Chapter 24

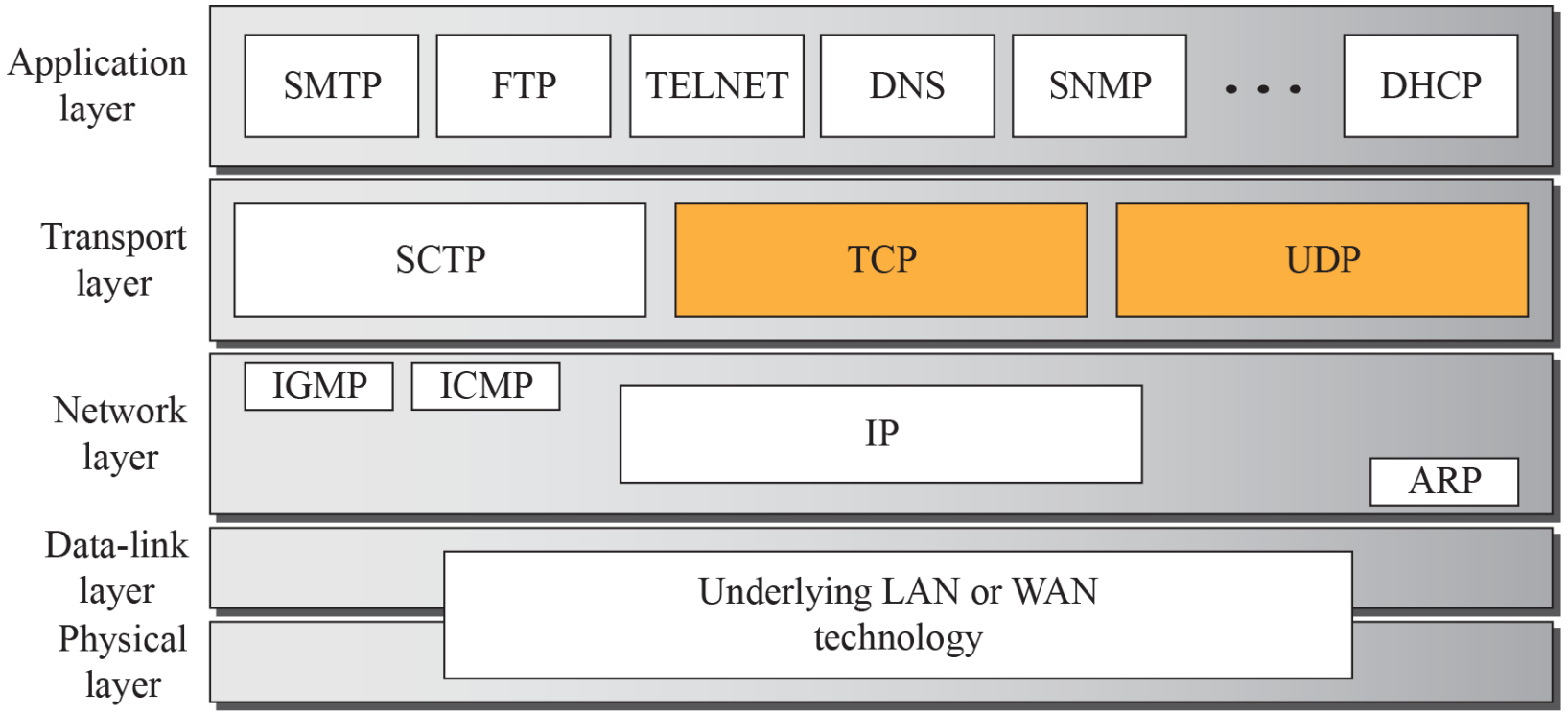
Transport Layer Protocols

INTRODUCTION

- concentrate on the transport protocols in the Internet in this chapter.

- figure on next slide shows the position of three protocols in the TCP/IP protocol suite.

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



Services

Each protocol provides a different type of service and should be used appropriately.

Port Numbers

- *Port numbers provide end-to-end addresses at the transport layer*

- *allow multiplexing and demultiplexing, just as IP addresses do at the network layer*

- *Table gives some common port numbers*

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

UDP

The User Datagram Protocol (UDP) is

- *a connectionless, unreliable transport protocol*

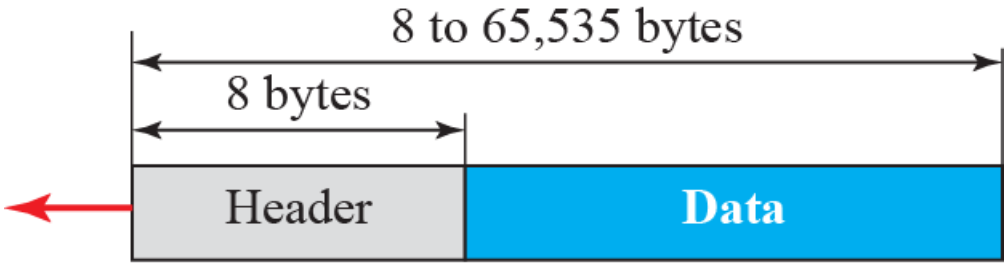
- *UDP is a very simple protocol using a minimum of overhead.*

