What is HTTP?

* HTTP stands for Hyper Text Transfer Protocol, a protocol used for transmitting data over the internet.
* Communication between end user and web server is done by sending HTTPrequest and receiving HTTP response.

HTTP Request and Response

* Step1 - End user sends an HTTP request to the web.
* Step2 - Web server receives the request.
* Step3 - Server run an application to process the request.
* Step4 - Server returns an HTTP response to the browser.
* Step5 - End user receives the response.

HTTP request / response circle

* If browser requests an HTML page, then server returns an HTML file.
* If browser requests a style sheet, the server returns a cascade style sheet file.
* If browser requests an JPEG image, then server returns a JPEG file.
* If browser requests java script code, then server returns a JS file.
* If browser requests data, then server returns data (in XML or JSON).

HTTP/1.1 and HTTP/2

HTTP/1.1 and HTTP/2 are both protocols for transmitting data over the web, but they have significant difference in terms of performance, efficiency and features.

The main difference between HTTP/1.1 and HTTP/2 are below.

|  |  |
| --- | --- |
| **HTTP/1.1** | **HTTP/2** |
| Multiplexing:   * Each request and response typically require a separate connection. This can lead to inefficiencies, especially when dealing with multiple resources on a webpage. | Multiplexing:   * Supports multiplexing, allowing multiple requests and responses to be sent and received on the same connection simultaneously. This reduces latency and improves efficiency, particularly for websites with many resources. |
| Header Compression:   * Headers are sent as plain text with each request and response, leading to overhead, especially for large headers or multiple requests. | Header Compression:   * Introduces header compression, reducing the overhead associated with headers by compressing them before transmission. This reduces bandwidth usage and speeds up communication between clients and server. |
| Server Push:   * Client must request each resource individually from the server. This can result in suboptimal performance, especially for resources that the server knows the client will need. | Server Push:   * Supports server push, allowing the server to proactively send resources to the client before they are request. This can improve page load times by reducing the number of round trips between the client and server. |
| Binary Protocol:   * Uses plain text for communication between clients and servers, which can be human-readable but less efficient in terms of parsing and processing. | Binary Protocol:   * Uses binary protocol for communication, which is more efficient for both machines to parse and process. While it’s not human readable like HTTP/1.1, it offers better performance and reduces overhead. |
| Connection Management:   * Multiple connections are often used to load resources in parallel, leading to connection management overload. | Connection Management:   * Uses a single, multiplexed connection for all communication, reducing the need for connection management overhead and improving resource loading time. |

Both HTTP/1.1 and HTTP/2 are widely used on the internet today, with HTTP/2 gradually becoming more prevalent as web servers and browsers add support for it.