SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE NAME :19IT401 COMPUTER NETWORKS

II YEAR /IV SEMESTER

Unit 4-Transport layer

Topics 2 : TCP - services – segment-connection

TCP

✓Transmission Control Protocol (TCP) is a connection-oriented, reliable protocol.

✓TCP explicitly defines connection establishment, data transfer, and connection teardown phases to provide a connection-oriented service.

✓TCP uses a combination of GBN (**Go-Back-N Protocol)**and SR (**Selective Repeat Protocol)** protocols to provide reliability.

✓To achieve this goal, TCP uses checksum (for error detection), retransmission of lost or corrupted packets, cumulative and selective acknowledgments, and timers.

TCP services

Process-to-Process Communication

Stream Delivery Service

✓TCP, unlike UDP(packet oriented), is a stream-oriented protocol. TCP, on the other hand, allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes.

Sending and Receiving Buffers

✓Because the sending and the receiving processes may not necessarily write or read data at the same rate, TCP needs buffers for storage.

✓There are two buffers, the sending buffer and the receiving buffer, one for each direction.

✓These buffers are also necessary for flow- and error-control mechanisms used by TCP.

Full-Duplex Communication

✓TCP offers *full-duplex service, where data can flow in both directions at the same time.*

✓Each TCP endpoint then has its own sending and receiving buffer, and segments move in both directions.

Multiplexing and Demultiplexing

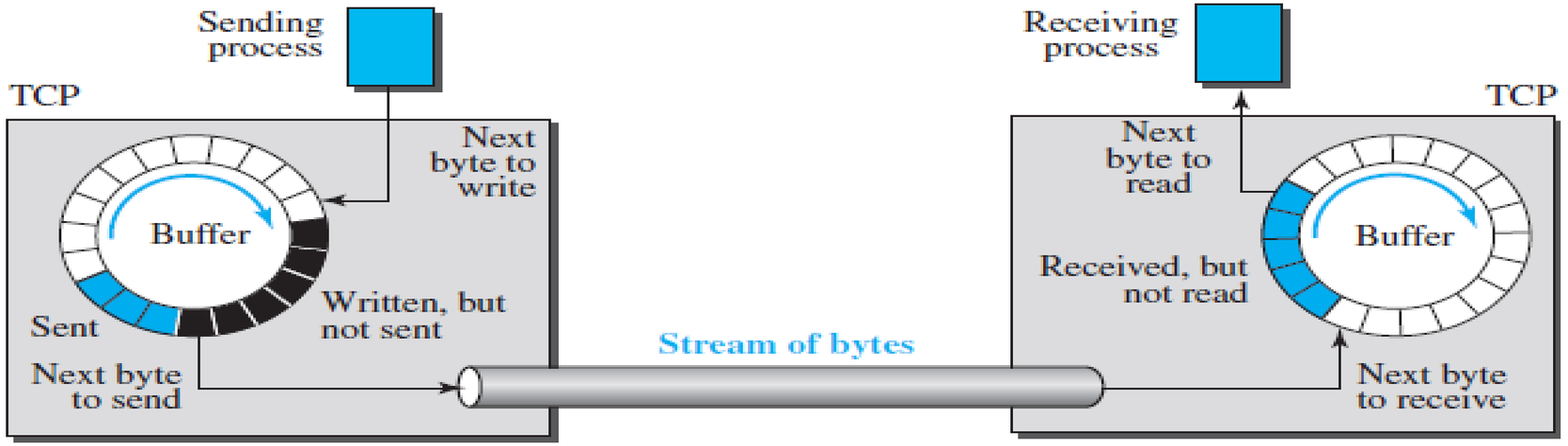
✓Like UDP, TCP performs multiplexing at the sender and demultiplexing at the receiver.

✓However, since TCP is a connection-oriented protocol, a connection needs to be established for each pair of processes.

Connection-Oriented Service

Reliable service

Figure 24.5 *Sending and receiving buffers*



Suppose a TCP connection is transferring a file of 5,000 bytes. The first byte is numbered 10,001. What are the sequence numbers for each segment if data are sent in five segments, each carrying 1,000 bytes?

