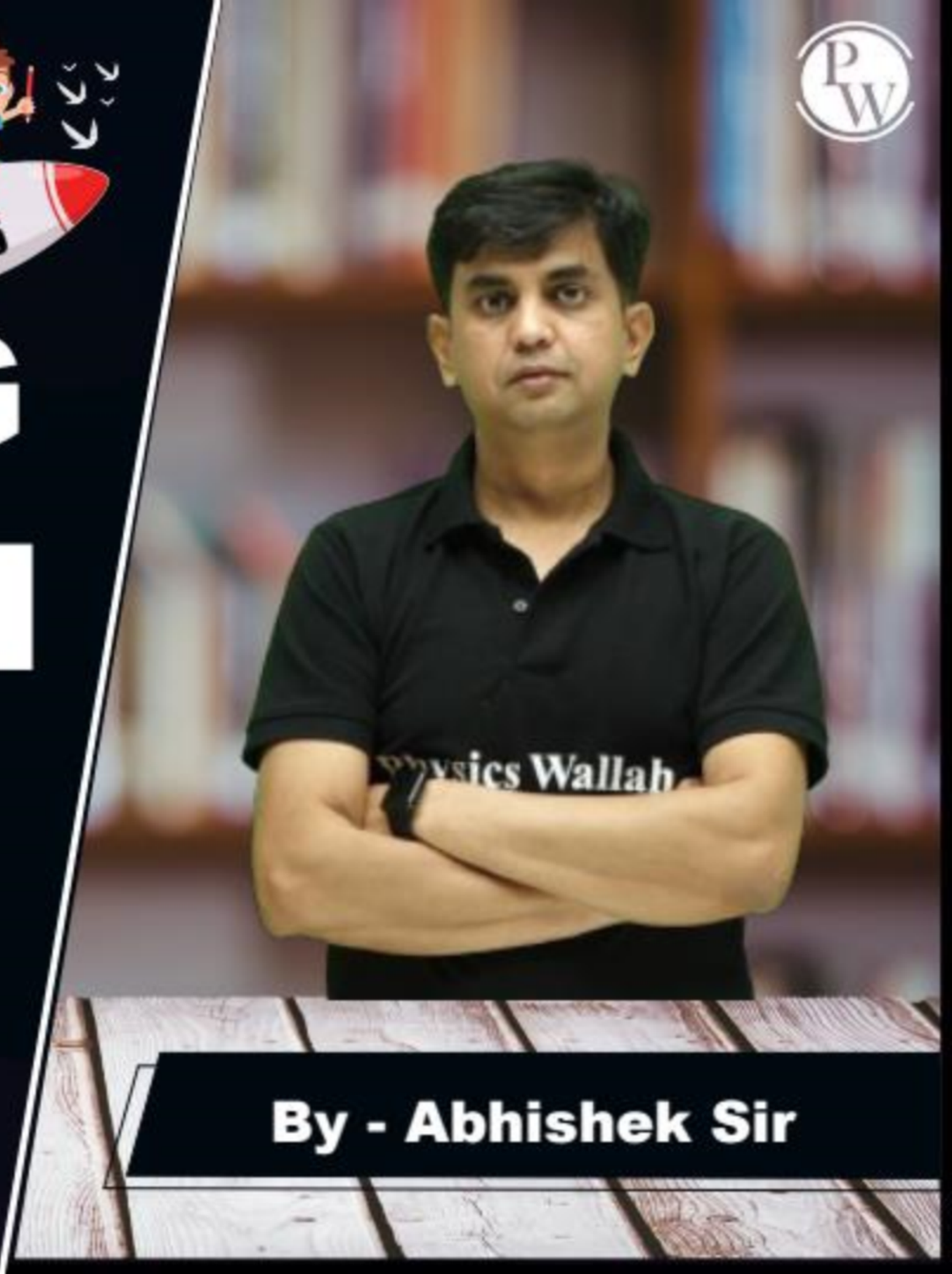
# CS & IT ENGINEERING


## Computer Network

## Transport Layer

Lecture No. - 02



By - Abhishek Sir

# Recap of Previous Lecture

Topic DHCP

Topic Transport Layer

Topic UDP

# Topics to be Covered

- Topic UDP
- Topic TCP

# ABOUT ME

**Hello, I'm Abhishek**

- GATE CS AIR - 96
- M.Tech (CS) - IIT Kharagpur
- 12 years of GATE CS teaching experience

