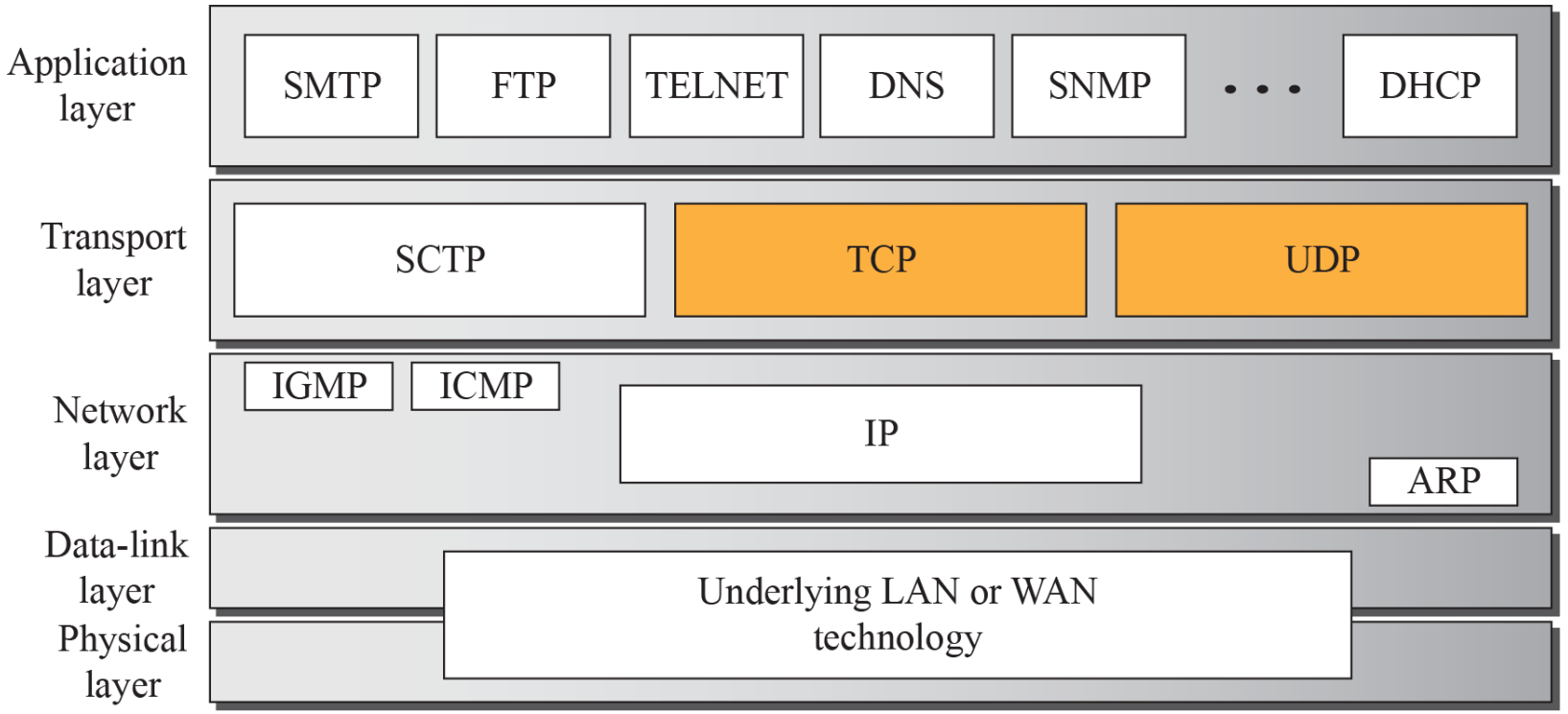
Transport Layer Protocols

INTRODUCTION

- *The main task of the transport layer is to provide reliable,cost effective data transport from the source machine to destination machine, independent of the physical network.*

Position of transport-layer protocols in the TCP/IP protocol suite



Port Numbers

- ***Port numbers(16 bit integers between 0 to 65,535) provide end-to-end addresses at the transport layer***

IANA Ranges

- Internet Assigned Number Authority had divided the port number into three ranges

-**Well Known Ports:** ranging from 0 to 1023 that are assigned and controlled by IANA

- **Registered Ports:** ranging from1024 to 49,151 that are not assigned or controlled by IANA. They can only be from regestered with IANA to prevent duplication.

-**Dynamic Ports:** The ports ranging 49,152 to 65,535 are *neither controlled nor registered. They can be used by any Process* .

