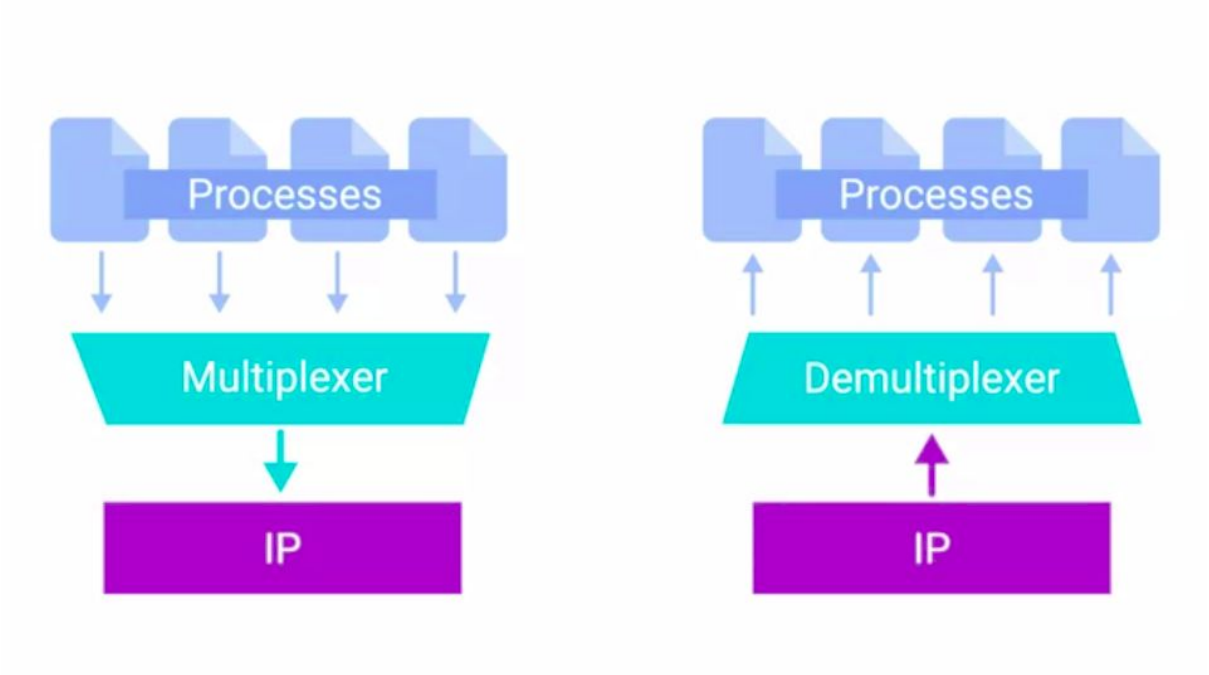
WEEK - 3

Introduction to the Transport & Application Layers

Transport Layer :

Transport layer handles multiplexing and demultiplexing through *ports*.



Port : A 16 bit number that's used to direct traffic to specific services running on a networked computer. Ports are normally denoted with a colon after the IP addresses.

Eg : 10.1.1.100:80 - Here 80 is the port where remaining things are IP. Both IP and Port are collectively called as Socket Address/ socket number.

Source Port : A high- numbered port chosen from a special section of ports known as ephemeral ports.

Destination port : The port of the service the traffic is intended for.

Note : Traditional port for HTTP is port 80 where port 21 for FTP

Connection oriented & Connectionless protocols :

Connection oriented protocol (TCP):

Establishes a connection, and uses this to ensure that all data has been properly transmitted.

Connectionless protocol (UDP):

UDP doesn't rely on connections & it doesn't even support the concept of acknowledgement. Mostly UDP is used in the cases like video streaming, broadcast etc.,

WEEK - 3

Transmission Control Protocol (TCP)

TCP is a connection oriented protocol used in wide applications. It follows a special process called Handshake to establish / terminate the connections.

TCP CONTROL FLAGS :

- **URG** (Urgent)
- **ACK** (Acknowledge - Sending/ receiving an alert message)
- **PSH** (Push - Push the data to the buffer)
- **RST** (Reset)
- **SYN** (Synchronize - It means Let's catchup)
- **FIN** (Finish - It means the transmitting computer doesn't have any more data to send and the connection be closed.)

HANDSHAKE :

A way for two devices to ensure that they are speaking the same protocol and will be able to understand each other.

THREE WAY HANDSHAKING (Connection Establishment) :

Consider Two systems **Computer A & Computer B.** To start the process off, **Computer A,** sends a TCP segment to **Computer B** with a **SYN** flag set. This is Computer A's way of saying ***"Let's establish a connection and look at my sequence number field. So, we know where this conversation starts".***

Computer B then responds with a TCP segment where both **SYN & ACK** flags are set. This is **Computer B** of saying "***Let's establish a connection and I acknowledge your sequence number***".

Then **Computer A** responds again with just the **ACK** flag set. Just say "***I acknowledge your acknowledgement. Let's start sending data***".

This is known to be three way handshaking. Once the three way handshaking is done, TCP connection is established. Now **Computer A** is free to send whatever data it wants to **Computer B** & vice versa.

FOUR WAY HANDSHAKING (Connection Termination)

If two systems want to close their connection they need to carry out four way handshaking.

Consider the same two systems, **Computer A** & **Computer B ,** If Computer B wants to close the connection means it will send a FIN flag set to **Computer A**. By saying like *"**Computer A** **shall we close our connection and I wish to close our connection**"*

Computer A accepts the flag set and sends **ACK** & **FIN** flag set to **Computer B**. By saying "***Okie, let's wind up our connection***". **Note** : Here **ACK** & **FIN** flag set will be sent to **Computer B** as individual TCP segments.

Finally, **Computer B** sends the **TCP** segment with the ACK flag set by saying ***"Okie thanks for your connection and acknowledgement".***

WEEK - 3

This is known to be four-way handshaking. Once the four-way handshaking is done,TCP connection is terminated.

TCP SOCKET STATES :

Socket : The instantiation of an end-point in a potential TCP connection. There are 7 states in TCP connections.

- **LISTEN** - A TCP socket is ready & listening for incoming connections.
- **SYN_SENT** - A synchronization request has been sent, but the connection hasn't been established yet.
- **SYN_RECEIVED** - A socket previously in a **LISTEN** state has received a synchronization request and sent a SYN/ACK back.
- **ESTABLISHED** - The TCP connection is in working order and both sides are free to send each other data.
- **FIN_WAIT** - A FIN has been sent, but the corresponding ACK from the other end hasn't been received yet,
- **CLOSE_WAIT** - Connection is closed, but the application receiving sockets are not closed.
- **CLOSED** - Complete close.

Application Layer

Application layer is generally of two things: **Web browsers** & **Web servers.** Most Web browsers are called Client and Web servers are called Servers.

- Examples for Web browsers are Chrome, Firefox, IE, Safari etc.,.
- ExamplesExamples for Web Servers are Microsoft IIS, Apacke , nginx etc.,

--- THE END ---