**Difference between HTTP1.1 vs HTTP2**

HTTP stands for hypertext transfer protocol & it is used in client-server communication. By using HTTP user sends the request to the server & the server sends the response to the user. There are several stages of development of HTTP but we will focus mainly on HTTP/1.1 which was created in 1997 & the new one is HTTP/2 which was created in 2015.

**HTTP/1.1**

HTTP protocol was developed in 1989 as the common language that enables client and server machines’ interaction. Process steps are as enlisted:

* The client (browser) has to send a request to the server using the method (GET/POST).
* Server responds with the requested resource, for example – image, alongside the status of what it did to the client’s request.

Keep in mind that this is not a one-time process. Such requests and responses needs to be transferred between both these machines until the client receives all the resources, essential to load a web page on the end-user’s (your) screen.

This request-response exchange can be regarded as an IP stack being handled by transfer layer and networking layers before finally reaching to the application layer. Now, let’s see how HTTP/2 handles the same scenario.

This specification defines the protocol referred to as "HTTP/1.1". This protocol includes more stringent requirements than HTTP/1.0 in order to ensure reliable implementation of its features.

**HTTP/2**

HTTP/2 was released at Google as the significant improvement of its predecessor. It was initially modeled after the SPDY protocol and went through significant changes to include features like multiplexing, header compression, and stream prioritization to minimize page load latency. After its release, Google announced that it would not provide support for SPDY in favor of HTTP/2.

The major feature that differentiates HTTP/2 from HTTP/1.1 is the binary framing layer. Unlike HTTP/1.1, HTTP/2 uses a binary framing layer. This layer encapsulates messages – converted to its binary equivalent – while making sure that its HTTP semantics (method details, header information, etc.) remain untamed. This feature of HTTP/2 enables gRPC to use lesser resources.

#### **Benefits:**

* Enhanced encoding mechanism between the socket interface and HTTP API.
* Binary protocols are more efficient to parse
* They are more compact and more efficient over the network.
* Less error-prone than HTTP/1, as HTTP/1 has many “helpers” to deal with whitespace, capitalization, line endings, etc. The more complexity, the more chance of errors.
* More secure, as security concerns associated with textual attacks, such as splitting, are no longer relevant.

## **HTTP/1.1 And HTTP/2 Differences**

**HTTP/1.1**

HTTP protocol was developed in 1989 as the common language that enables client and server machines’ interaction. Process steps are as enlisted:

* The client (browser) has to send a request to the server using the method (GET/POST).
* Server responds with the requested resource, for example – image, alongside the status of what it did to the client’s request.

Keep in mind that this is not a one-time process. Such requests and responses needs to be transferred between both these machines until the client receives all the resources, essential to load a web page on the end-user’s (your) screen.

This request-response exchange can be regarded as an IP stack being handled by transfer layer and networking layers before finally reaching to the application layer. Now, let’s see how HTTP/2 handles the same scenario.

**HTTP/2**

HTTP/2 was released at Google as the significant improvement of its predecessor. It was initially modeled after the SPDY protocol and went through significant changes to include features like multiplexing, header compression, and stream prioritization to minimize page load latency. After its release, Google announced that it would not provide support for SPDY in favor of HTTP/2.

The major feature that differentiates HTTP/2 from HTTP/1.1 is the binary framing layer. Unlike HTTP/1.1, HTTP/2 uses a binary framing layer. This layer encapsulates messages – converted to its binary equivalent – while making sure that its HTTP semantics (method details, header information, etc.) remain untamed. This feature of HTTP/2 enables gRPC to use lesser resources.

## **Conclusion**

The influence and control of HTTP/2 in the cyber world are absolutely inexorable. The core features of HTTP/2 provide greater levels of control that can be used to optimize the web application performance. Certainly, the tech world is rapidly evolving with each passing year, which needs advanced technologies every now and then. HTTP/3 is the upcoming internet protocol developed to fix the shortcomings of its predecessor. However, there is so much left to do, and HTTP/2 is not going away any time soon.