Socket Addresses(IP + Port number)

-Process to process delivery need two identifiers, IP address and port number at each end to make a connection .

- A Transport Layer protocol needs a pair of Socket addresses(client,server).
- IP address 200.23.56.8 port number 69

  Socket address 200.23.56.8 60
- The addressing mechanism allows multiplexing and demultiplexing by the transport layer

Some well-known ports used with UDP and TCP

Port	*Protocol*	*UDP*	*TCP*	*Description*
7	Echo	√		Echoes back a received datagram
9	Discard	√		Discards any datagram that is received
11	Users	√	√	Active users
13	Daytime	√	√	Returns the date and the time
17	Quote	√	√	Returns a quote of the day
19	Chargen	√	√	Returns a string of characters
20, 21	FTP		√	File Transfer Protocol
23	TELNET		√	Terminal Network
25	SMTP		√	Simple Mail Transfer Protocol
53	DNS	√	√	Domain Name Service
67	DHCP	√	√	Dynamic Host Configuration Protocol
69	TFTP	√		Trivial File Transfer Protocol
80	HTTP		√	Hypertext Transfer Protocol
111	RPC	√	√	Remote Procedure Call
123	NTP	√	√	Network Time Protocol
161, 162	SNMP		√	Simple Network Management Protocol

from

UDP

The User Datagram Protocol (UDP) is

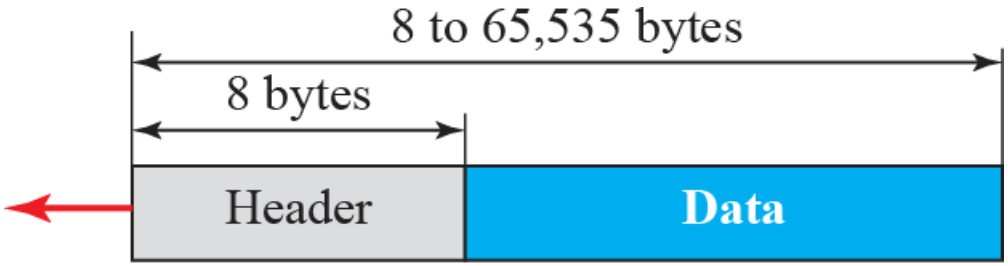
- *a connectionless, unreliable transport protocol*
- *not much care about reliability, very limited error checking.*
- *UDP is a very simple protocol using a minimum of overhead.*
- *Is a convenient protocol for multimedia and multicasting applications.*

User Datagram

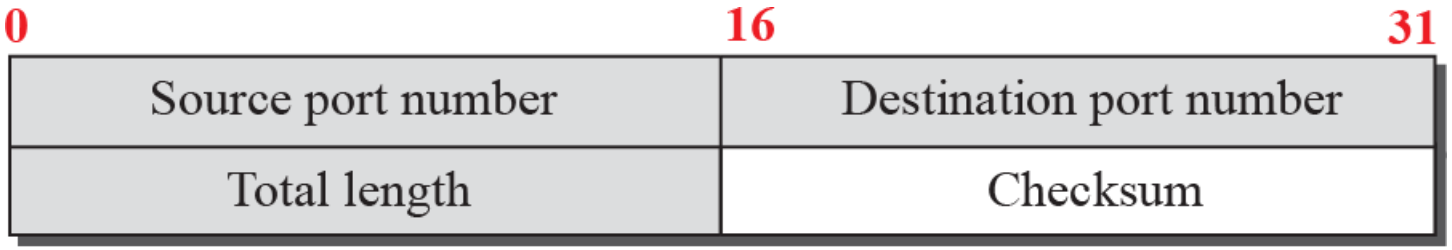
- *UDP packets, called user datagrams, have a fixed-size header of 8 bytes made of four fields, each of 2 bytes (16 bits)*

- *Figure shows the format of a user datagram*
  - *The first two fields define the source and destination port numbers.*
  - *The third field defines the total length of the user datagram, header plus data*
  - *The last field can carry the checksum*

User datagram packet format



a. UDP user datagram



b. Header format

Example 24.1

the contents of a UDP header in hexadecimal format.

CB84000D001C001C

a. What is the source port number?
b. What is the destination port number?
c. What is the total length of the user datagram?
d. What is the length of the data?
e. Is the packet directed from a client to a server or vice versa?
f. What is the client process?