User Datagram

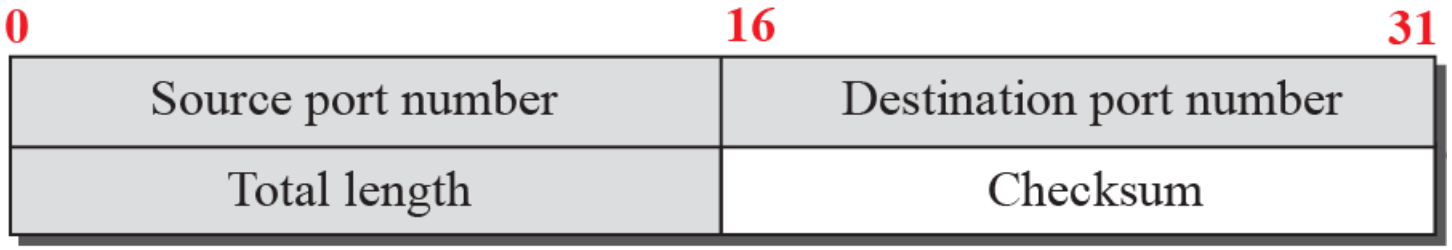
- *UDP packets, called user datagrams, have a fixed-size header of 8 bytes made of four fields, each of 2 bytes (16 bits)*

- *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 following is 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)

Solution

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.

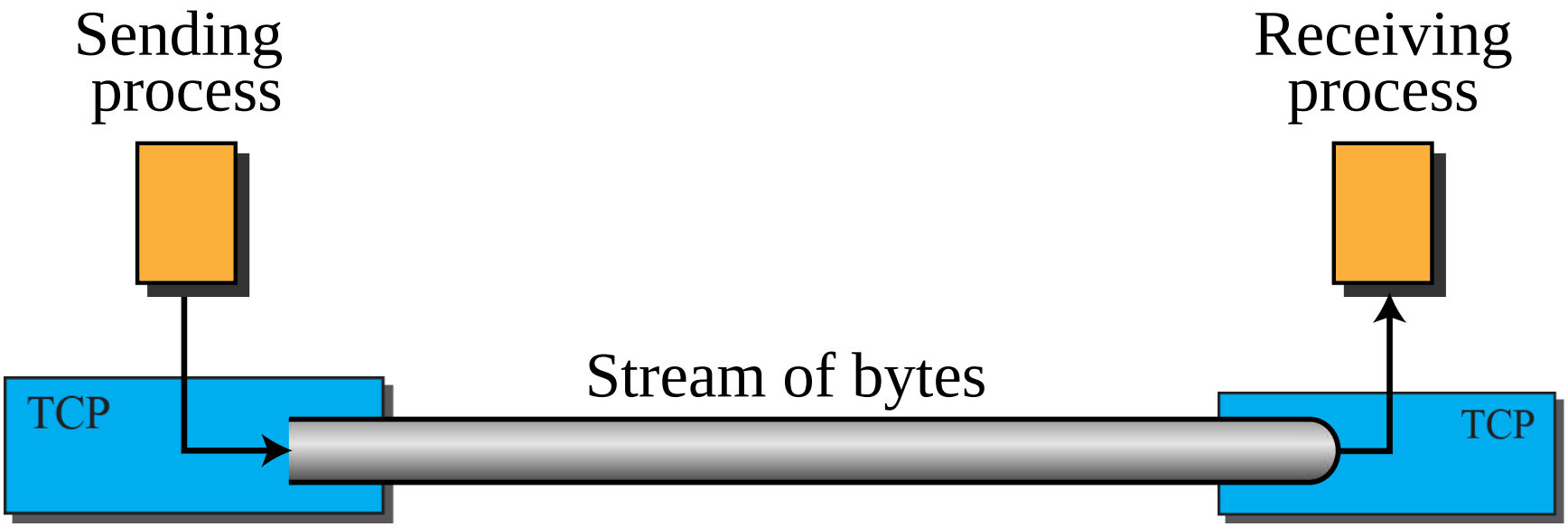
e. Since the destination port number is 13 (well-known port), the packet is from the client to the server.

f. The client process is the Daytime (see Table 3.1).