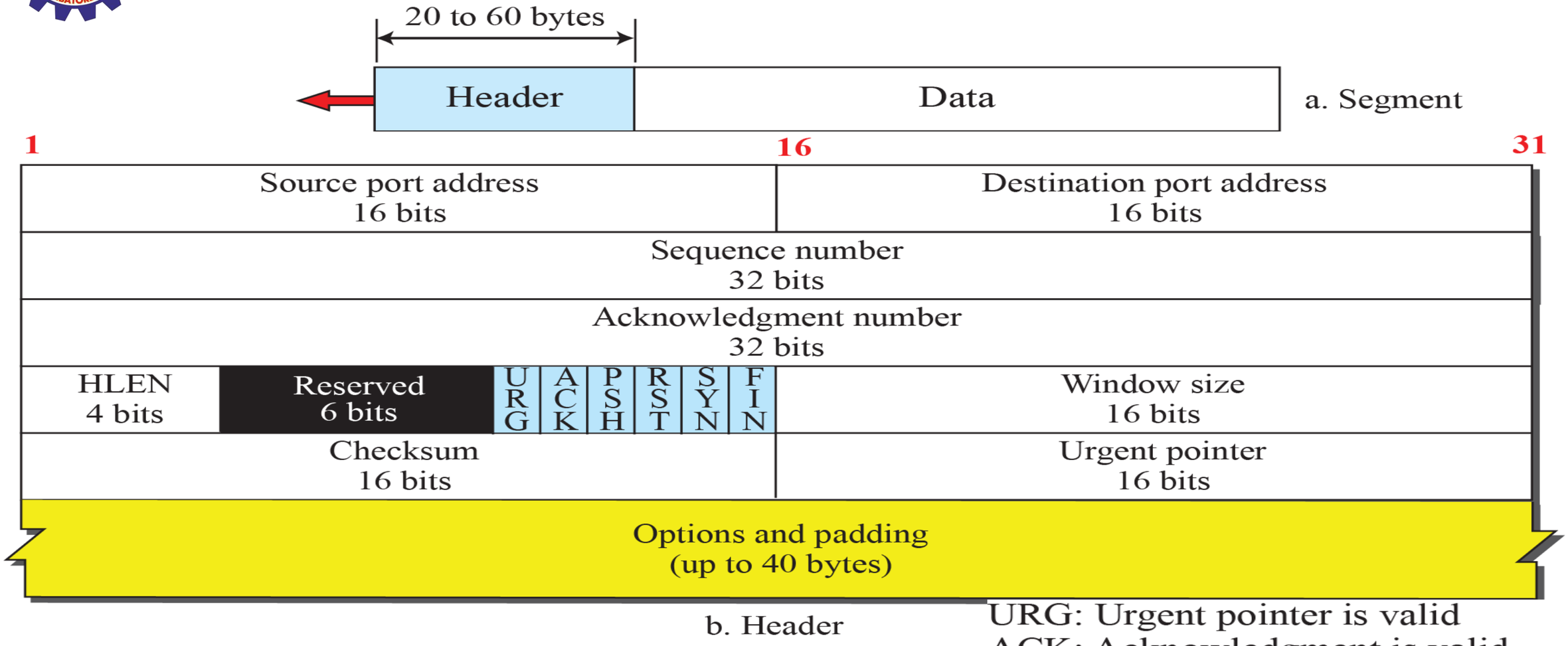
Solution

The following shows the sequence number for each segment:

Segment 1	→	Sequence Number:	10,001	**Range:**	10,001	to	11,000
Segment 2	→	Sequence Number:	11,001	**Range:**	11,001	to	12,000
Segment 3	→	Sequence Number:	12,001	**Range:**	12,001	to	13,000
Segment 4	→	Sequence Number:	13,001	**Range:**	13,001	to	14,000
Segment 5	→	Sequence Number:	14,001	**Range:**	14,001	to	15,000

