**DIFFERENCE BETWEEN HTTP/1.1 AND HTTP/2 :**

**HTTP/1.1 :**

**HTTP/1.1** relies on plain text for communication between clients (such as web browsers) and servers. Requests and responses are transmitted in a human-readable format, which makes it easy to inspect using tools like browser developer consoles or packet sniffers.

* HTTP/1.1 keeps all requests and responses in plain text format.
* HTTP/1.1 processes requests and responses sequentially, which can lead to blocking and latency.
* HTTP/1.1 requires separate connections for parallel requests, leading to overhead.
* HTTP/1.1 sends headers in plain text with each request and response, resulting in overhead.

**HTTP/2 :**

**HTTP/2** uses a **binary framing layer** to encapsulate messages. This binary format allows for more efficient parsing and transmission of data compared to the plain text format of **HTTP/1.1**.

* [**HTTP/2**, on the other hand, uses a **binary framing layer** to encapsulate messages in binary format while still maintaining HTTP semantics (such as verbs, methods, and headers)](https://www.digitalocean.com/community/tutorials/http-1-1-vs-http-2-what-s-the-difference).
* **HTTP/2** is **fully multiplexed**, allowing multiple requests and responses to be sent concurrently over a single connection. This significantly reduces latency, especially when dealing with mobile platforms and server-intensive graphics or videos.
* **HTTP/2** can use a **single connection** for parallelism, improving efficiency by avoiding the need for multiple connections.
* **HTTP/2** uses **header compression** to reduce this overhead, making communication more efficient  .
* **HTTP/2** introduces **Server Pushing**, allowing the server to proactively send additional resources (such as style sheets or images) to the client’s cache. This optimizes page loading speed.