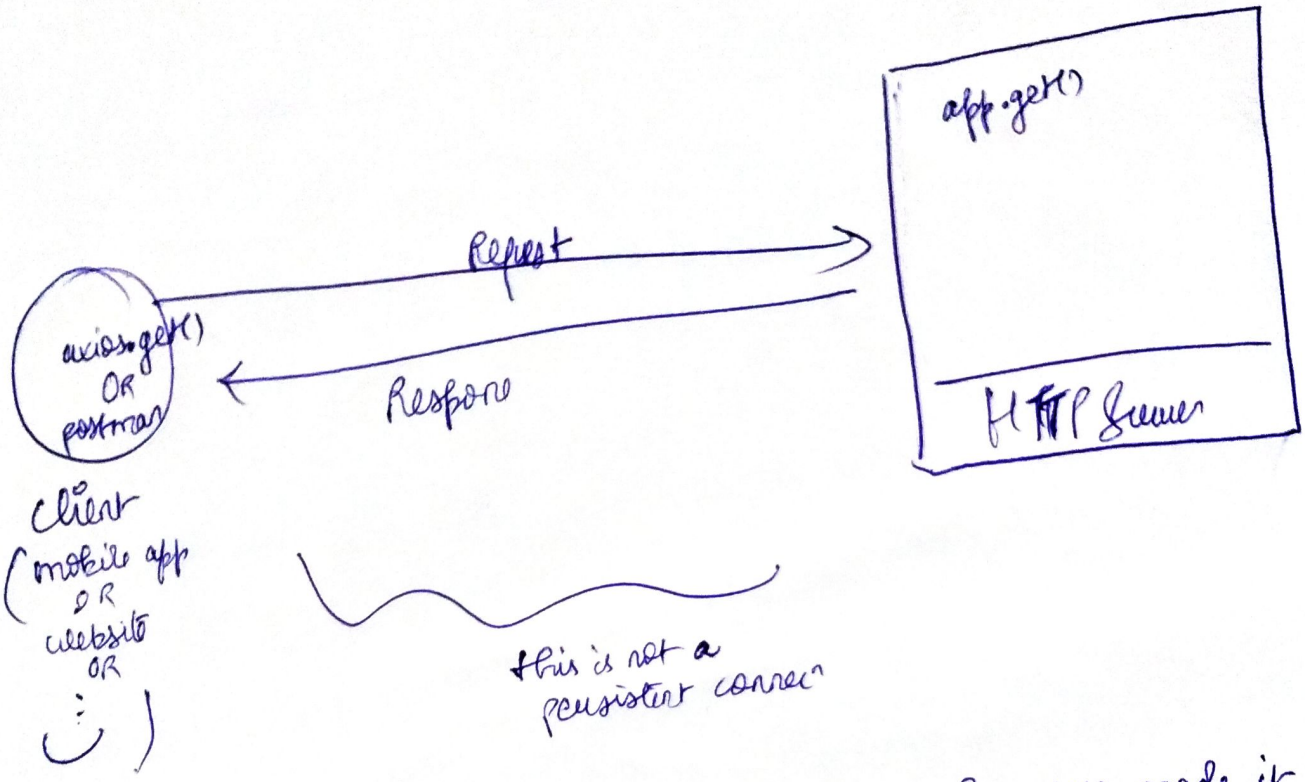
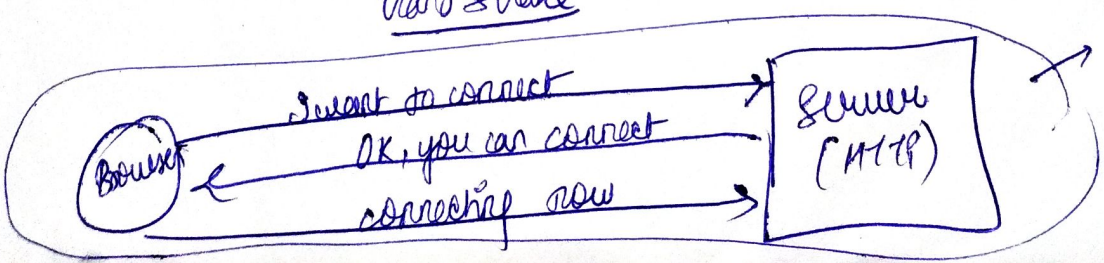
★ Web Sockets

Web sockets provide a way to establish a persistent, full-duplex communication channel over a single TCP connection between the client (typically a web browser) and the server.



Persistent Connection → It means, once connection has been made, it stays there, you can keep multiple requests on same connection & you may or may not get multiple responses, but browser can again & again send requests.

(NOW YOU CAN ARGUE, that this could be done using HTTP server also, by client sending multiple axios.get requests. So, will that be persistent connection?) → ans — NO, because it HTTP request browser undergoes three way handshake

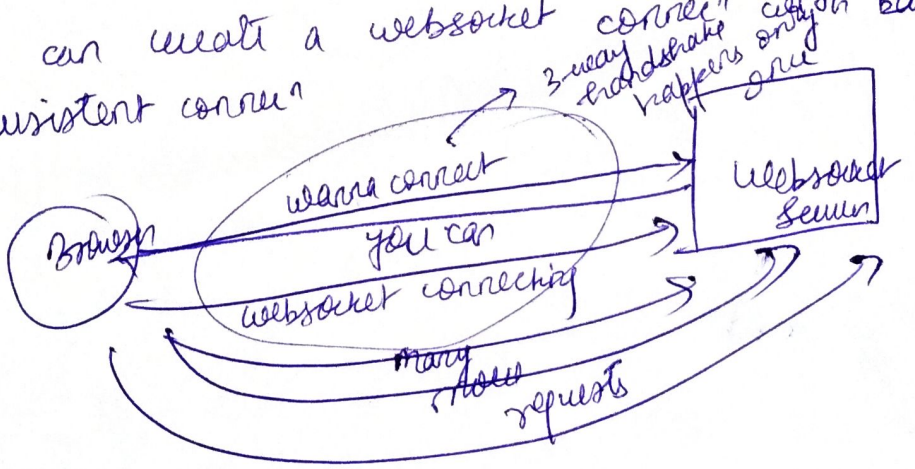


So, when you send axios.get underneath 3 way handshake happens & that three way handshake closes when this request resolved / stops / OR response is sent.

