UNIT-IV

TRANSPORT LAYER

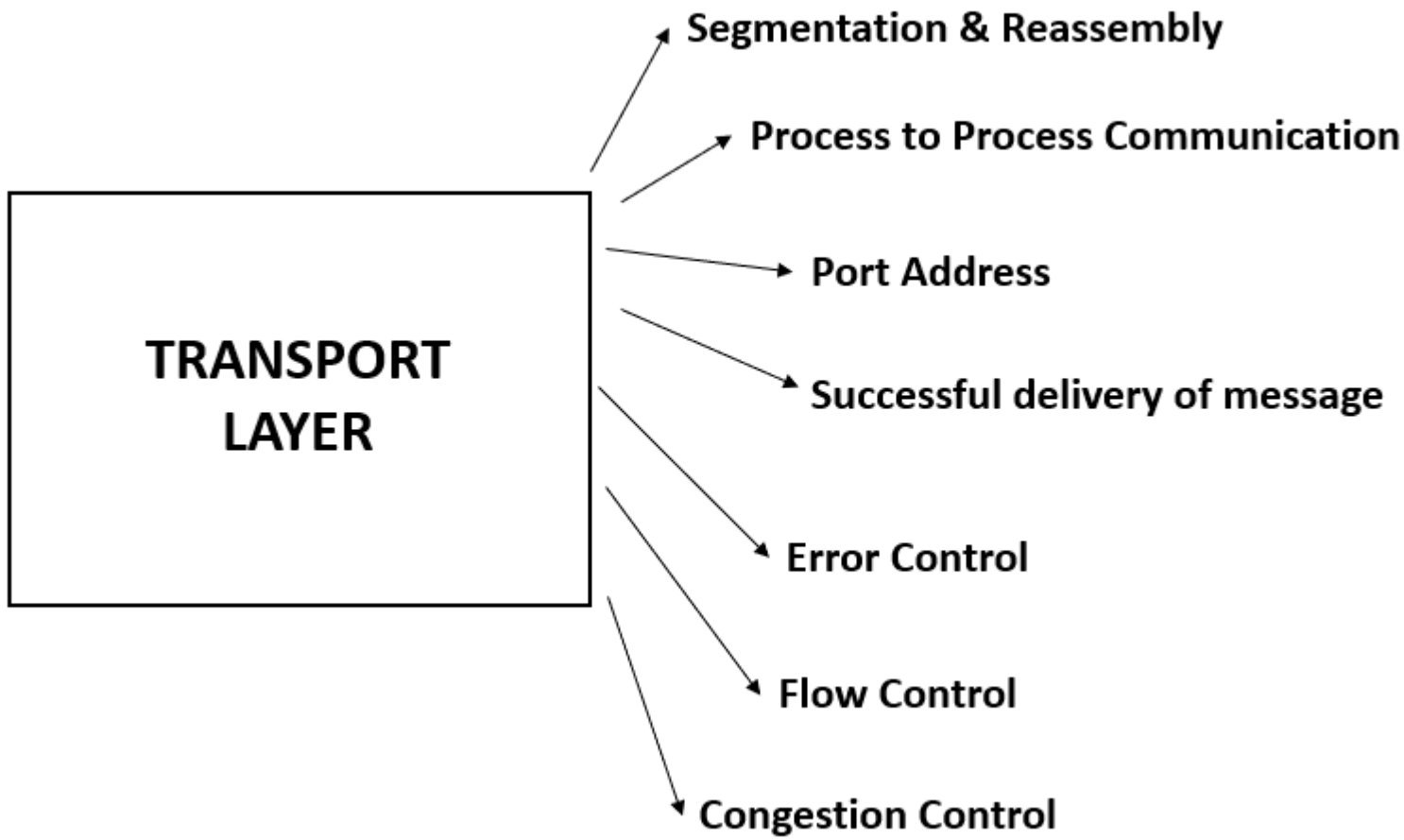
-Prepared By

Ms. Sumedha Seniaray

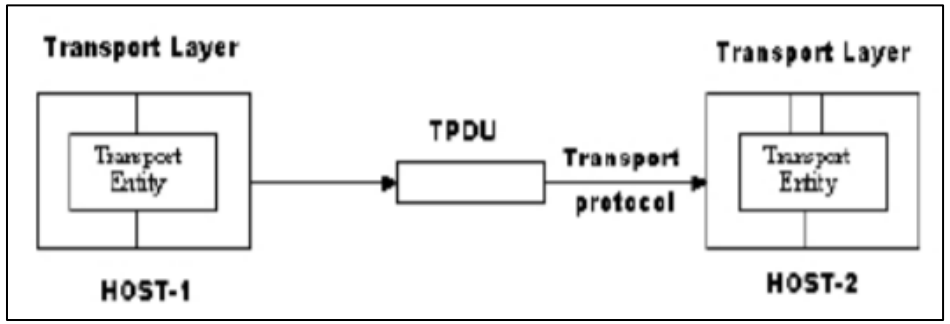
Assistant Professor

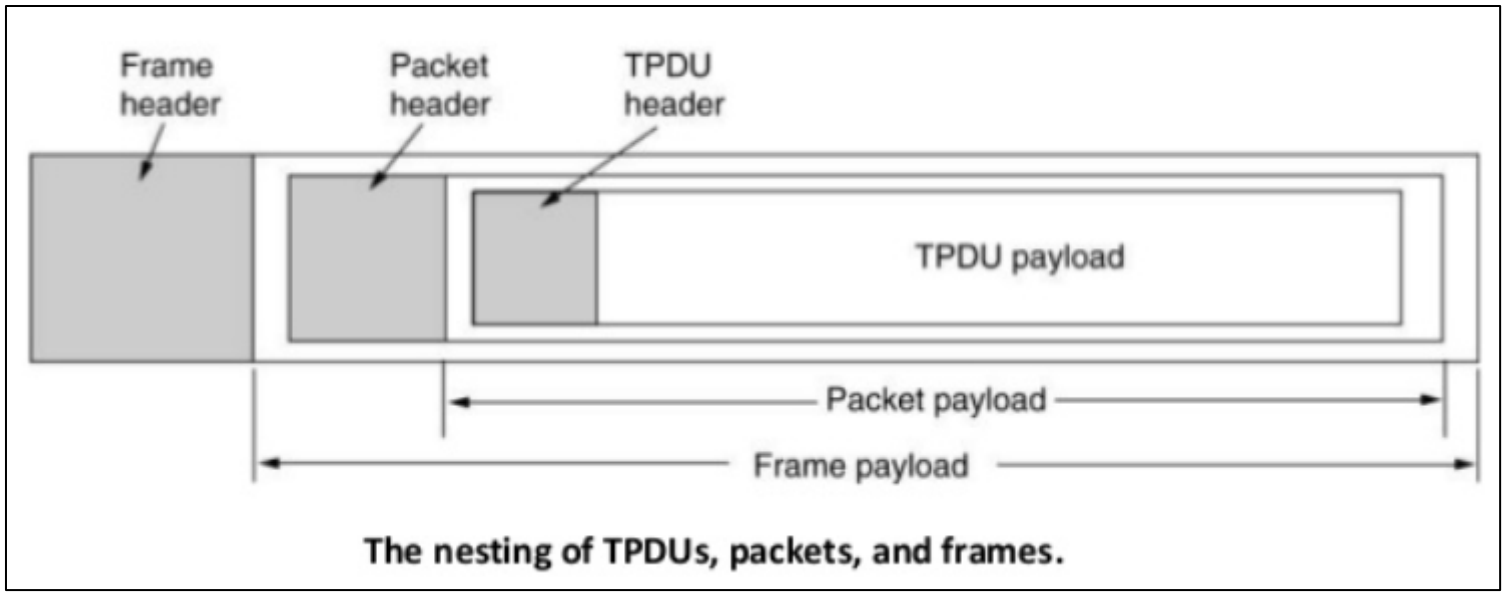
Department of Applied Mathematics, DTU

Transport Layer



- **Transport Entity:** The hardware and/or software which make use of services provided by the network layer, (within the transport layer) is called transport entity.
- **TPDU (Transport Protocol Data Unit):** Transmissions of message between 2 transport entities are carried out by TPDU. The transport entity carries out the transport service primitives and encapsulated in the payload of this packet is a transport layer message for the server's transport entity. The task of the transport layer is to provide reliable, cost-effective data transport from the source machine to the destination machine, independent of physical network or networks currently in use.