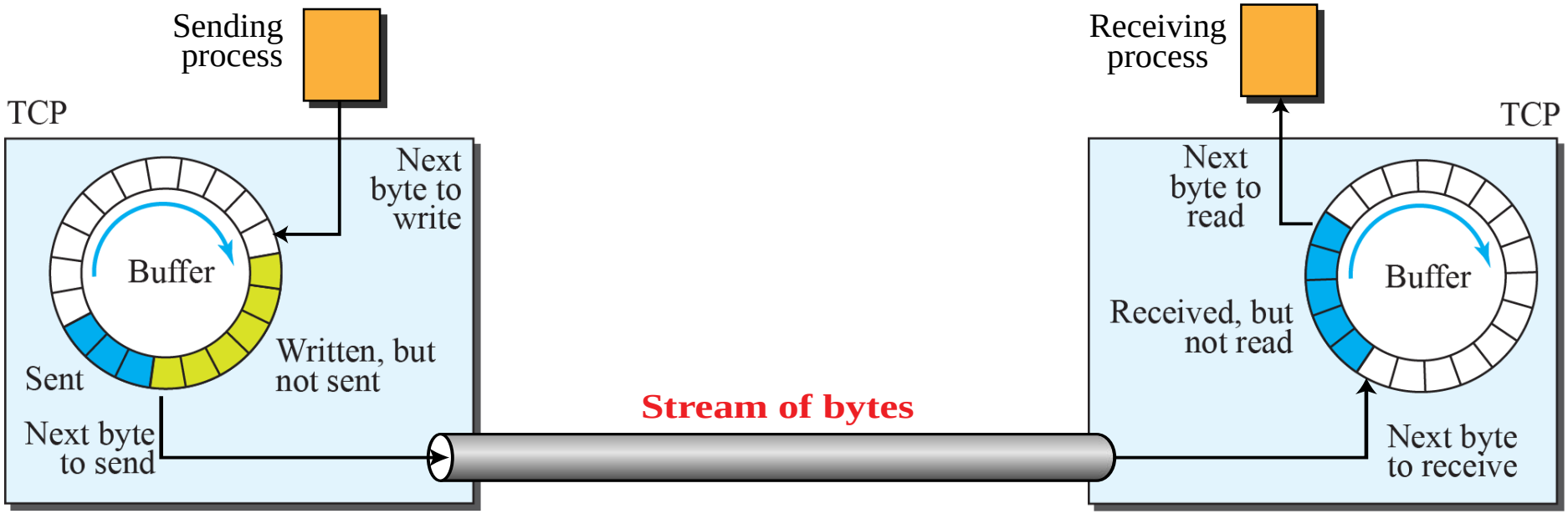
TCP

- *is a connection-oriented*
- *reliable protocol*
- *explicitly defines connection establishment, data transfer, and connection teardown phases*
- *uses a combination of GBN and SR protocols to provide reliability*