❑*Source port address.*
❑ *Destination port address*
❑ *Sequence number. This 32-bit field defines the number assigned to the first byte of* data contained in this segment.
❑ *Acknowledgment number* If the receiver of the segment has successfully received byte number *x from the other party, it* returns *x + 1 as the acknowledgment number. Acknowledgment and data can be* piggybacked together.
❑ *Header length.*. The length of the header can be between 20 and 60 bytes
❑ *Control. This field defines 6 different control bits or flags, as shown in Figure .*
One or more of these bits can be set at a time. These bits enable flow control, connection establishment and termination, connection abortion, and the mode of data transfer in TCP.
❑***Window size*** This value is normally referred to as the receiving window (*rwnd) and is determined by the receiver. The sender must obey the dictation of the* receiver in this case.
❑ ***Checksum*** use of the checksum for TCP is mandatory . The same(UDP) pseudoheader, serving the same purpose, is added to the segment. For the TCP pseudoheader, the value for the protocol field is 6.
❑ ***Urgent pointer.*** TCP urgent mode is a service by which the application program at the sender side marks some portion of the byte stream as needing special treatment by the application program at the receiver side. The receiving TCP delivers bytes (urgent or nonurgent) to the application program in order
❑ ***Options. There can be up to 40 bytes of optional information in the TCP header.***

TCP segment format



URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection

TCP Connection

TCP is connection-oriented. All of the segments belonging to a message are then sent over this logical path. Using a single logical pathway for the entire message facilitates the acknowledgment process as well as retransmission of damaged or lost frames.

In TCP, connection-oriented transmission requires three phases:

1. connection establishment
2. data transfer
3. connection termination

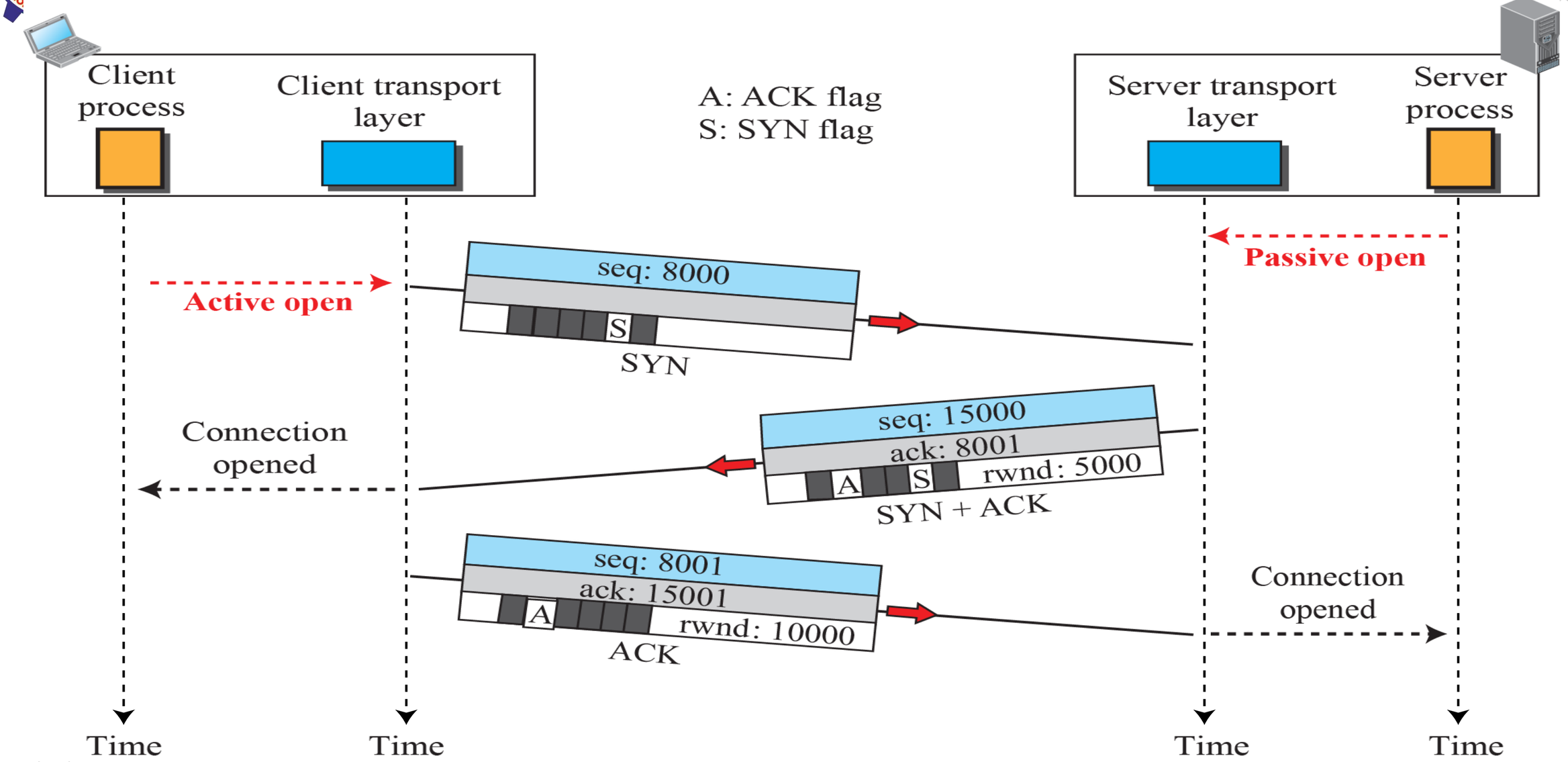
The three steps in this phase are as follows.

