What is HTTP?

HTTP stands for Hyper Text Transfer Protocol. HTTP is the back bone of Web Development.

HTTP uses a request-response model. Clients makes request to servers for resources like pages, images etc servers send back responses with status code like 200(okay), 404(not found), or (500) internal server error. The requested data are responded in the response body.

HTTP defines methods like GET, POST, PUT, DELETE that trigger different operations on the server

Example Get retrieves data, POST submits form data, DELETE removes resources

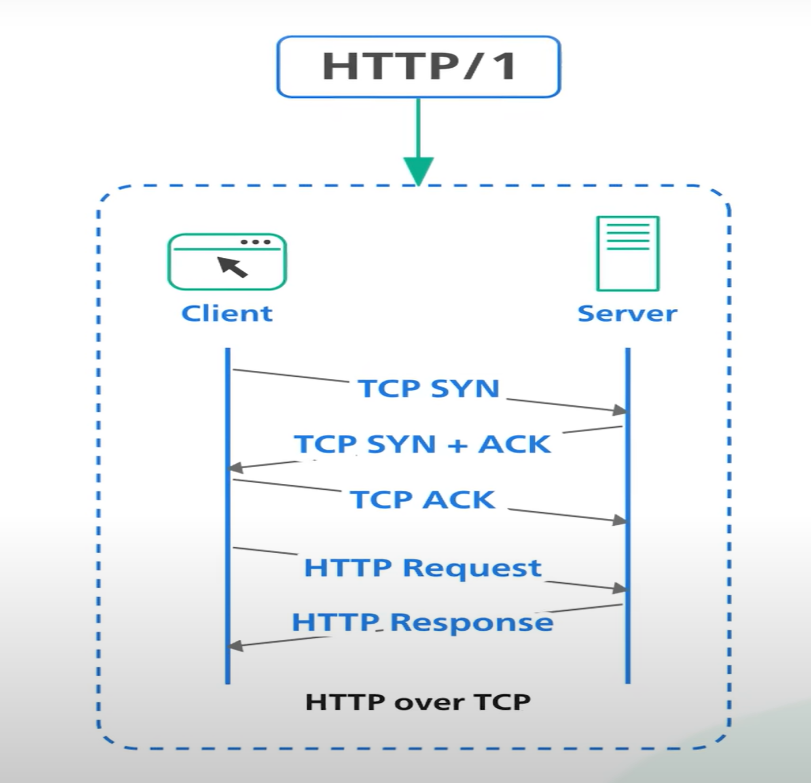
TCP

TCP stands for Transmission control Protocol. TCP is like an intermediator which connects the client and the server.

TCP is the standard that defines how to establish and maintain a network conversation by which applications can exchange Data.

