8

Transmission Control Protocol

TCP

- TCP stands for Transmission Control Protocol.

- It is a transport layer protocol.

- The application layer sends a lot of raw data and the transport layer converts it into data segments meaning it is divided into chunks, adds headers, checksums, etc.

- It may also collect data from the network layer and then network layer divides these segments into more smaller packets or chunks.

- This provides congestion(traffic) control.

- It takes care of these things:

  ```
                  - data received or not.
                  - maintains order of the packets using the sequence numbers.
  ```

- Various other protocols use TCP such as SMTP, POP3, IMAP, etc.

Features

- TCP is connection oriented meaning it first establishes the connection and then it starts processing or sending the data.
- Provides error control.
- Provides congestion control.
- The established connection is `bi-directional` or `full-duplex`
- **Full-duplex :** There are two computers (A & B) connected to each other.A can send a file to computer B and also B can send another file to A or both can send file simultaneously.
- This means that PC_a can send data to PC_b, PC_b can send data to PC_a, and both PC_a and PC_b can send data to each other simultaneously.
- Using a TCP connection, there can only be two end points.

Three Way Handshake

- This is a pre connection process.
- First our computer will ask our friend's computer "Hey, I want to make a connection with you. Can I?".
- The friend's computer replies "Yeah, sure.".
- And then our computer will respond, "Okay!, I am creating a connection now.".
- And then the connection is established.
- This is known as Three Way Handshake.

- The client send Syncronization flag to the server this means new connection is being to establish.Then the server will send a random sequence number for security purposes.
- Then it sends a acknowledgement flag it also have random seq. no.
- acknowledgement no.= prevoius seq. no. +1.

Real world example

- Let's say we want to connect to a server.
- We first send a connection request to the server. We send a `Synchronization Flag` which means we want to establish a connection (a value inside header of a packet).
- The packet also has a sequence number.
- These sequence numbers are random because if they are not, then it can be hacked by hackers.
- In return the server responds "I have got your request." and sends an `Acknowledgement Flag` with a sequence number as the sequence number from the previously sent packet by us and do some math with it.
- And then client acknowledges the flag sent by the server and again sends the acknowledgement flag to the server about we are establishing a connection.
- There is acknowledgement number also in the packet's header which is previous sequence number + 1.