it receives. This means that the client must first request the HTML of a web page, parse it, and then make additional requests for any additional resources it needs to render the page. This results in delayed page load times.

Contrary to this, HTTP 2 allows for server push, so the server proactively pushes resources to the client without the client needing to request them. This speeds up page load times as the client starts processing and rendering resources as soon as they are received.

**Buffer Overflow**

In HTTP 1.1, a buffer overflow can occur when a client sends a request with a header that is larger than the server’s buffer size. This can cause the server to crash or become unresponsive. To prevent buffer overflow, servers typically have a maximum buffer size for incoming requests and reject any requests exceeding this limit.

HTTP 2, on the other hand, uses a more sophisticated approach to prevent buffer overflow. It uses a flow control mechanism that allows the server to send data to the client in small chunks rather than sending all the data at once.

**Multiplexing**

HTTP 2 supports multiplexing, allowing multiple requests and responses to be sent over a single connection simultaneously. This helps to reduce the latency and increase the overall performance of the connection. In contrast, HTTP 1.1 uses a separate connection for each request and response, resulting in increased latency and reduced performance.

**Binary Protocol**

HTTP 1.1 uses plain text to encode and transmit data. Though it is easy for humans to read and understand the data, it can be less efficient than a binary protocol.

HTTP 2 uses a series of binary codes to encode and transmit data rather than plain text. Binary protocols are generally more efficient than text-based protocols because they can transmit data more compactly.

**Which Browser Supports HTTP 2 and HTTP 1.1?**

Most modern web browsers support HTTP 1.1 and  HTTP 2 including [Google Chrome](https://cheapsslweb.com/blog/how-to-fix-err-ssl-version-interference-error-in-chrome), Mozilla Firefox, Apple Safari, Microsoft Edge, and Opera.

However, the HTTP version might vary based on the version of the browser, and some older versions may not support it. So it all comes down to the operating system, as well as the specific configuration of the browser.