**Telegram Link : https://t.me/abhisheksirCS_PW**

#Q. For a host machine that uses the token bucket algorithm for congestion control, the token bucket has a capacity of 1 megabyte and the maximum output rate is 20 megabytes per second. Tokens arrive at a rate to sustain output at a rate of 10 megabytes per second. The token bucket is currently full and the machine needs to send 12 megabytes of data. The minimum time required to transmit the data is _____ seconds. [GATE 2016]

① Without Token Bucket [Token Bucket Empty]

1 sec ⟶ 10 MB

10 MB ⟶ 1 sec

12 MB ⟶ 1.2 sec

② With Token Buck

1 sec ⟶ 20 MB

0.1 sec ⟶ 2 MB

---

1 sec ⟶ 10 MB

10 MB ⟶ 1 sec

Ans = 1.1 sec

Solution :-

Data Size = 12 MB

C = 1 MB (Megabytes) (Full)

M = 20 MBps

R = 10 MBps

S = Maximum duration for which the computer can transmit at the full capacity

$$S = \frac{C}{(M-R)} = \frac{1\,MB}{(20\,MBps - 10\,MBps)} = \frac{1\,MB}{10\,MBps} = 0.1\,sec$$

$$\text{Minimum time required to transmit the data} = \boxed{S} + \frac{(DataSize - M * S)}{R}$$

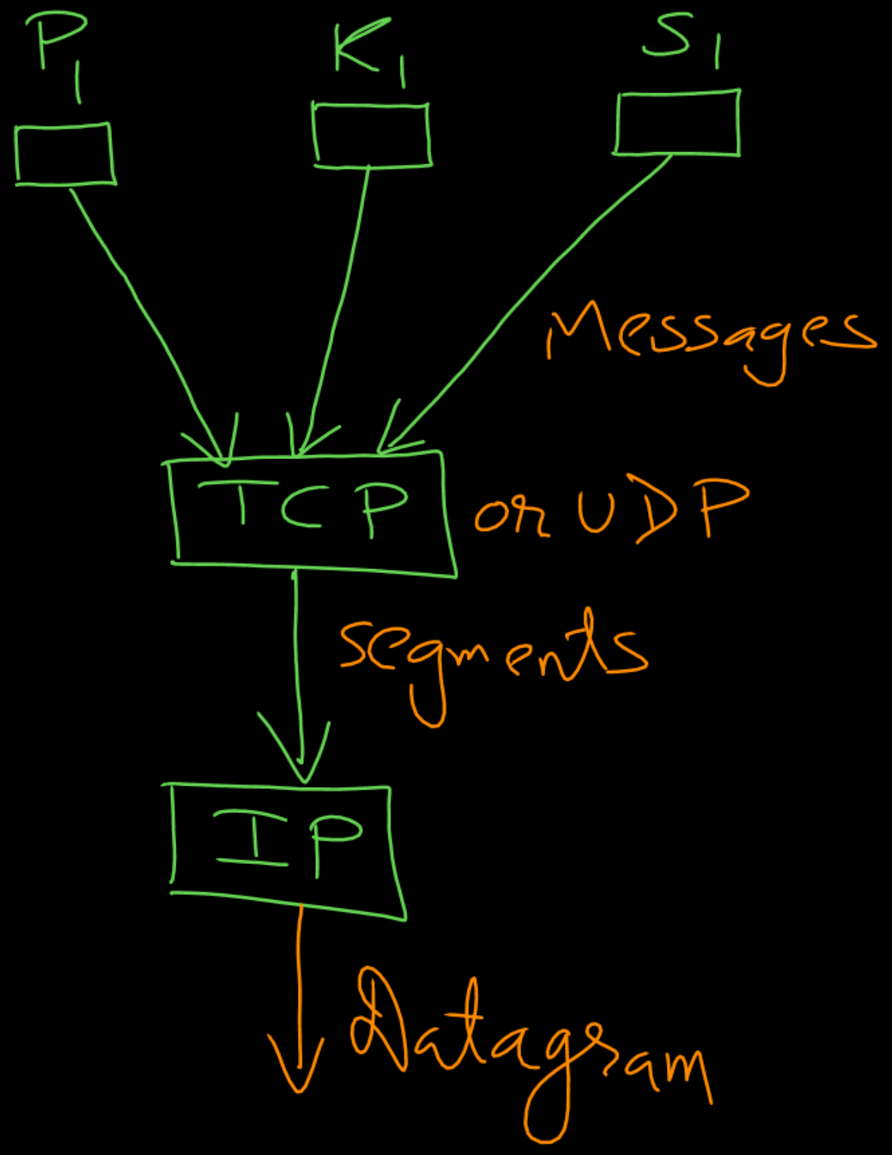
$$= 0.1\,sec + \frac{(12\,MB - 20\,MB/sec * 0.1\,sec)}{10\,MB/sec} = 0.1\,sec + 1\,sec = 1.1\,sec$$