1. The client sends the first segment, a SYN segment, in which only the SYN flag is set. This segment is for synchronization of sequence numbers.

2.The server sends the second segment, a SYN + ACK segment with two flag bits set as: SYN and ACK. This segment has a dual purpose. First, it is a SYN segment for communication in the other direction. The server uses this segment to initialize a sequence number for numbering the bytes sent from the server to the client. The server also acknowledges the receipt of the SYN segment from the client by setting the ACK flag and displaying the next sequence number it expects to receive from the client.

3.The client sends the third segment. This is just an ACK segment. It acknowledges the receipt of the second segment with the ACK flag and acknowledgment number field.

Connection establishment using three-way handshaking



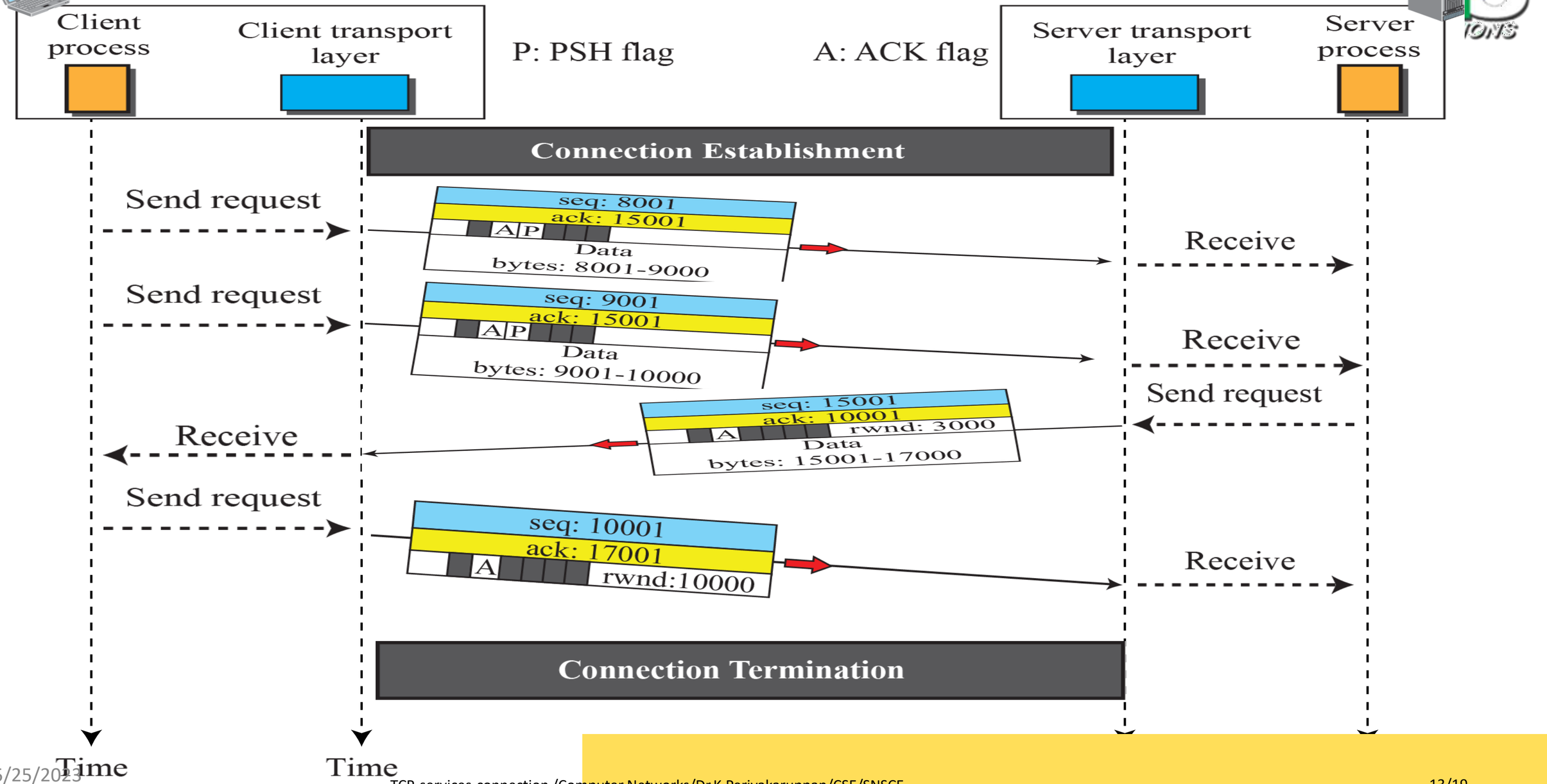
Data Transfer

After connection is established, bidirectional data transfer can take place. The client and server can send data and acknowledgments in both directions.

The data segments sent by the client have the PSH (push) flag set so that the server TCP knows to deliver data to the server process as soon as they are received.

This means that the sending TCP must not wait for the window to be filled. It must create a segment and send it immediately