The nesting of TPDUs, packets, and frames.

Transport Layer Design Issues

- Transport layer delivers the message from 1 process to another running on two different hosts. Thus, it has to fulfil various responsibilities/ perform certain functions to ensure accurate delivery of message.
- **Various functions of Transport layer are:**

  - Establishing, maintaining and releasing connection
  - Addressing
  - Data transfer
  - Flow control
  - Error control
  - Congestion control

I. Connection Management:

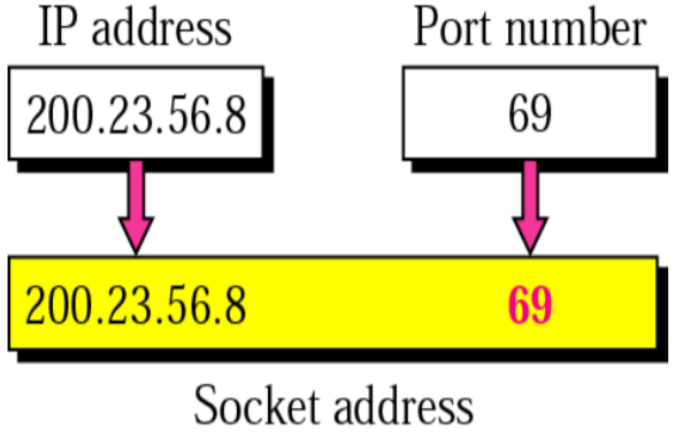
- 3 phases:
  - Establishment of connection (established by the upper layers)
  - Maintain + Data transfer
  - Release connection (when requested by the upper layers)

II. Addressing:

- At the transport layer, transport layer address called a **port no.** is required to choose among the multiple processes running on the destination host.
- Therefore, port no. defines one of the processes on the destination host.
- Destination port no. is required for delivery and Source port no. is required for reply.
- In the Internet model (TCP/IP model), the port nos. are 16 bit integers between 0 and 65,535.

Socket Address

- Socket interface is based on UNIX, defines a set of system calls or procedure. The communication structure needed in such a programming is called a **socket**.
- Socket acts as an end point.
- 2 processes can communicate iff they have a socket at each end.
- **Socket address** is a combination of IP address and Port no.
- It is a 48 bit address.



III. Data Transfer:

- Transport layer breaks user data into smaller units and attaches a transport layer header to each unit forming a TPDU.
- TPDU is handed over to the Network layer for routing it to the destination.
- TPDU header contains port no., sequence no., ack no., checksum etc.

IV. Flow Control:

- Flow control in transport layer is performed end-to-end rather than node-to-node as in DLL.
- Transport layers uses sliding window protocol to perform flow control.

V. Error Control:

- Transport layer also provides end-to-end error control facility.
- Some of the types of errors are:
  - Error due to:
    - Damaged bits
    - Non-delivery of TPDU
    - Duplicate delivery of TPDU
    - Delivery of TPDU to a wrong destination.

VI. Congestion control:

- Several congestion control algorithms are used to avoid congestion.

Transport Layer Services

- There are 2 types are Transport layer services:
  - Connection-oriented service
  - Connectionless service

I. **Connection-Oriented Service:**

Step 1 : Establish connection between the sender and the receiver.