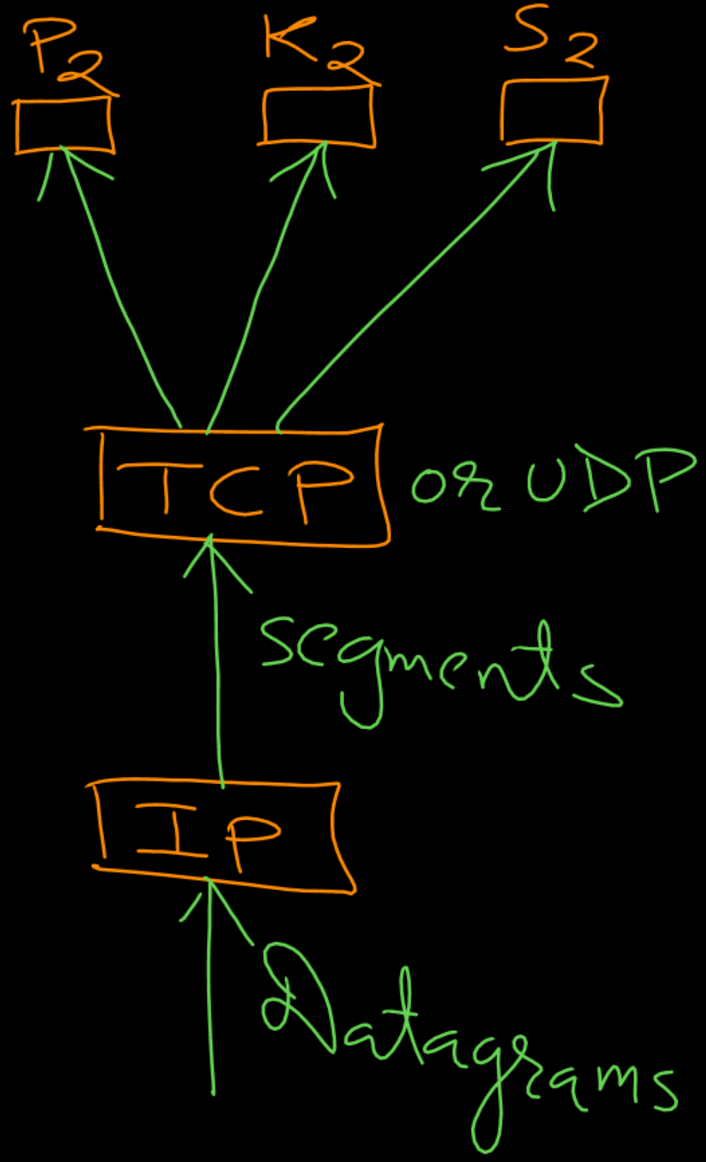
# Topic : Multiplexing

Multiplexing at sender :

-> Handle data from multiple sockets

-> Add transport layer protocol header
[Used for demultiplexing at receiver)