So, every time you do HTTP request from axios, you do 3-way handshake every single time so, this is not persistent connection (so, establishing connection with backend every time) HTTP protocol was written only for this use case - Request-Response model. It was never meant to create persistent connection.

So, In Order to Create Persistent Connection we use other protocols Websockets are one of them, also WebRTC & many more

So, browser can create a websocket connection with backend for persistent connection



↓

Now you can send data incrementally, as much as you want as many times until connection is closed manually

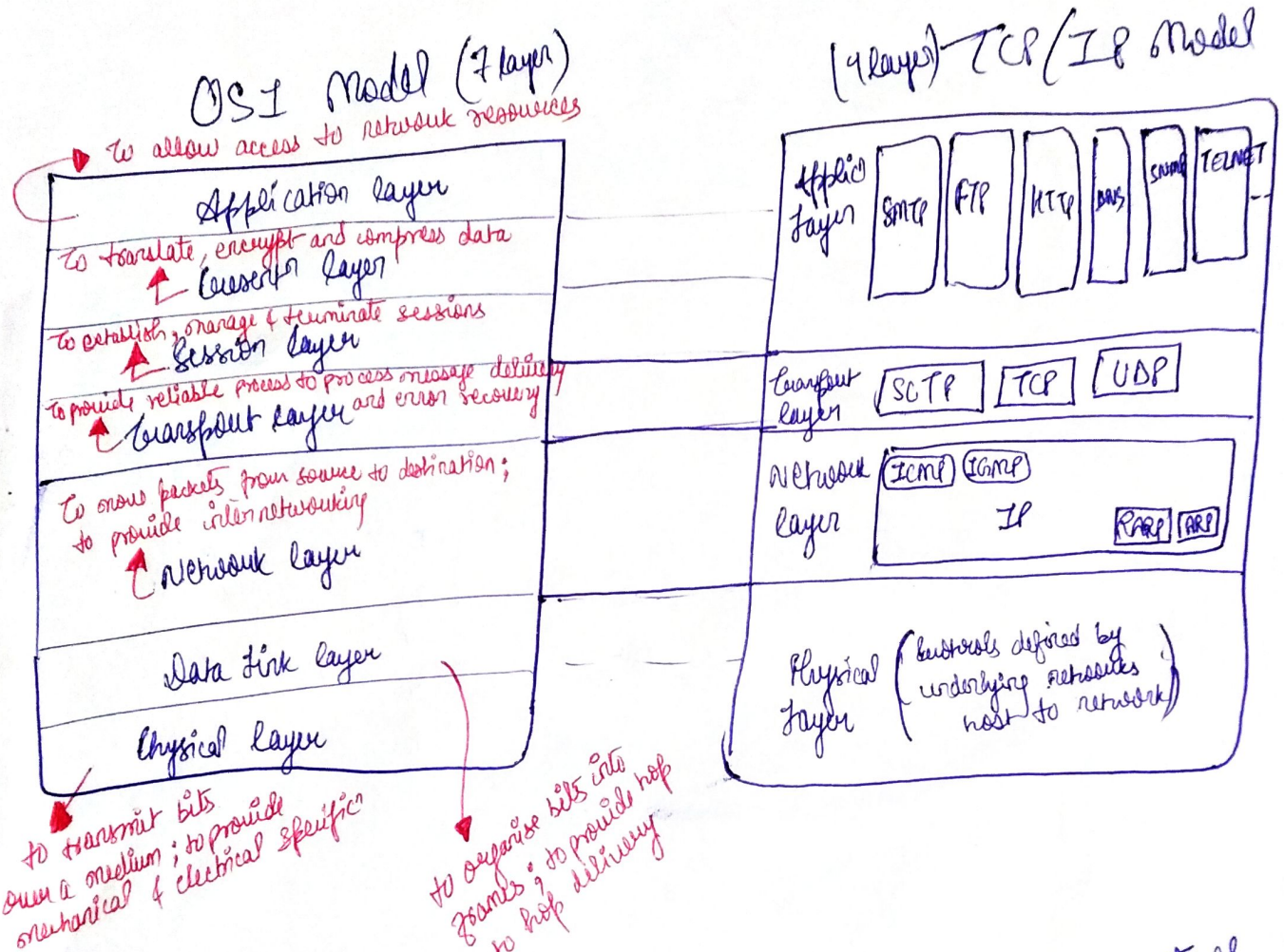
→ Full Duplex → Both way connection
Browser can send multiple requests & server can respond with multiple responses

TCP Connⁿ → Transfer control protocol

So, we have OSI model & TCP/IP model

↳ We have various layers of network.

& TCP & UDP are various protocols



So, HTTP, WebRTC, websockets → Applic layer protocol

i.e. while coding you have to deal with this. So, don't have to deal with making of TCP protocol, it gets made itself

TCP, UDP → Transport layer protocols

So, websocket server & HTTP server → use TCP protocol underhood

WebRTC → can use TCP & UDP any protocol underhood usually uses UDP