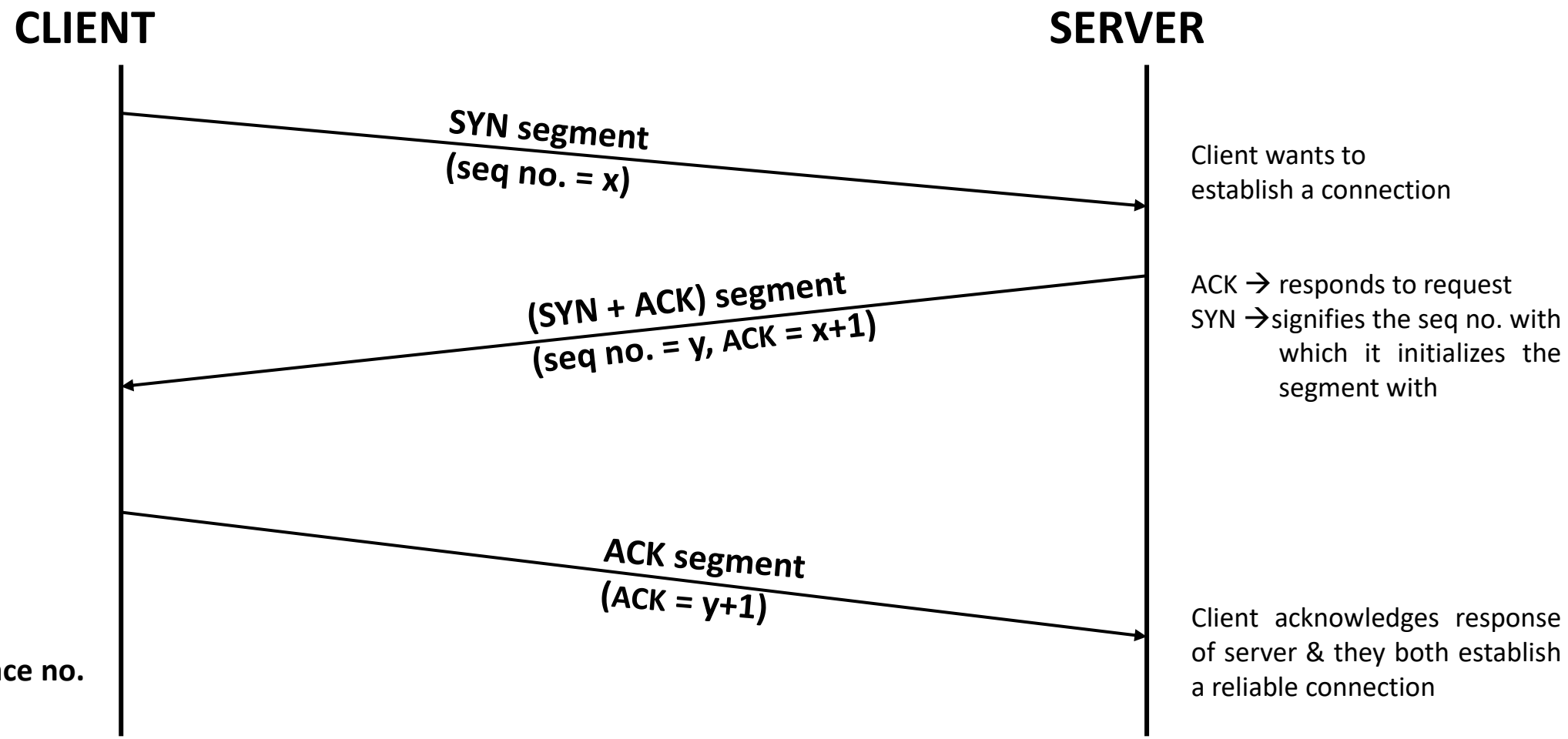
Step 2 : Data transfer

Step 3 : Connection is released.