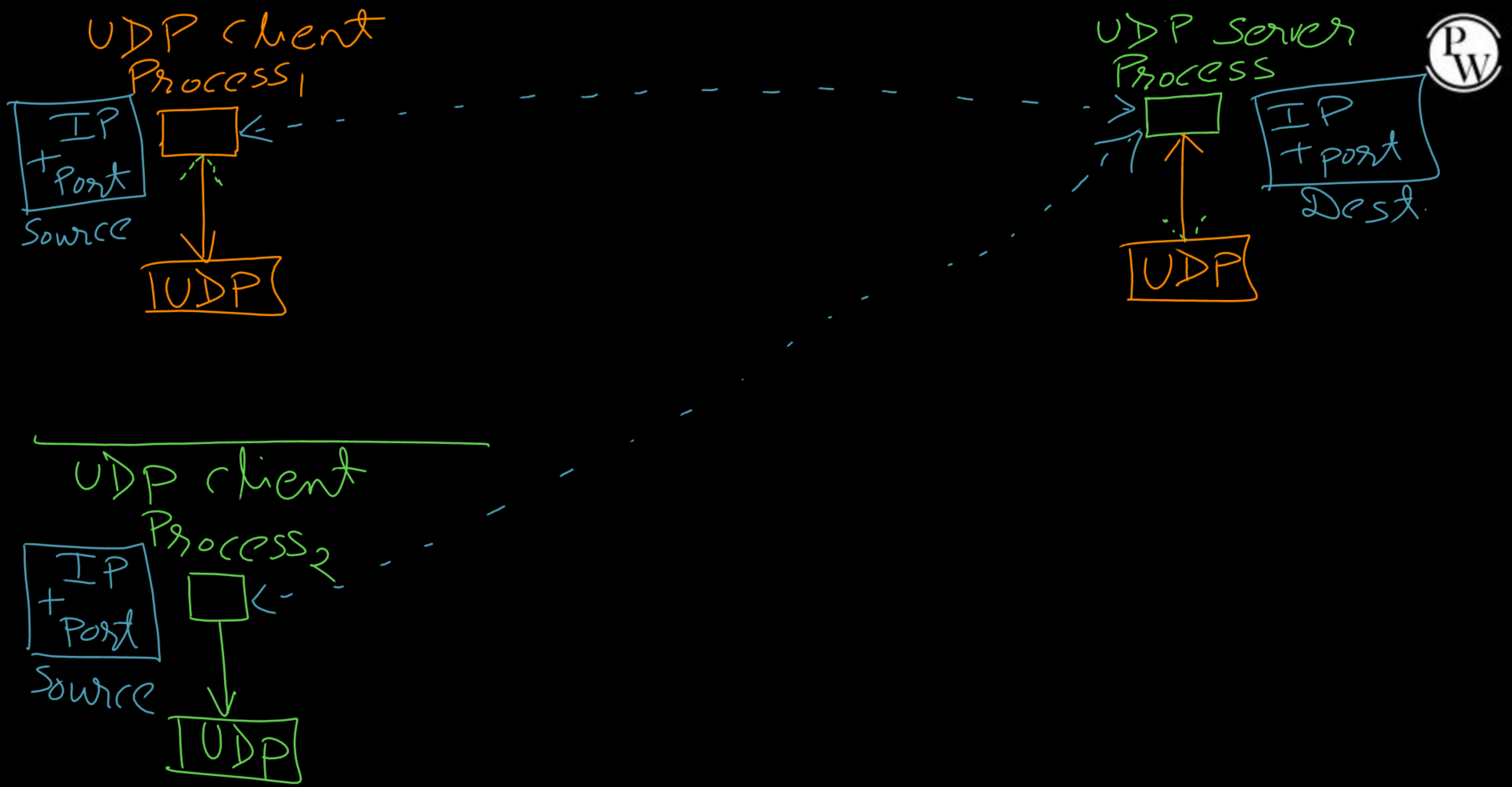

P1
K1
S1
Messages
TCP
or UDP
segments
IP
Datagram

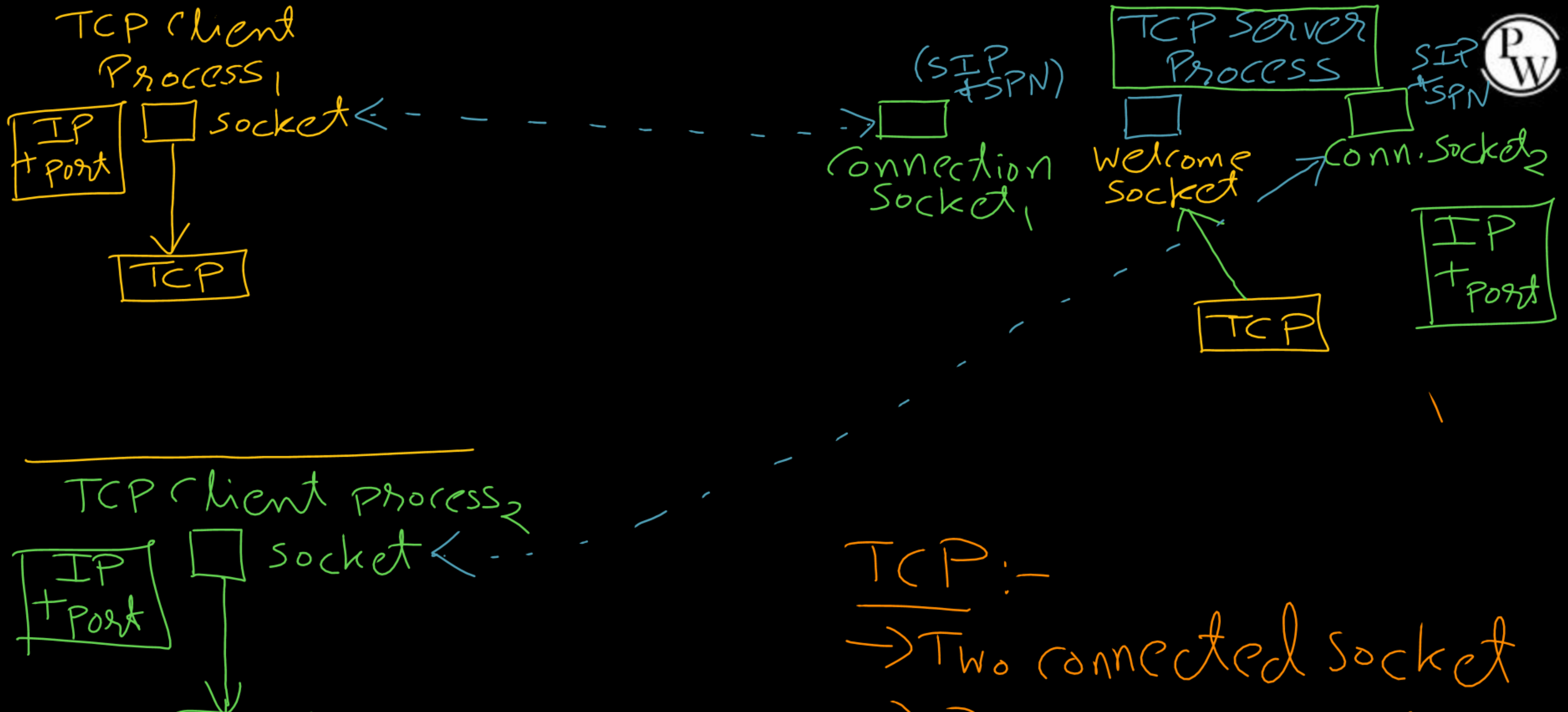

P2
K2
S2
TCP
or UDP
segments
IP
Datagrams

# Topic : Demultiplexing

Demultiplexing at receiver :

1. Connection-less demultiplexing
   -> UDP sockets identified by 2 tuples :
   Destination IP address and Destination Port Number
2. Connection-oriented demultiplexing
   -> TCP sockets identified by 4 tuples :
   Source and Destination IP address
   Source and Destination Port Number


UDP client
Process1
IP
+
Port
Source
UDP
UDP Server
Process
IP
+
port
Dest.
UDP
UDP client
Process2
IP
+
Port
Source
UDP




TCP :-

→ Two connected socket

→ Point to Point Full Duplex comm

# Topic : UDP

=> Connection-less :

-> No handshaking between UDP sender and receiver (Unlike TCP)

-> Each UDP segment handled independently of others

-> Delivery of messages can be any order to the communicating process

=> Unrealiable :