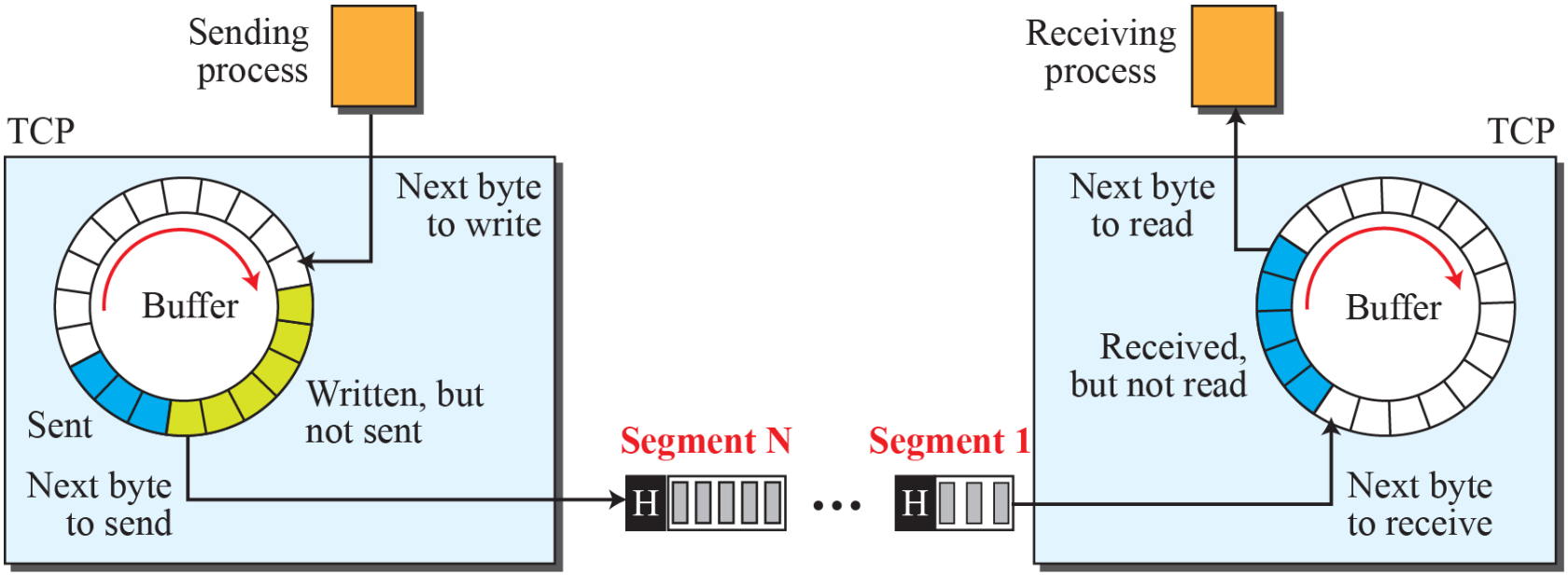
Stream delivery



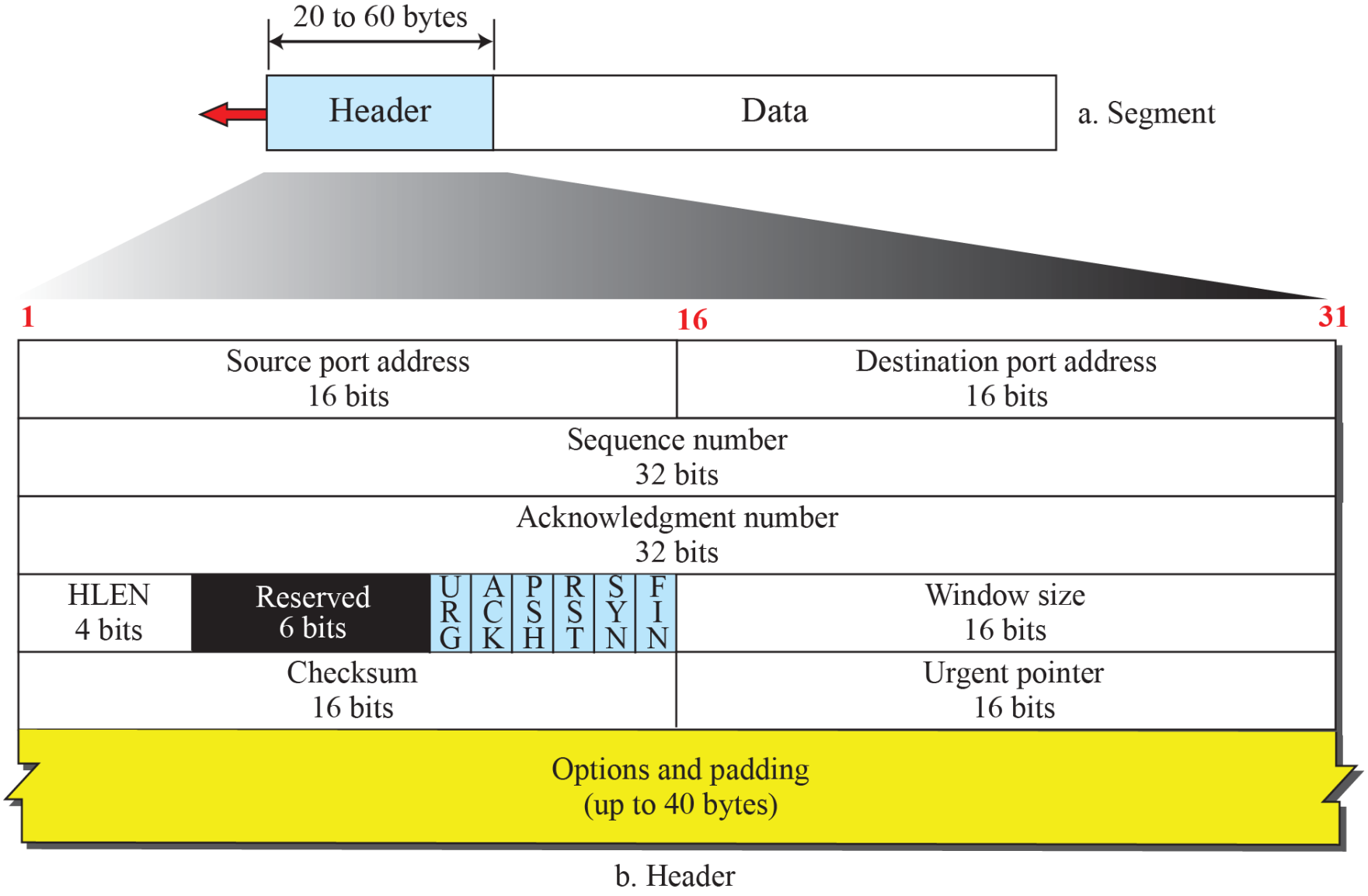
Sending and receiving buffers