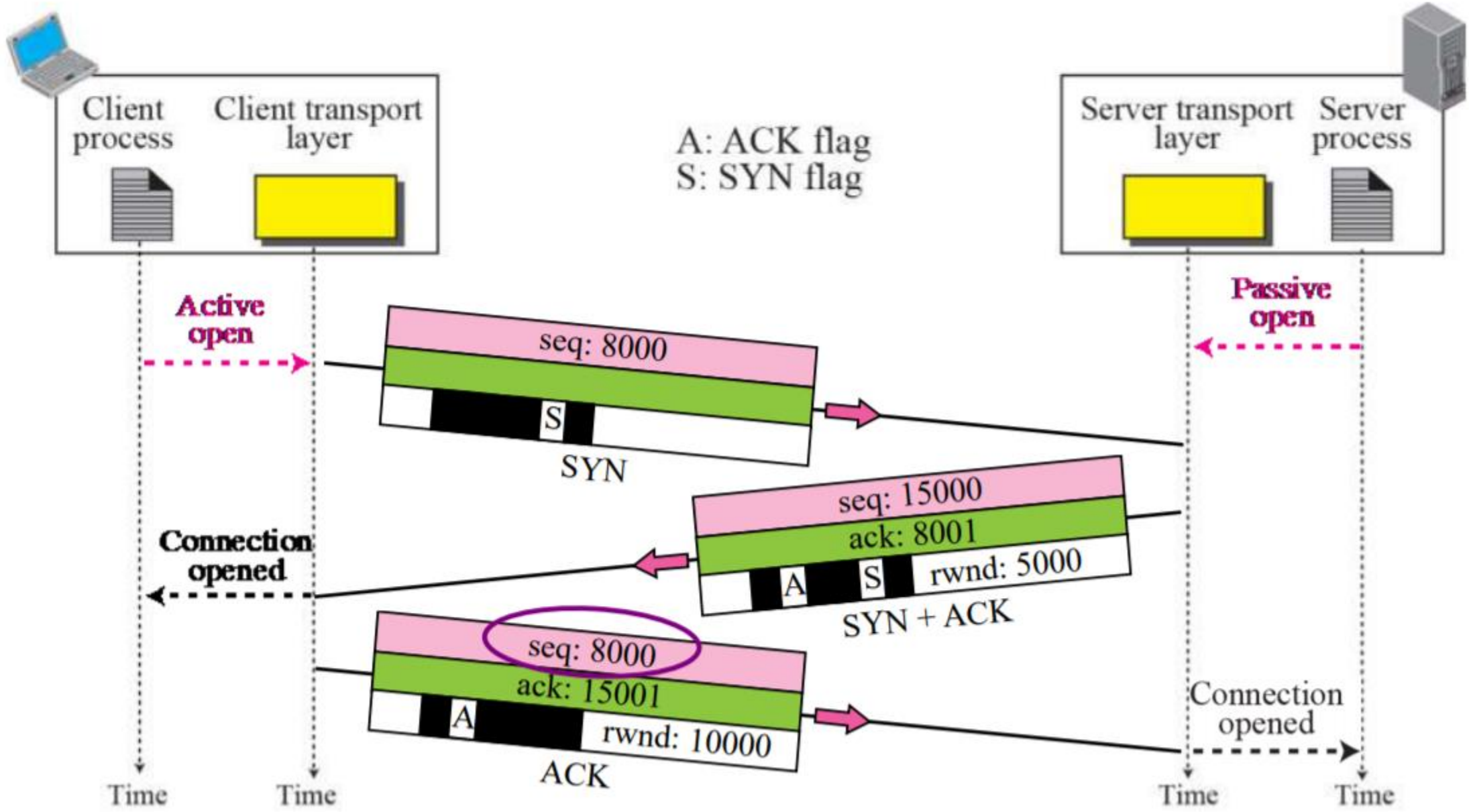
- Connection-oriented service is generally **reliable.**
- Transport Layer Protocol that support connection-oriented service are **TCP (Transmission Control Protocol) and SCTP (Stream Control Transmission Protocol).**

- We use **3-way handshaking technique** for connection management in this type of service.