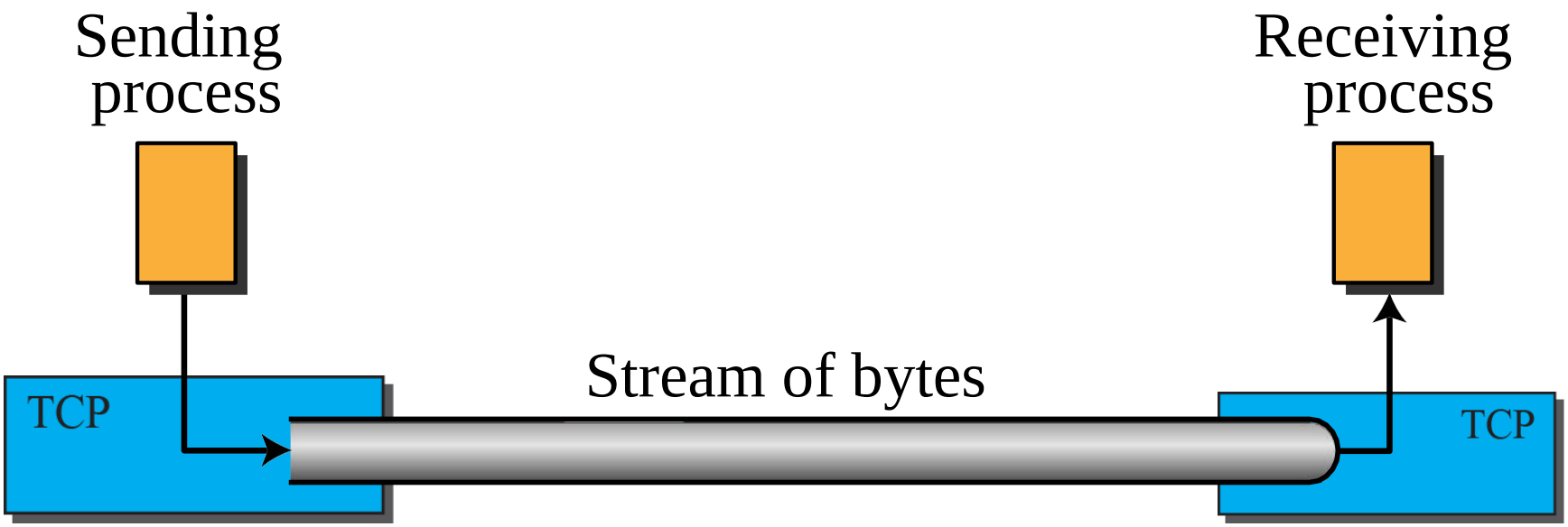
Example 24.1 (continued)

a. The source port number is the first four hexadecimal digits $(CB84)_{16}$ or 52100
b. The destination port number is the second four hexadecimal digits $(000D)_{16}$ or 13.
c. The third four hexadecimal digits $(001C)_{16}$ define the length of the whole UDP packet as 28 bytes.
d. The length of the data is the length of the whole packet minus the length of the header, or 28 − 8 = 20 bytes.
e. Since the destination port number is 13 (well-known port), the packet is from the client to the server.
f. The client process is the Daytime (see Table 3.1).