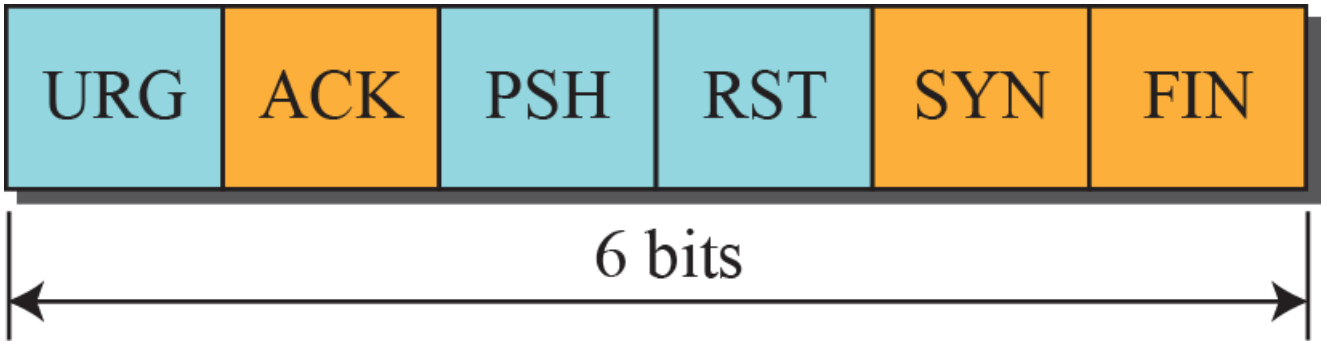
TCP segments



TCP segment format

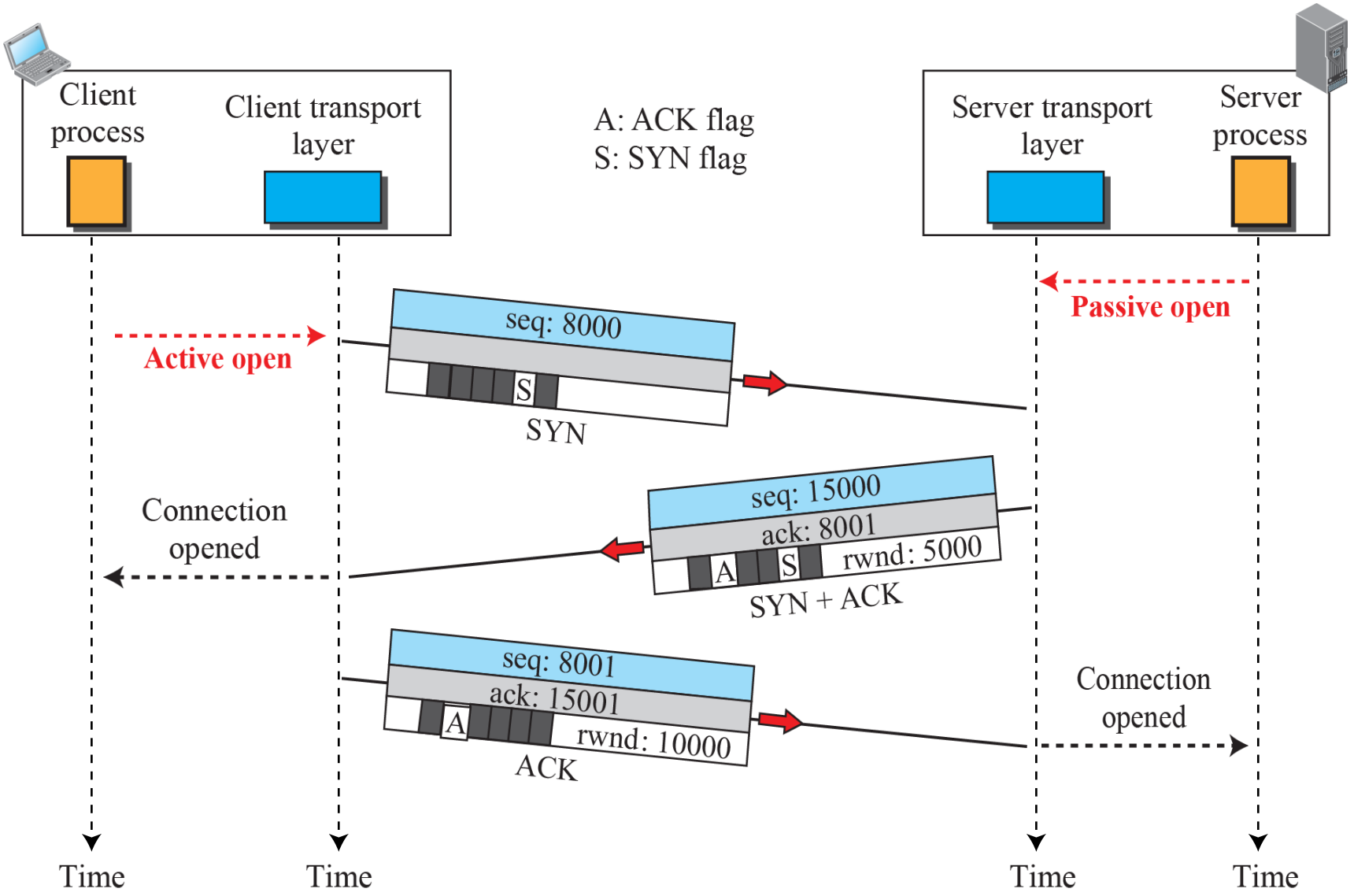


