5) TCP v/s UDP Crash Course

TCP and UDP are Layer 4 protocols

Layer 4 is the transport Layer, having ports attached.

A machine has an ip, but on that machine multiple applications might be running. So the port number is used to distinguish between multiple applications running on that machine.

TCP → Transmission Control Protocol

HTTP → layer 7 protocol internally uses

tcp → (Layer 4) protocol

Databases uses tcp all the time, UDP is used in some.

[Pros of TCP]

* Acknowledgement : Tcp attaches some additional info to each packet, like sequence no etc. to make sure the packet has been received.

* Guaranteed Delivery → If the packet is not received. Server will not send an ack. Server will re-request the packet from the sender.

(*** V.Imp)

* Connection based → The client and the server, they need to communicate with one-another and for that, they have to form a Unique connection between them.

[This is required to hold a stateful connection.]

[*** Makes it slow]

* Congestion Control → When the server gets overloaded with packets, the client has the capability to [tcp feature]. It will send data only when the network can handle it essentially.

* Ordered Packets → Internet does not guarantee ordering of Packets, and the server sees the sequence and orders them.

(Cons)

* Larger Packets → Because Seq/Ack have to be attached.
* More Bandwidth → more data is need to send large packets.
* Slower than udp
* Stateful → (will be explained.)

Cons)

Server memory :- Since TCP is Stateful, hence each connection has to be maintained. this requires some memory to be allocated on the server.

Hence there is a limit on the no. of tcp connections the server can have.

UDP → (User Data gram Protocol)

CONS

* No Acknowledgement -
* No Guranteed Delivery -
* No Congestion Control - Traffic or no traffic the user does not care.
* No ordering of Packets
* No security → Since the algo is stateless it does not maintain a secure connection, Hence anyone can send data to the server.

Pros

* Smaller packets
* Less Bandwidth
* Faster than tcp
* Stateless → [Discussed in programming Patterns playlist]

Kill the server and bring it back again, and send message from anyone, and it will ~~serve~~ serve the request.

No need for establishing secure connection first.

* (No memory Required)