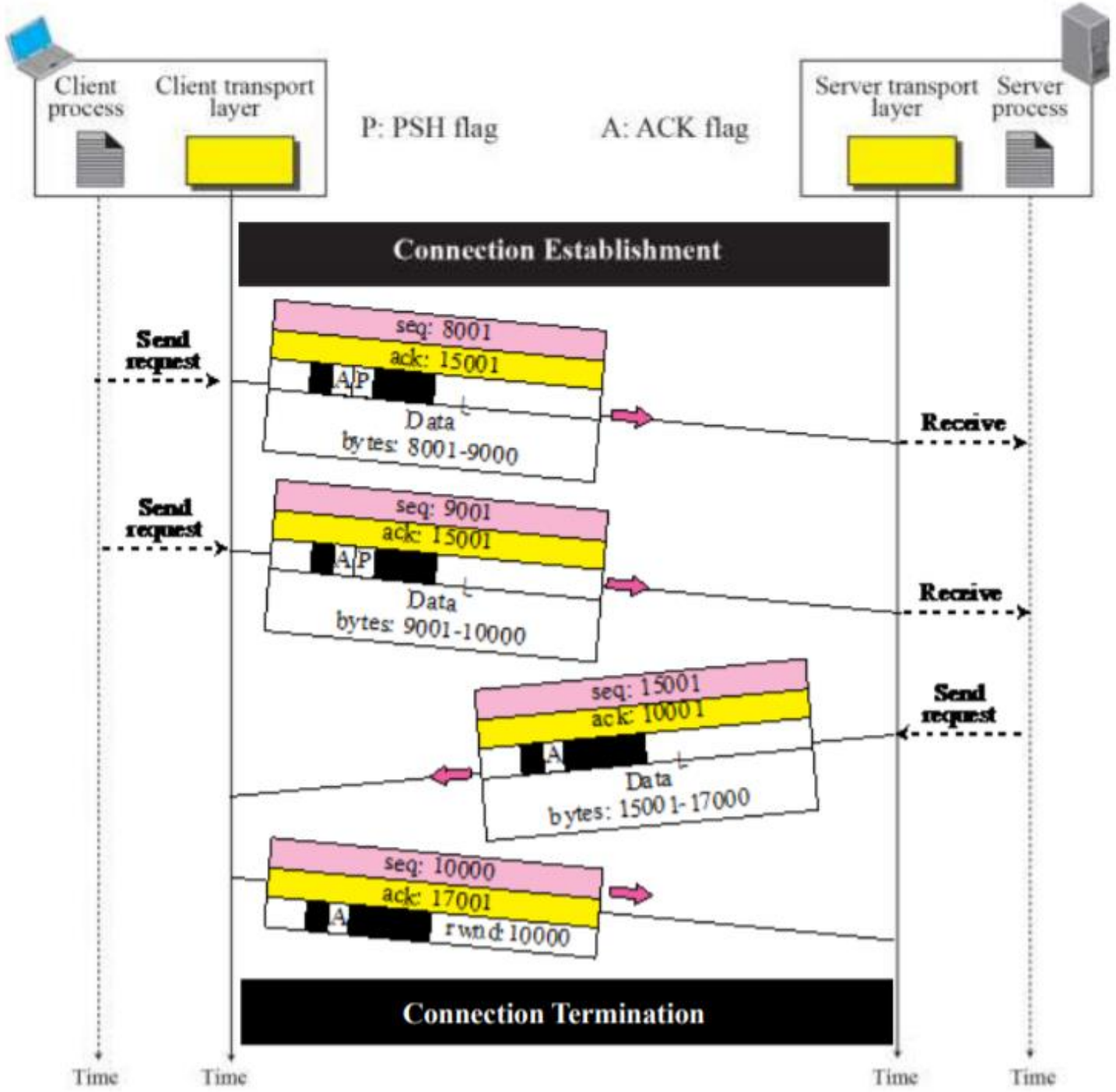
CONNECTION ESTABLISHMENT:



