**Difference Between HTTPS 1.1 and HTTPS 2 :**

**Hyper Text Transfer Protocol :**

HTTP stands for hypertext transfer protocol, and it is the basis for almost all web applications. More specifically, HTTP is the method computers and servers use to request and send information. For instance, when someone navigates to cloudflare.com on their laptop, their web browser sends an HTTP request to the Cloudflare servers for the content that appears on the page. Then, Cloudflare servers send HTTP responses with the text, images, and formatting that the browser displays to the user.

**HTTPS/1.1 :**

The first usable version of HTTP was created in 1997. Because it went through several stages of development, this first version of HTTP was called HTTP/1.1. This version is still in use on the web.

**HTTPS 2 :**

In 2015, a new version of HTTP called HTTP/2 was created. HTTP/2 solves several problems that the creators of HTTP/1.1 did not anticipate. In particular, HTTP/2 is much faster and more efficient than HTTP/1.1. One of the ways in which HTTP/2 is faster is in how it prioritizes content during the loading process.

**HTTPS/1.1 Specifications :**

The internet has revolutionized the way we communicate, work, and consume content. At the heart of this digital transformation lies the Hypertext Transfer Protocol (HTTP), which facilitates the exchange of information between a user's web browser and the server hosting the requested content. Over the years, HTTP has evolved, with HTTP/1.1 being the long-standing protocol and HTTP/2 bringing significant improvements to enhance web performance. In this blog, we'll delve into the key differences between HTTP/1.1 and HTTP/2 and how they impact web browsing experience.

* **Serial Request Handling:** In HTTP/1.1, each request is handled serially, meaning that the browser sends one request at a time and waits for the server to respond before sending the next request. This leads to latency issues, as subsequent requests have to wait for the completion of the previous ones.
* **Header Redundancy:** With every request in HTTP/1.1, a set of headers is sent. Since these headers often contain redundant information, it leads to increased overhead and longer load times.
* **Resource Bundling:** To optimize page loading, developers often bundle multiple resources (like CSS and JavaScript files) together. However, this led to inefficient resource management and made it harder to cache or update specific components.

**HTTPS 2 Specifications :**

* **Multiplexing:** One of the most significant improvements in HTTP/2 is multiplexing. It allows multiple requests and responses to be sent and received in parallel over a single connection. This eliminates the bottleneck caused by serial request handling in HTTP/1.1, significantly reducing latency and improving page load times.
* **Header Compression:** HTTP/2 uses HPACK compression to reduce the size of headers before they are sent over the network. This minimizes redundant data and lowers overhead, further enhancing load times.
* **Server Push:** Unlike HTTP/1.1, where the server could only respond to client requests, HTTP/2 introduces server push. This feature enables the server to proactively send resources to the client before they are explicitly requested, improving page rendering speed.
* **Binary Protocol:** HTTP/2 uses a binary protocol instead of the plain text protocol used in HTTP/1.1. While this change might not be directly visible to end-users, it enhances the efficiency of data transfer and reduces parsing complexity.
* **Stream Prioritization:** In HTTP/2, streams can be assigned priorities, allowing more critical resources to be prioritized for quicker delivery. This ensures that essential elements of a page are loaded first, improving the user experience.

**Conclusion**

HTTP/2 represents a significant leap forward in web protocol technology, addressing many of the limitations inherent in HTTP/1.1. The introduction of multiplexing, header compression, server push, and other optimizations has led to faster and more efficient web browsing experiences. While HTTP/1.1 will continue to be in use for some time due to legacy systems, the adoption of HTTP/2 is growing steadily as websites and web applications seek to provide users with a smoother and more responsive online interaction. As we move forward, it's important to keep an eye on further protocol developments and improvements that could shape the future of web communication.