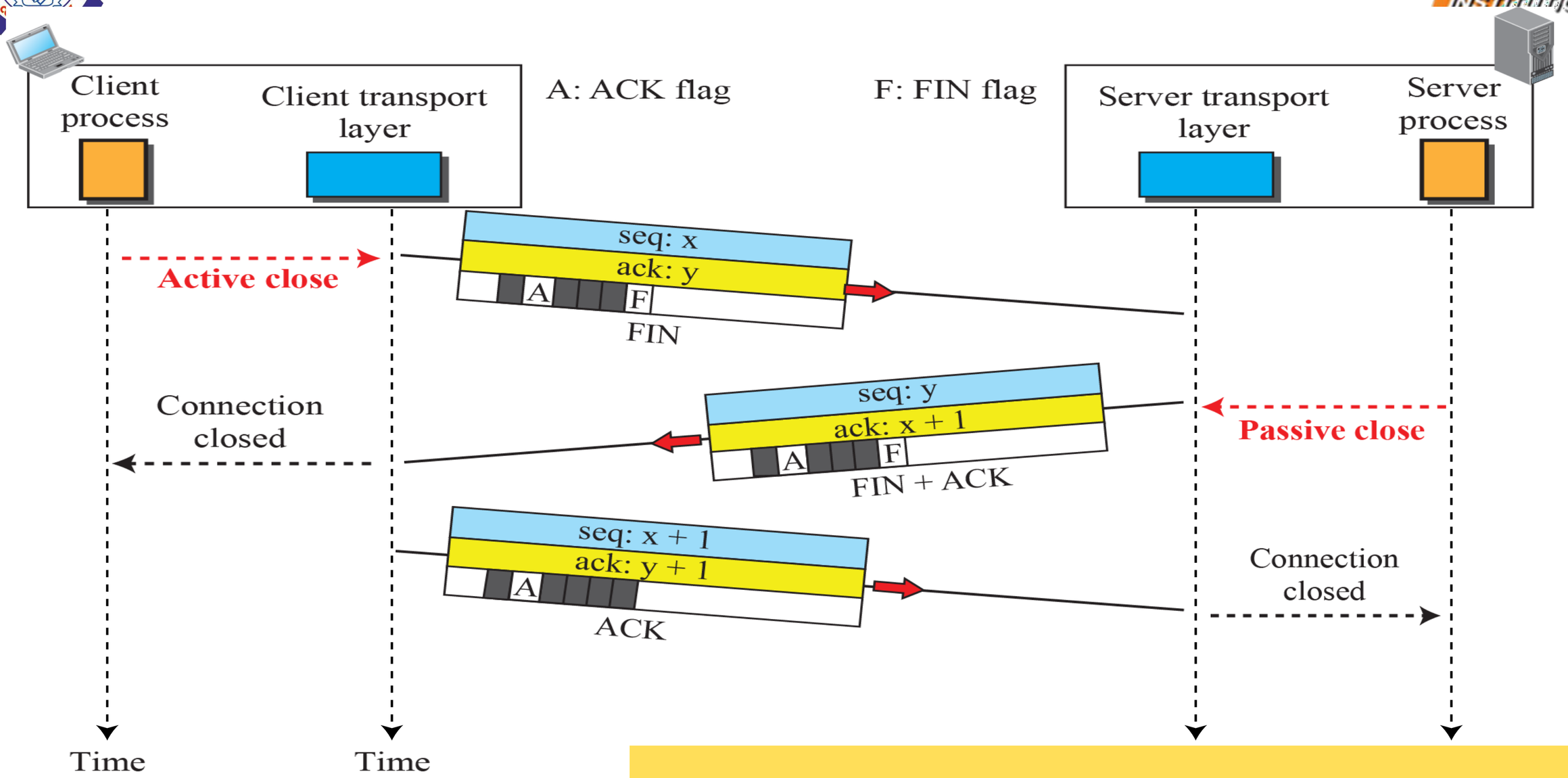
Data transfer



Three-Way Handshaking- connection termination

1. The client TCP, after receiving a close command from the client process, sends the first segment, a FIN segment in which the FIN flag is set
2. The server TCP, after receiving the FIN segment, informs its process of the situation and sends the second segment, a FIN + ACK segment, to confirm the receipt of the FIN segment from the client and at the same time to announce the closing of the connection in the other direction
3. The client TCP sends the last segment, an ACK segment, to confirm the receipt of the FIN segment from the TCP server. This segment contains the acknowledgment number, which is one plus the sequence number received in the FIN segment from the server.