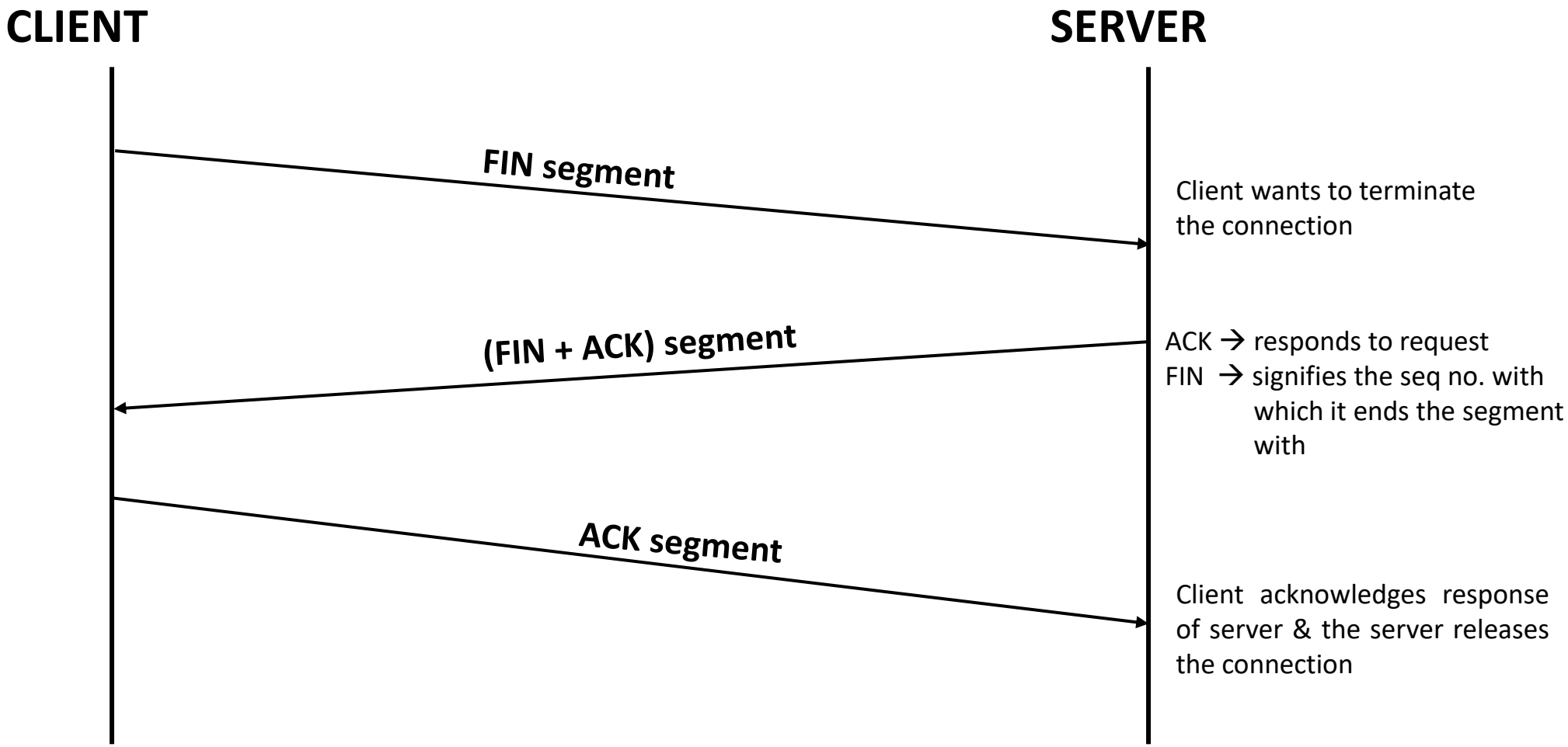
SYN: Synchronize sequence no.
or
Synthesis