-> Messages may be lost

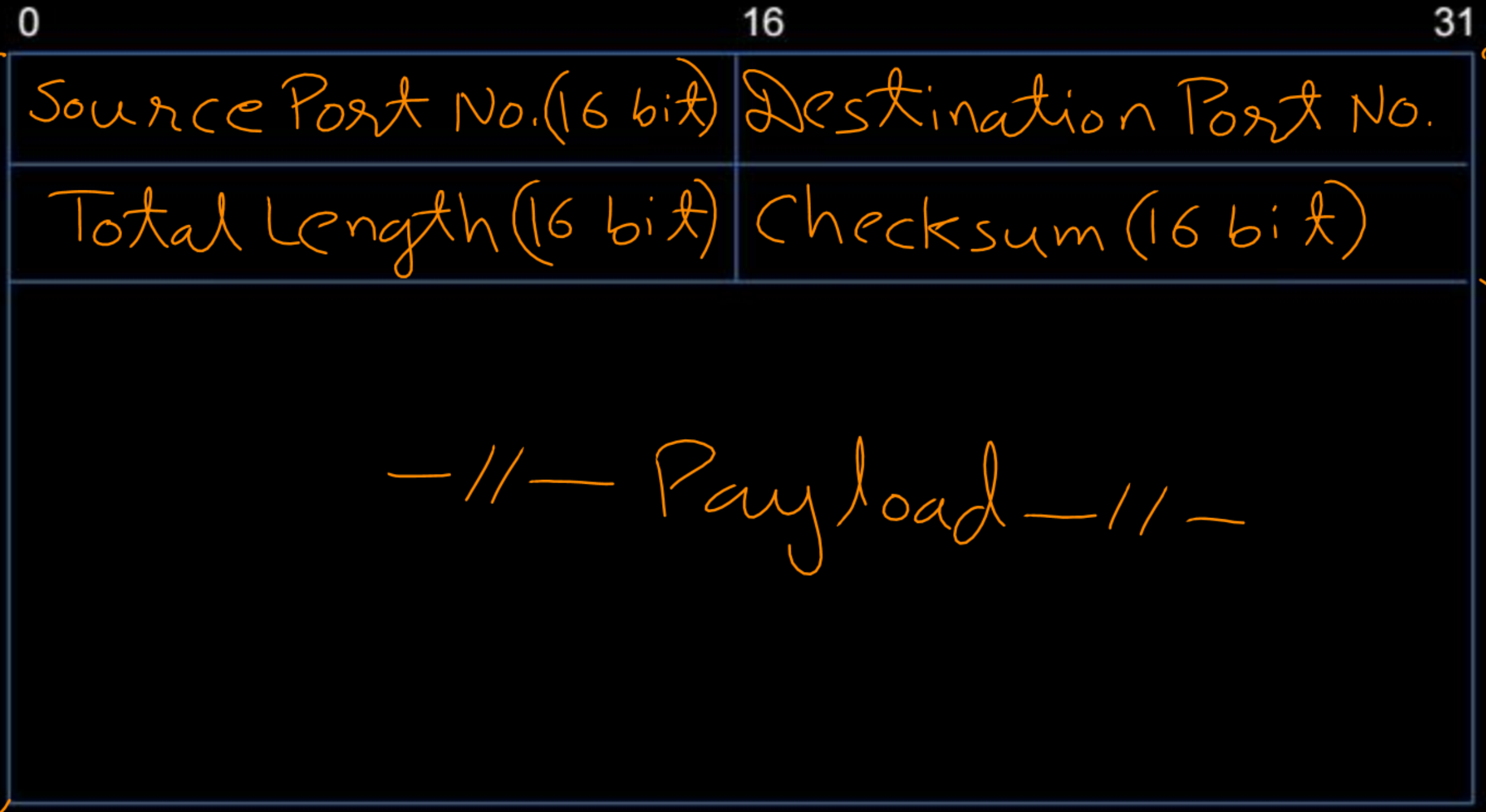
=> Simple and Fast :

-> No connection state at sender and receiver

-> Small (8 bytes, fixed) header size

-> No congestion control

# Topic : UDP Header



# Topic : UDP Header

=> Source port number : 16-bits

=> Destination port number : 16-bits

=> Total Length : 16-bits

-> Size of UDP segment in bytes

-> Including 8 bytes header size

-> Maximum UDP segment size = $(2^{16} - 1)$ bytes

-> Maximum UDP payload size = $(2^{16} - 9)$ bytes

UDP Packet Payload size = [Total length − 8] bytes

# Topic : UDP Header

=> Checksum : 16-bits

-> 16-bit one's complement of the one's complement sum of all 16-bit words in UDP segment

-> Error detection is optional in UDP, it depends upon the application

-> If application wants to disable error detection process then sender set all the 16 bits are zero of checksum field

# 2 mins Summary

Topic UDP

Topic TCP

# THANK - YOU