Control field

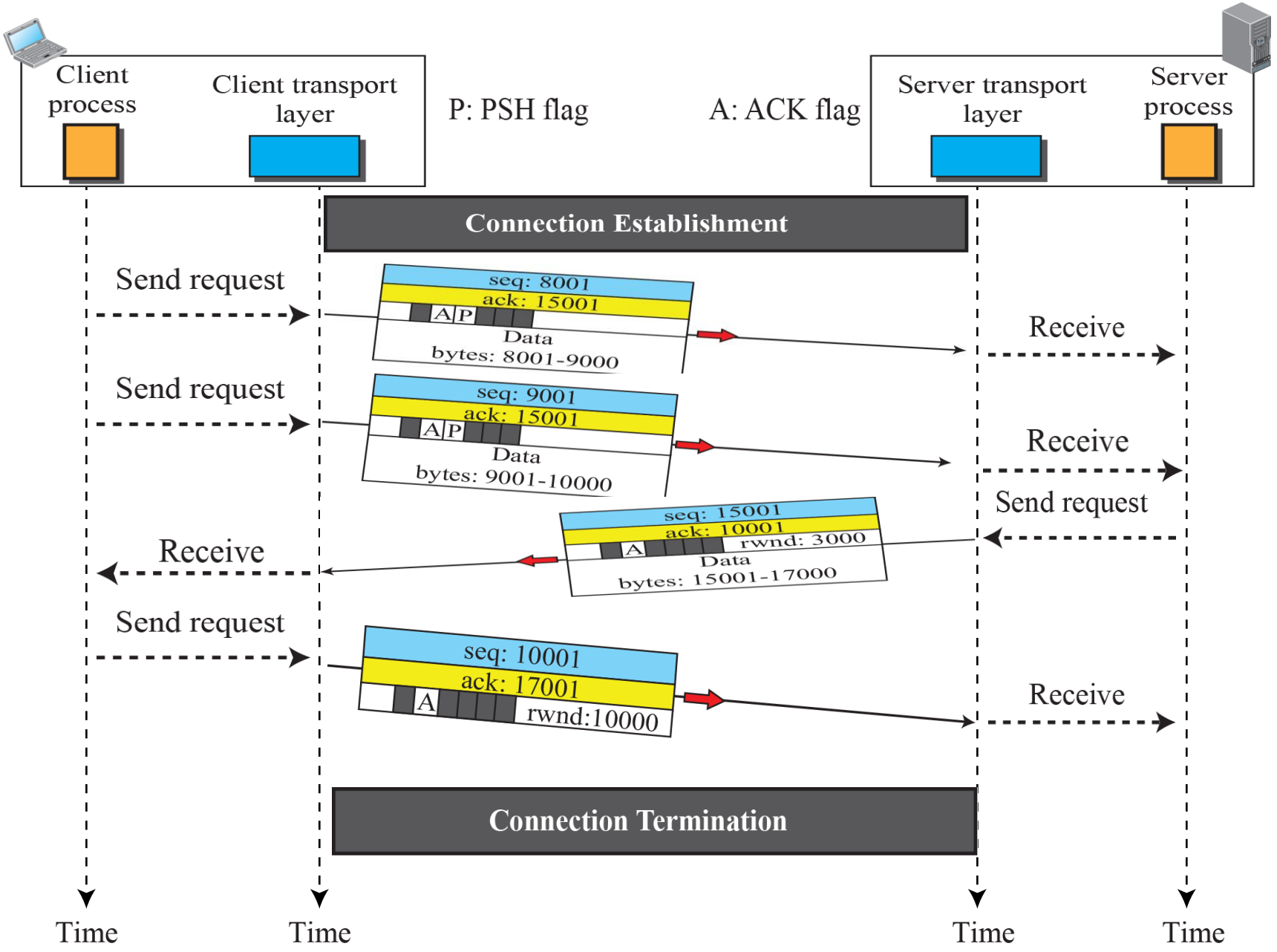


URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection

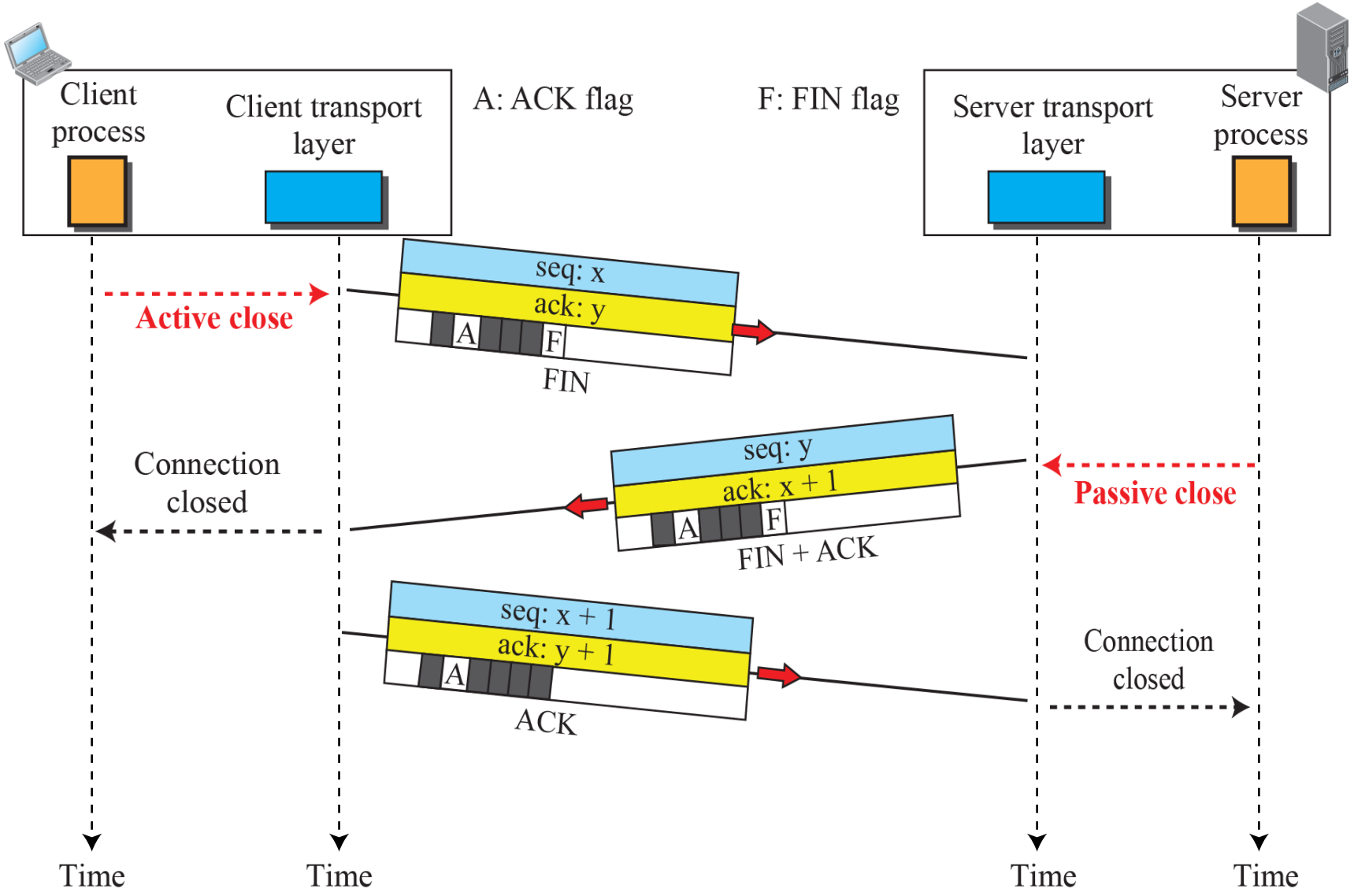
Connection establishment using three-way handshaking



