Difference between HTTP1.1 & HTTP 2

* One of the significant differences between HTTP/1.1 and HTTP/2 is multiplexing. In HTTP/1.1, each request and response is sent sequentially, which means that if a request is blocked or slow, it can delay subsequent requests.
* HTTP/2, on the other hand, supports multiplexing, allowing multiple requests and responses to be sent and received in parallel over a single connection. This significantly improves performance, especially on high-latency connections.
* HTTP/2 introduces header compression, which reduces overhead by compressing HTTP header fields.
* In HTTP/1.1, headers are sent as plaintext and can consume significant bandwidth, especially for large requests or responses.
* With header compression in HTTP/2, this overhead is reduced, leading to faster communication.
* HTTP/2 supports server push, a feature that allows the server to send resources to the client before they are explicitly requested. This can improve performance by anticipating the client's needs and reducing the number of round trips needed to load a page.
* HTTP/1.1 does not have native support for server push, although similar functionality can be achieved through techniques like inlining resources or using techniques like HTTP pipelining.
* While HTTP/1.1 uses plaintext for communication, HTTP/2 uses a binary protocol.
* This makes it more efficient to parse and reduces the likelihood of errors due to parsing issues. Additionally, the binary format allows for more efficient representation of data, further improving performance.
* HTTP/2 introduces support for stream prioritization, allowing clients to specify the priority of individual resources. This helps optimize resource delivery, ensuring that critical resources are delivered first, improving the overall user experience.
* HTTP/1.1 does not have built-in support for stream prioritization.
* HTTP/2 encourages the reuse of connections for multiple requests, reducing the overhead of establishing and tearing down connections.
* In HTTP/1.1, each request typically requires a separate connection, which can lead to increased latency and resource consumption, especially on high-latency networks.

|  |  |  |
| --- | --- | --- |
| **FEATURE** | **HTTP/1.1** | **HTTP/2** |
| Multiplexing | No multiplexing, requests are sent serially | Supports multiplexing, parallel requests |
| Header Compression | No header compression | Header compression reduces overhead |
| Server Push | Not supported | Supports server push for preloading assets |
| Binary Protocol | Text-based protocol | Binary protocol for more efficient parsing |
| Stream Prioritization | Not supported | Supports prioritization for resource delivery |
| Connection Reuse | Persistent connections optional | Encourages connection reuse |

CONCLUSION:

* HTTP/2 offers significant improvements over HTTP/1.1 in terms of performance, efficiency, and features, making it the preferred choice for modern web applications.
* However, adoption of HTTP/2 may depend on various factors such as server and client support, network conditions, and specific application requirements.