**Blog about Http 1.1 and Http 2**

**Http:**

Http is a hypertext transfer protocol which helps to communicate client computers and web servers by sending and receiving http requests.

**Http 1.1**

* The **HTTP/1.1** is an application-level protocol used for distributed, collaborative, and hypermedia information systems.
* It was developed by **Tim Berners-Lee** in 1989 as a communication standard for the **World Wide Web**.
* It facilitates the exchange of information between a **client computer** (such as a web browser) and a **local or remote web server**.
* Clients send text-based requests (using methods like **GET** or **POST**) to servers, which respond with data.

In summary, HTTP/1.1 plays a crucial role in enabling web communication, allowing us to access and interact with online resources.

**Http 2:**

* **HTTP/2** (originally named **HTTP/2.0**) represents a significant evolution of the **Hypertext Transfer Protocol (HTTP)** used by the World Wide Web.
* It was derived from the experimental **SPDY protocol**, initially developed by Google.
* HTTP/2 aims to improve page load speed by reducing latency.

In summary, HTTP/2 enhances web performance, making websites load faster and improving the overall browsing experience.

**Let's delve into the technical differences between HTTP/1.1 and HTTP/2.**

|  |  |
| --- | --- |
| **Http 1.1** | **Http 2** |
| **Textual Format –** Itkeeps all requests and responses in plain text format | **Binary formatting -** uses a binary framing layer to encapsulate messages in binary format while still maintaining HTTP semantics **(such as verbs, methods, and headers).** |
| **Multiplexing -** processes requests and responses sequentially, which can lead to latency, especially when dealing with multiple resources | **Multiplexing** – It is fully multiplexed, allowing multiple requests and responses to be interleaved over a single connection.  This significantly reduces latency and improves performance |
| **Header Compression -** It sends headers for each request and response, which can be redundant and add overhead. | **Header Compression –** It uses header compression to reduce this overhead, resulting in more efficient communication between client and server. |
| **Parallelism –** It relies on multiple TCP connections to load resources in parallel. | **Parallelism –** It can use a single connection for parallelism, avoiding the need for multiple connections and reducing network delay. |
|  | **Server Push –** It introduces server push, allowing the server to proactively send additional resources (e.g., stylesheets, images) to the client before they are explicitly requested.  This optimizes page loading and caching. |