Connection termination using three-way handshaking



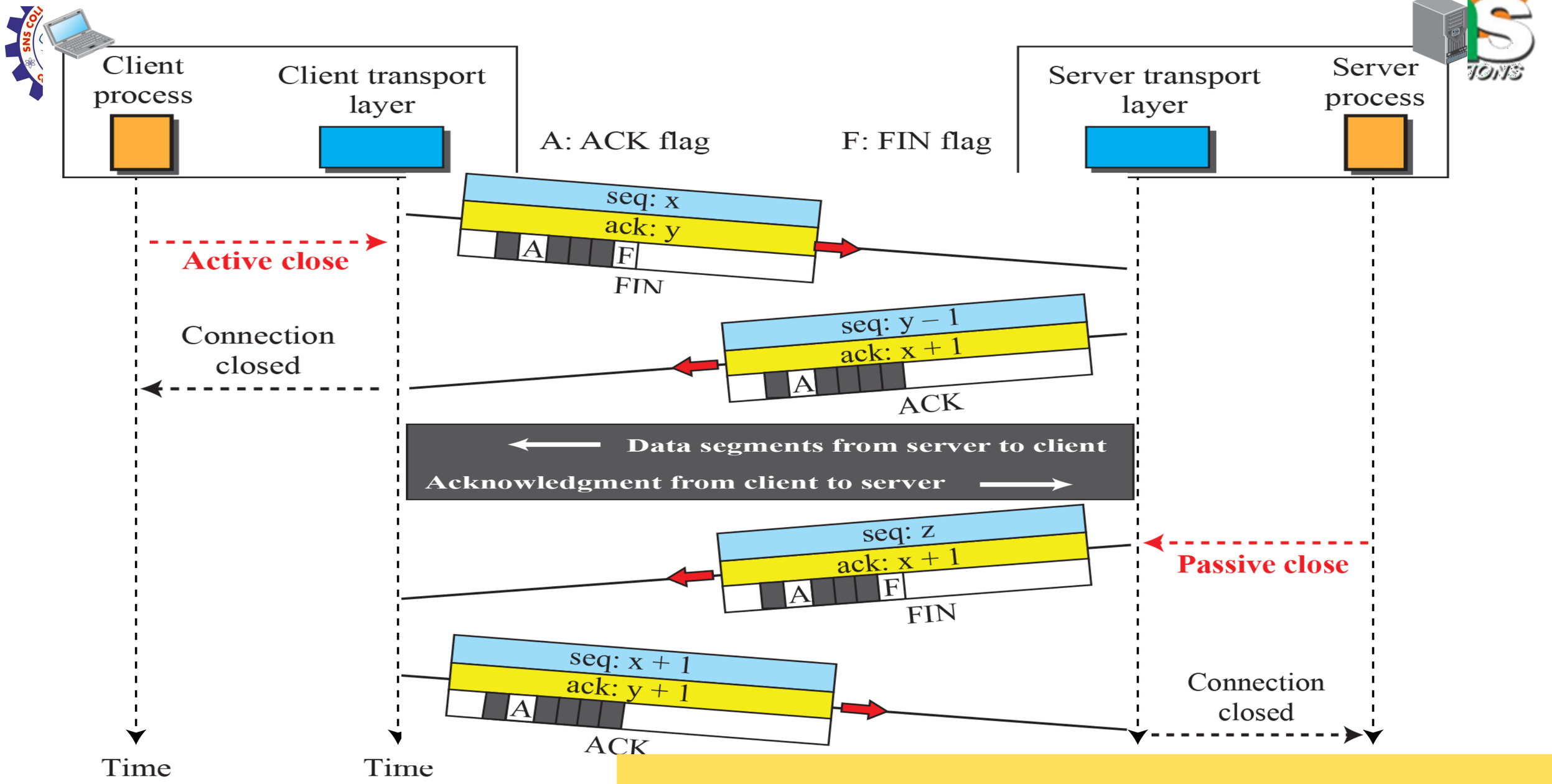
Half-Close

In TCP, one end can stop sending data while still receiving data. This is called a ***halfclose.***

Either the server or the client can issue a half-close request. It can occur when the server needs all the data before processing can begin. A good example is sorting.

the server-to-client direction must remain open to return the sorted data. The server, after receiving the data, still needs time for sorting

Half-close


Client process
Client transport layer
A: ACK flag
F: FIN flag
Server transport layer
Server process
seq: x
ack: y
A
F
FIN
Active close
Connection closed
seq: y – 1
ack: x + 1
A
ACK
Data segments from server to client
Acknowledgment from client to server
seq: z
ack: x + 1
A
F
FIN
Passive close
seq: x + 1
ack: y + 1
A
ACK
Connection closed
Time
Time

Assessment

a) List TCP services.
 b) What is TCP?
c) List the applications of TCP?
d) Discuss about TCP connection

Reference

TEXT BOOKS

Behrouz A. Forouzan, Data Communications and Networking, Fifth Edition TMH, 2013.

REFERENCES

1. William Stallings, Data and Computer Communications, Tenth Edition, Pearson Education, 2013.
2. Andrew Tanenbaum, Computer Networks, Fifth Edition, Pearson (5th Edition) Education, 2013.
3. James F. Kurose, Keith W. Ross, Computer Networking, A Top-Down Approach Featuring the Internet, Sixth Edition, Pearson Education, 2013.
4. Larry L. Peterson, Bruce S. Davie, Computer Networks: A Systems Approach, Fifth Edition, Morgan Kaufmann Publishers Inc., 2012.