HTTP/1

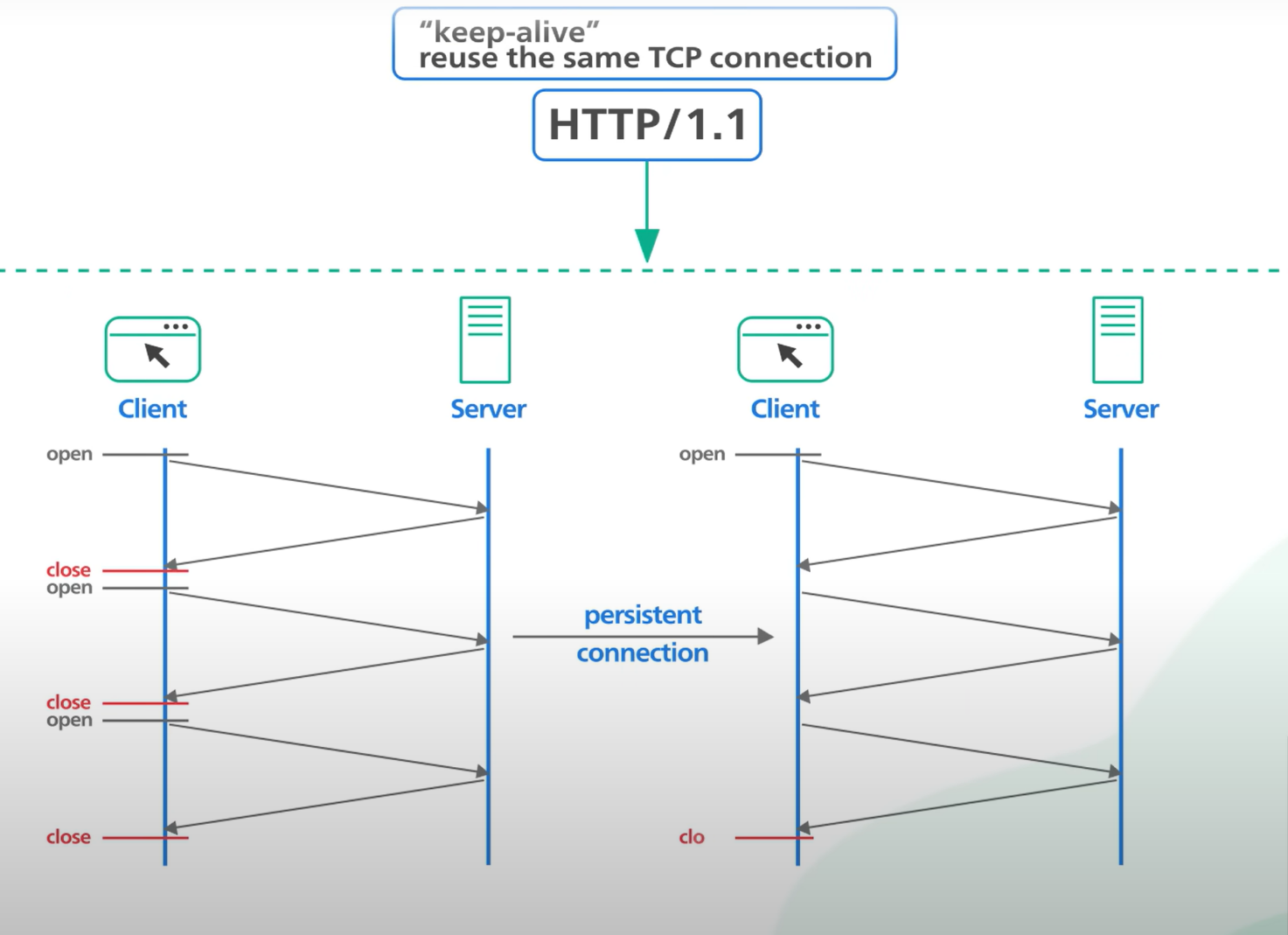
HTTP/1 was developed in 1996. In HTTP/1 every request to the same server requires a separate TCP connection which was expensive and time consuming which led to the development of HTTP/1.1



HTTP/1.1

HTTP/1.1 was introduced in 1997 with **‘keep alive’** Mechanism which reuses TCP connection for more than a single request. This reduces the request latency. In HTTP/1.1 the client need to initiate TCP three way handshake for every request.

HTTP/1.1 also added pipelining this allows the client to send multiple requests before waiting for response. The response is received in the same order as the requests. Many servers did not respond properly for pipelining so many browsers removed pipelining HTTP/1.1 with pipelining also suffers from a issue called “*head of line blocking*” subsequent request on the same connection must wait for the previous requests to complete. IF a request is blocked for reasons like packet loss all request on the same connection are also impacted



To keep loading at an acceptable level, browsers normally keep multiple TCP connections to same servers and send requests to it in parallel

HTTP/2

HTTP/2 was published in 2015. HTTP/2 introduced HTTP streams where multiple streams of requests could be sent to the server on a single TCP connection. Unlike HTTP/1.1 pipelining each stream is Independent from each other it does not send or receive in order.

HTTP/2 solves the headline blocking issue at the application layer but the issue still exists in the transportation layer with TCP. HTTP/2 also introduced a push capability to allow servers to send updates to the clients whenever new data is available without requiring the client to poll

