

## WEBSOCKETS

Workflow :

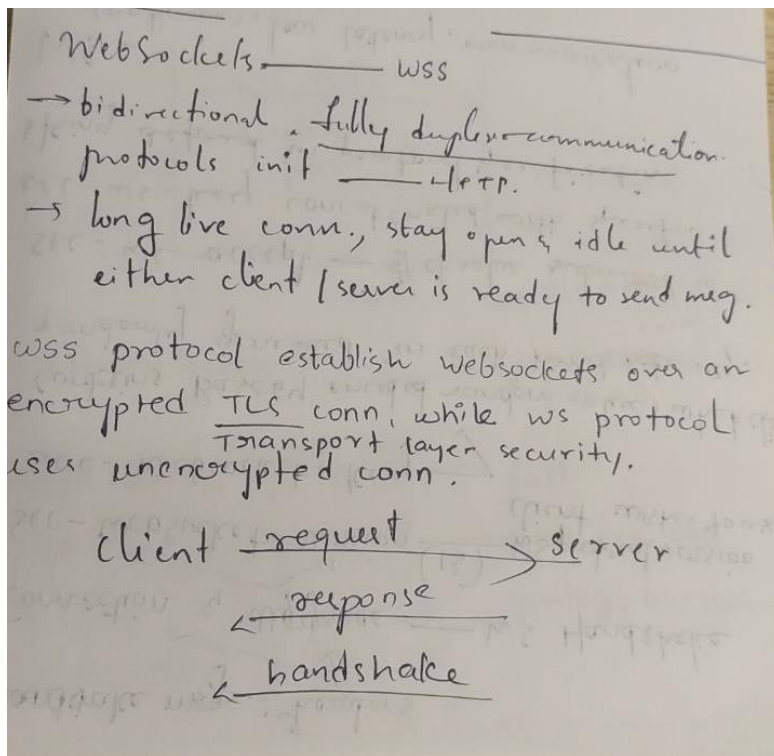
Server → listens for client connections and messages.

Client → connects to server, sends messages, and gets replies.

Send/Receive → server prints client messages and responds.

Close → connection ends when no data is sent.

Real-Time → data flows both ways continuously, like WebSockets.



Handshake msgs.: Headers

Connection & Upgrade — WS Handshake

Sec-WebSocket-Version (13) — WS protocol version  
client wishes to use

Sec-WebSocket-Key —

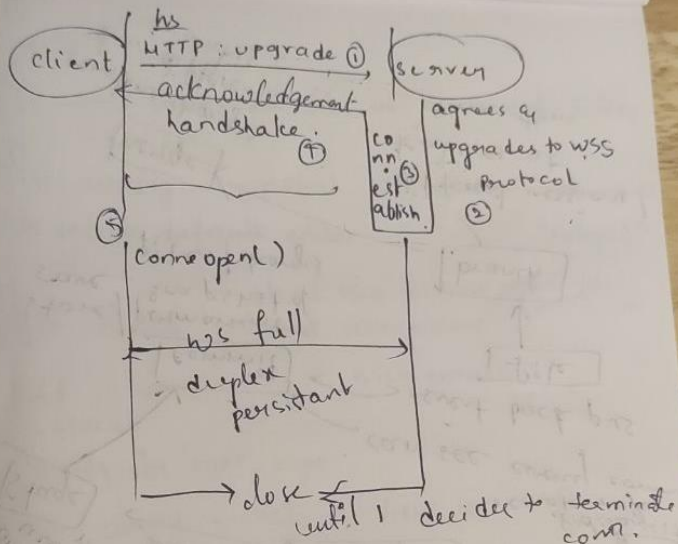
contains Base64 encoded random value, which is  
randomly generated in each handshake req.

Sec-WebSocket-Accept — # of value submitted on  
Sec-WebSocket-Key concatenated with specific  
string defined in protocol specification.

WS —  
continuous, low latency, communication  
model b/w client & server.

enables data exchange - bidirectional -  
no need for repeated HTTP req.

How WS work?



Stream Sockets  
(TCP)

- reliable
- Conn. oriented TCP
- browsing, email, file transfer.

Datagram sockets  
(UDP)

- fast
- conn. less UDP.
- gaming, live stream,