DIFFERENCE BETWEEN HTTP1.1 VS 2

HTTP

HTTP stands for **Hypertext Transfer Protocol** & it is used in client-server communication.

By using HTTP user sends the request to the server & server sends the response to the user.

Developed by Timothy Berners-Lee in 1989 as a communication standard for the World Wide Web, HTTP is a top-level application protocol that exchanges information between a client computer and a local or remote web server. In this process, a client sends a text-based request to a server by calling a method like **GET** or **POST**. In response, the server sends a resource like an HTML page back to the client.

DRAWBACKS IN HTTP/1.0

In HTTP/1.0, the client had to break and remake the TCP connection with every new request, a costly affair in terms of both time and resources.

HTTP/1.1

* HTTP1.1 was created in 1997.
* Assume the situation when you make a request to the server for the geeksforgeeks.html page & server responds to you as a resource geeksforgeeks.html page. before sending the request and the response there is a TCP connection established between client & server. Return you make a request to the server for image img.jpg & the server gives a response as an image img.jpg. the connection was not lost here after the first request because we add a keep-alive header which is the part of the request so there is an open connection between the server & client. there is a persistent connection which means several requests & responses are merged in a single connection.

GET /index.html HTTP/1.1

Host: www.example.com

This request uses the**GET** method, which asks for data from the host server listed after Host : . In response to this request, the example.com web server returns an HTML page to the requesting client, in addition to any images, stylesheets, or other resources called for in the HTML. Note that not all of the resources are returned to the client in the first call for data. The requests and responses will go back and forth between the server and client until the web browser has received all the resources necessary to render the contents of the HTML page on your screen.

DRAWBACKS IN HTTP1.1

* HTTP/1.1 transfer all the requests & responses in the plain text message form.
* Head of line blocking in which TCP connection is blocked all other requests until the response does not receive. all the information related to the header file is repeated in every request.

HTTP/2

* HTTP/2 is faster is in how it **prioritizes** content during the loading process.
* In HTTP/2, when a [client](https://www.cloudflare.com/learning/serverless/glossary/client-side-vs-server-side/) makes a request for a webpage, the server sends several streams of data to the client at once, instead of sending one thing after another. This method of data delivery is known as multiplexing.

Multiplexing: HTTP/1.1 loads resources one after the other, so if one resource cannot be loaded, it blocks all the other resources behind it. In contrast, HTTP/2 is able to use a single [TCP](https://www.cloudflare.com/learning/ddos/glossary/tcp-ip/) connection to send multiple streams of data at once so that no one resource blocks any other resource. HTTP/2 does this by splitting data into binary-code messages and numbering these messages so that the client knows which stream each binary message belongs to.

Server push: Typically, a server only serves content to a client device if the client asks for it. However, this approach is not always practical for modern webpages, which often involve several dozen separate resources that the client must request. HTTP/2 solves this problem by allowing a server to **"push"** content to a client before the client asks for it. The server also sends a message letting the client know what pushed content to expect

Header compression: HTTP/2 uses a more advanced compression method called **HPACK** that eliminates redundant information in HTTP header packets. This eliminates a few bytes from every HTTP packet. Given the volume of HTTP packets involved in loading even a single webpage, those bytes add up quickly, resulting in faster loading.