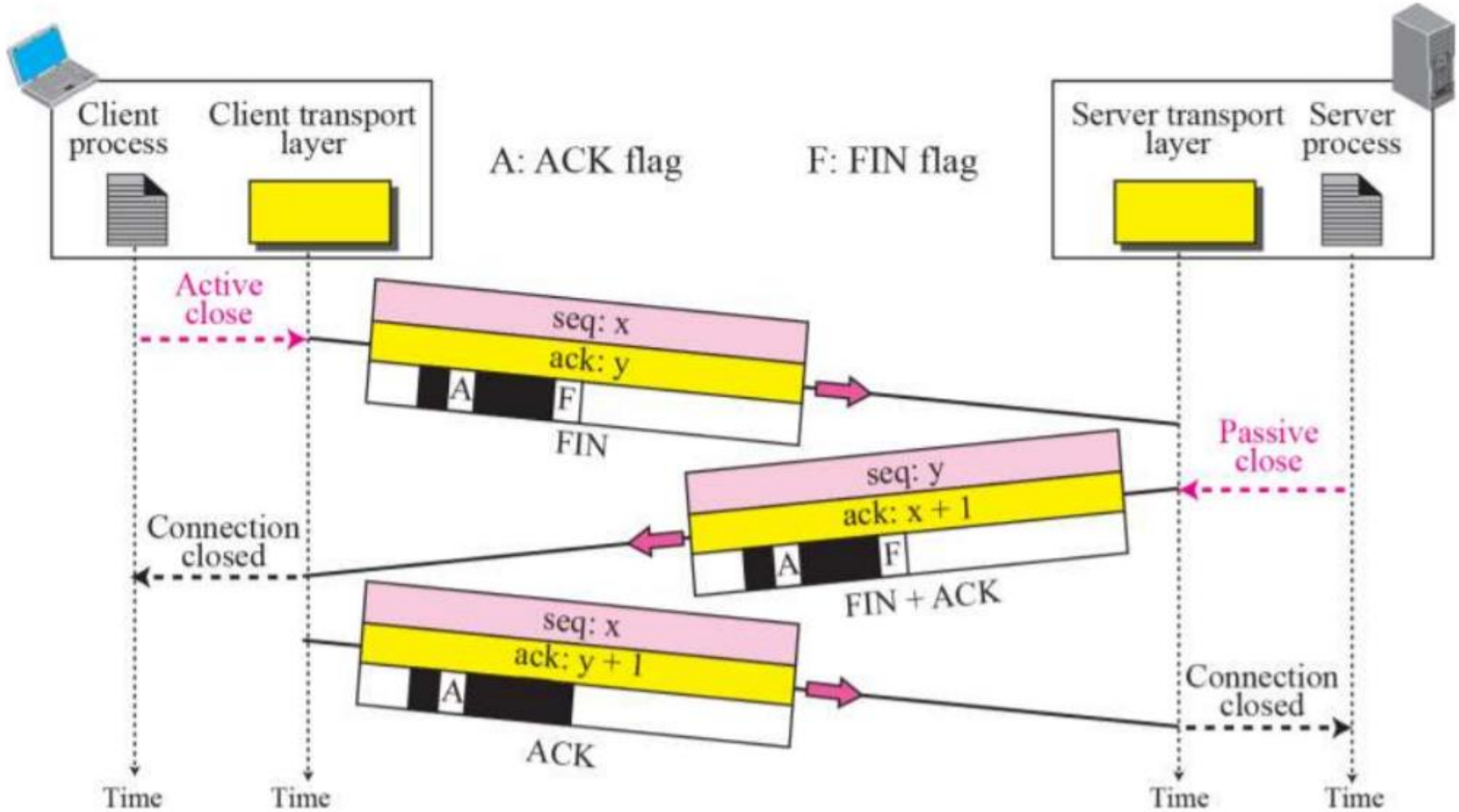

Client process
Client transport layer
A: ACK flag
S: SYN flag
Server transport layer
Server process
Active open
Passive open
seq: 8000
S
SYN
seq: 15000
ack: 8001
A
S
rwnd: 5000
SYN + ACK
Connection opened
seq: 8000
ack: 15001
A
rwnd: 10000
ACK
Connection opened
Time
Time
Time
Time

DATA TRANSFER:



CONNECTION RELEASE :




Client process
Client transport layer
A: ACK flag
F: FIN flag
Server transport layer
Server process
Active close
seq: x
ack: y
A
F
FIN
Passive close
seq: y
ack: x + 1
Connection closed
FIN + ACK
seq: x
ack: y + 1
ACK
Connection closed
Time
Time
Time
Time

II. **Connectionless Service:**

- In this service, packets are sent from sender to receiver without establishment of connection.
- In such a service, packets are not numbered.
- Packets maybe lost, corrupted, delayed or unordered.
- Therefore, connectionless service is **unreliable.**
- Transport layer protocol that provides this service is **UDP.**