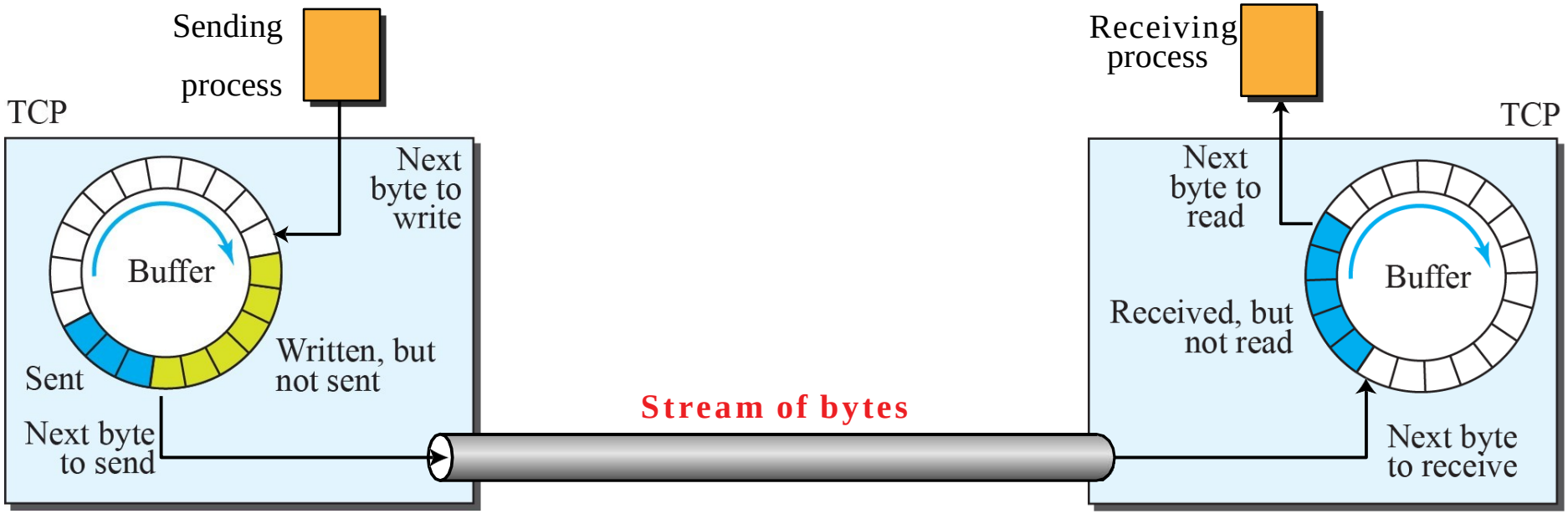
TCP

- *is a connection-oriented*
- *reliable protocol*
- *explicitly defines connection establishment, data transfer, and connection teardown phases*

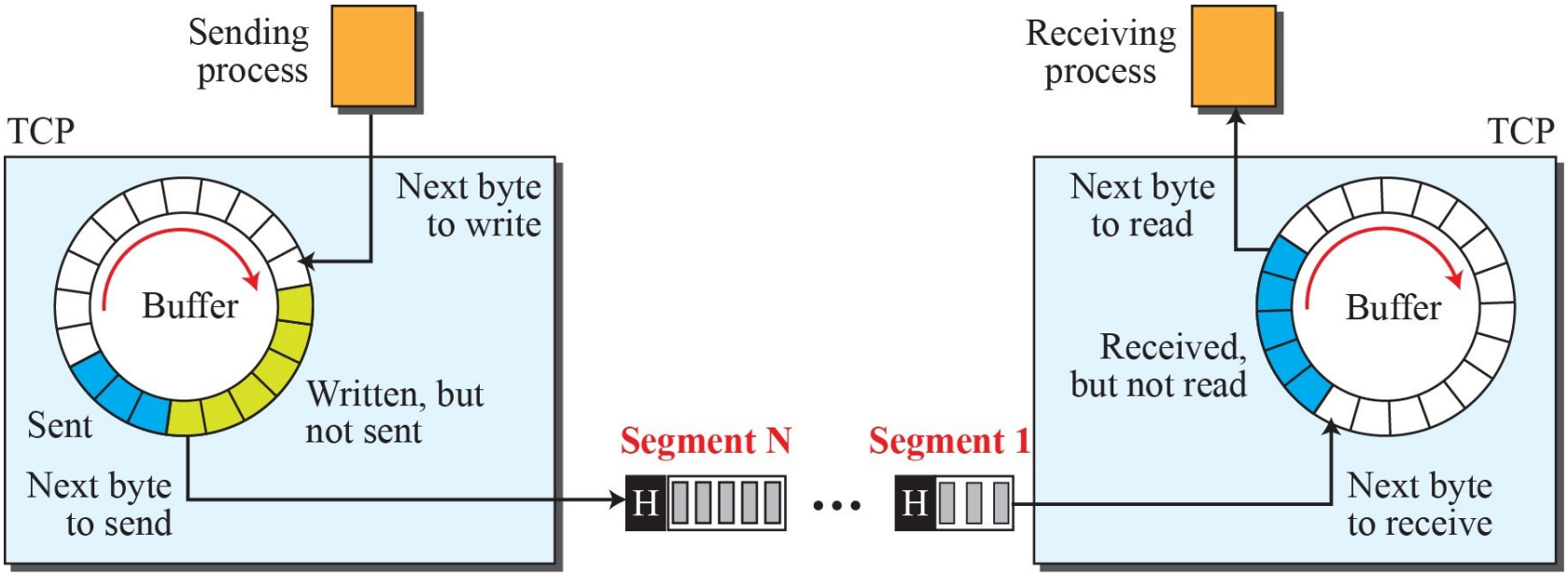
Stream delivery



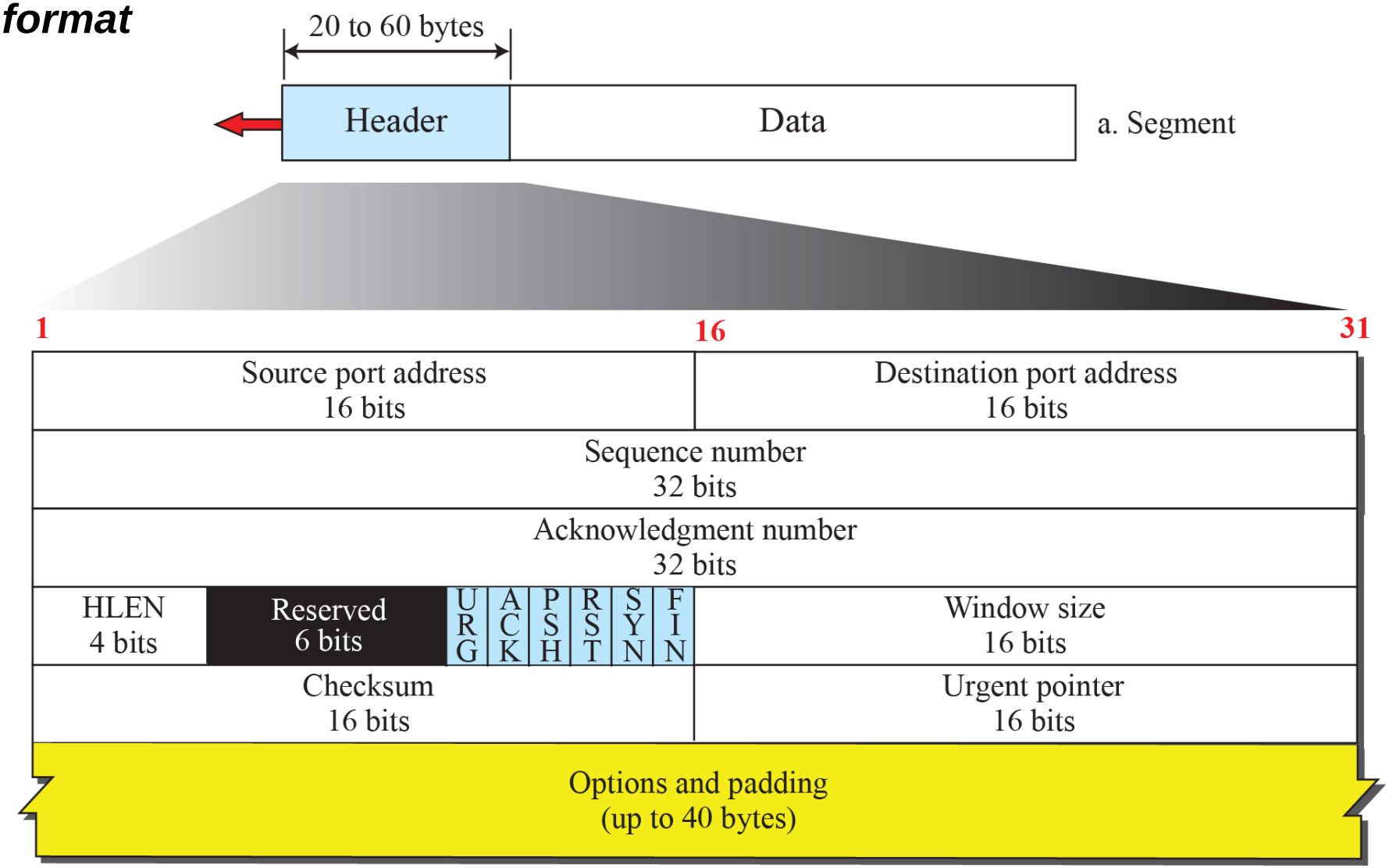
Sending and receiving buffers



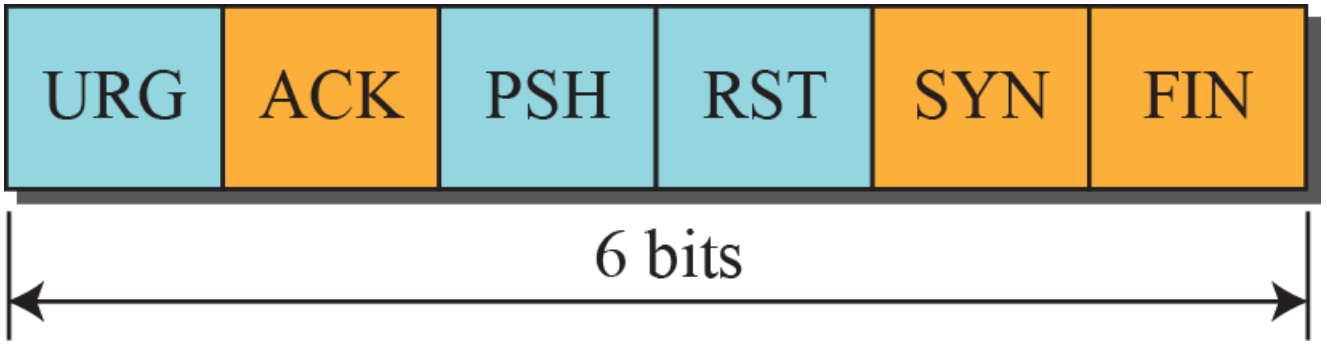
TCP segments



TCP segment format

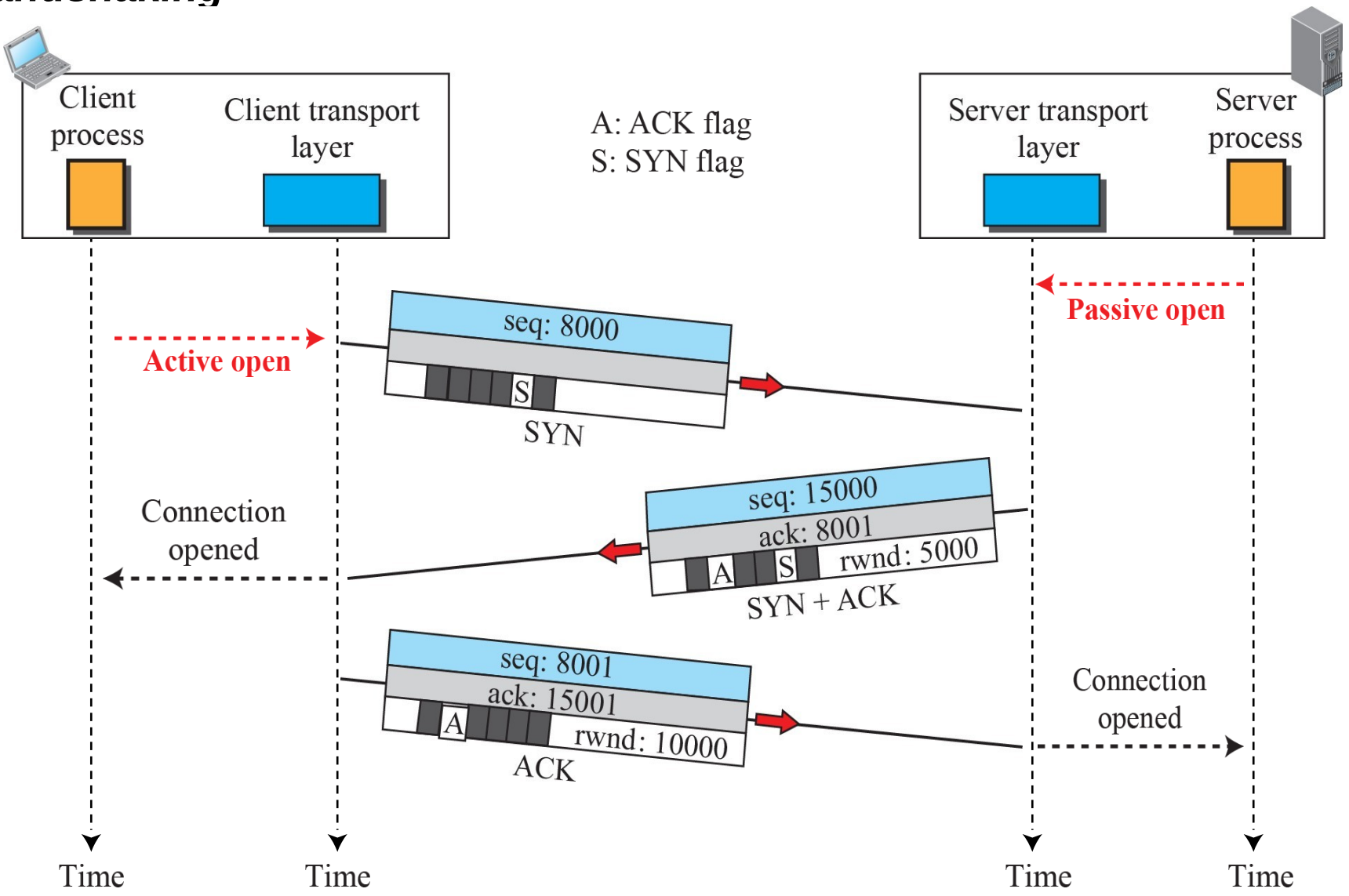


Control field

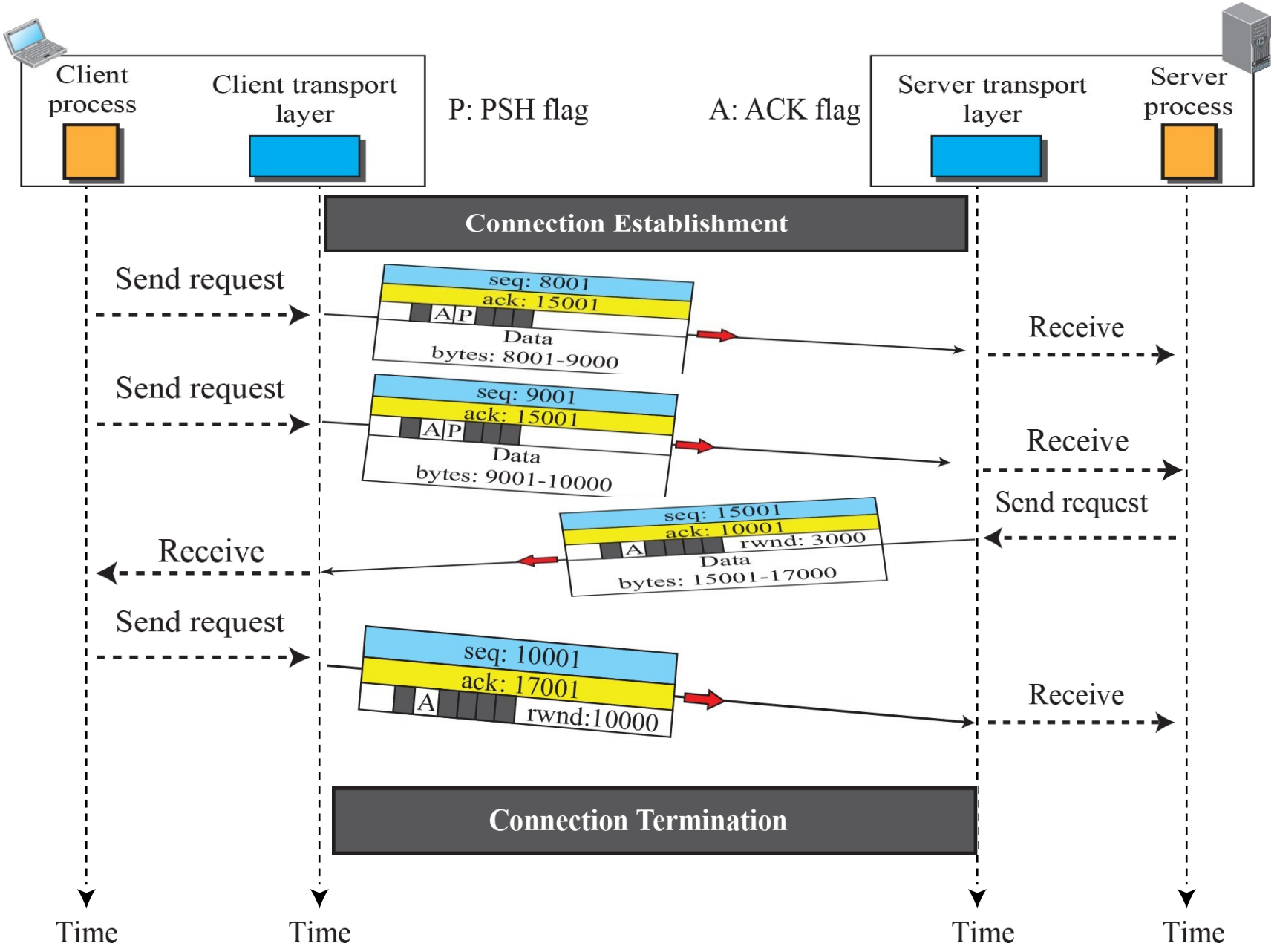


URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection

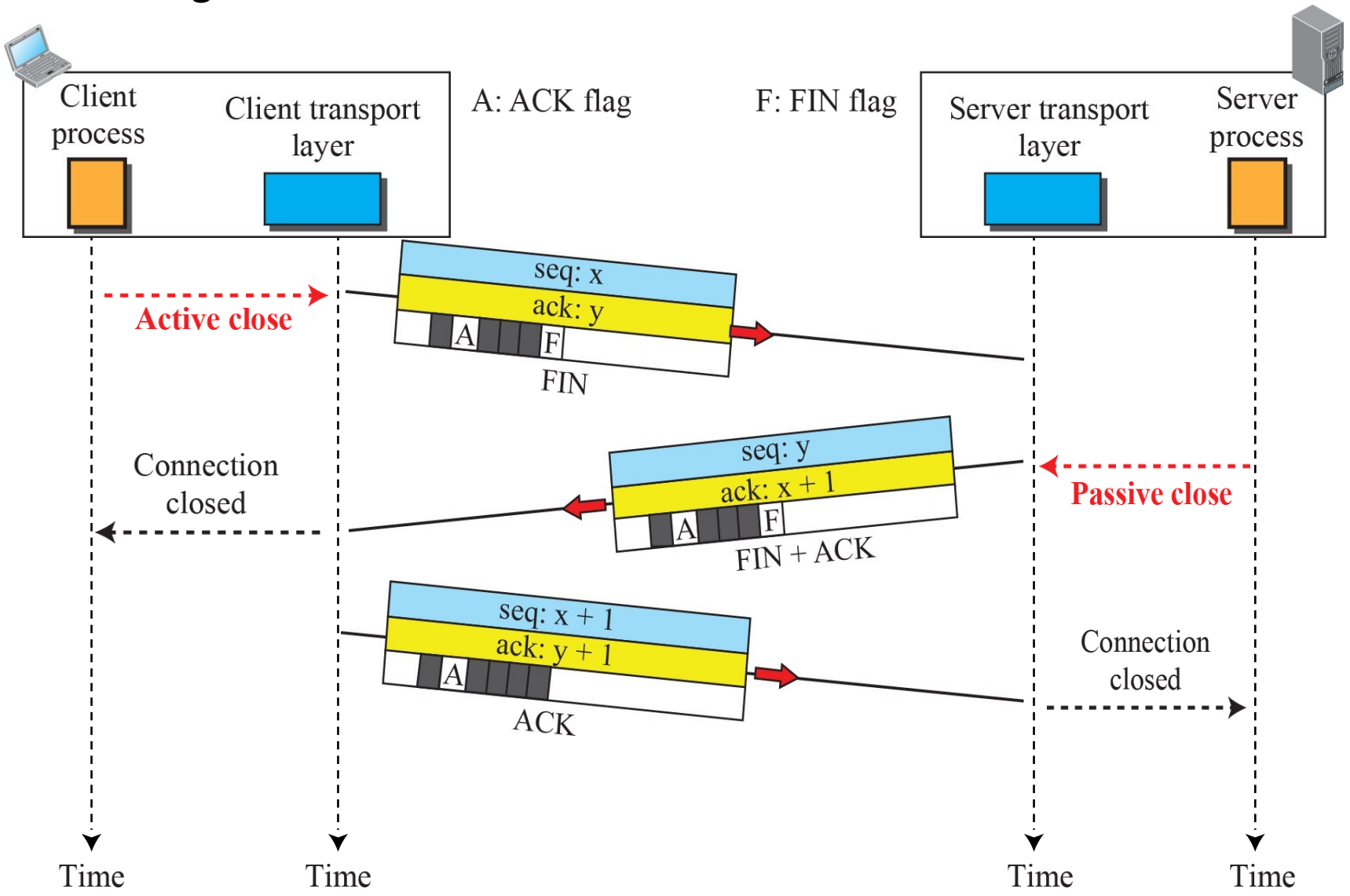
Connection establishment using three-way handshaking



Data transfer



Connection termination using three-way handshaking



Flow Control

- *As discussed before, flow control balances the rate a producer creates data with the rate a consumer can use the data*

- *We assume that the logical channel between the sending and receiving TCP is error-free.*

- *TCP uses a sliding window protocol to accomplish flow control in which both hosts use a window for each connection .*

Error Control

Error control in TCP is to ensure reliability in transport Layer protocol. The error control mechanisms in TCP detects,

1)Corrupted segments. 3)Out-of-order segments

2)Lost segments 4)Duplicated segments

TCP uses simple tools to detect the above errors,

1)Checksum

2)Acknowledgement

3)Time-out