Data transfer



Connection termination using three-way handshaking

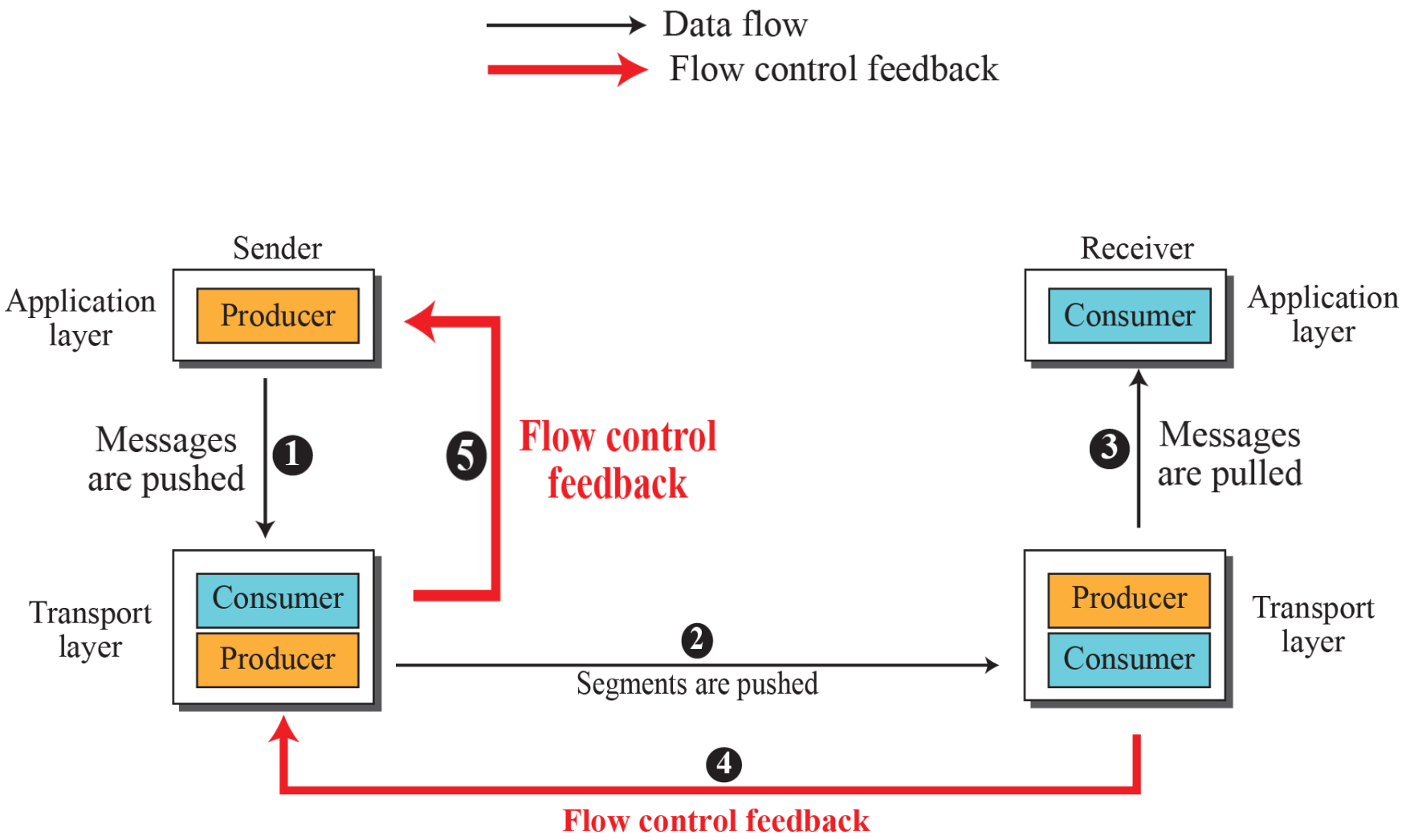


Flow Control

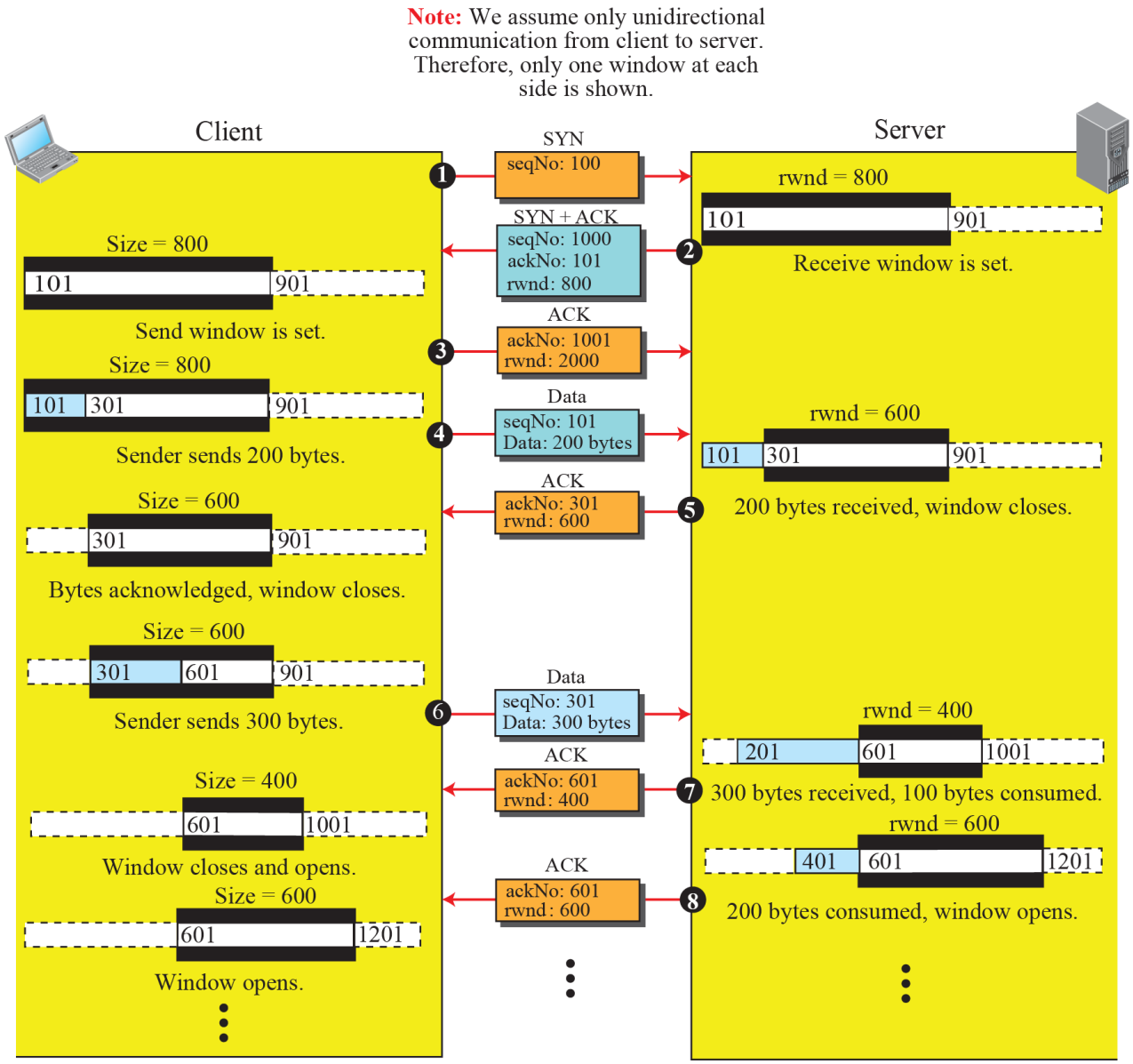
- *As discussed before, flow control balances the rate a producer creates data with the rate a consumer can use the data*

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

Data flow and flow control feedbacks in TCP



An example of flow control

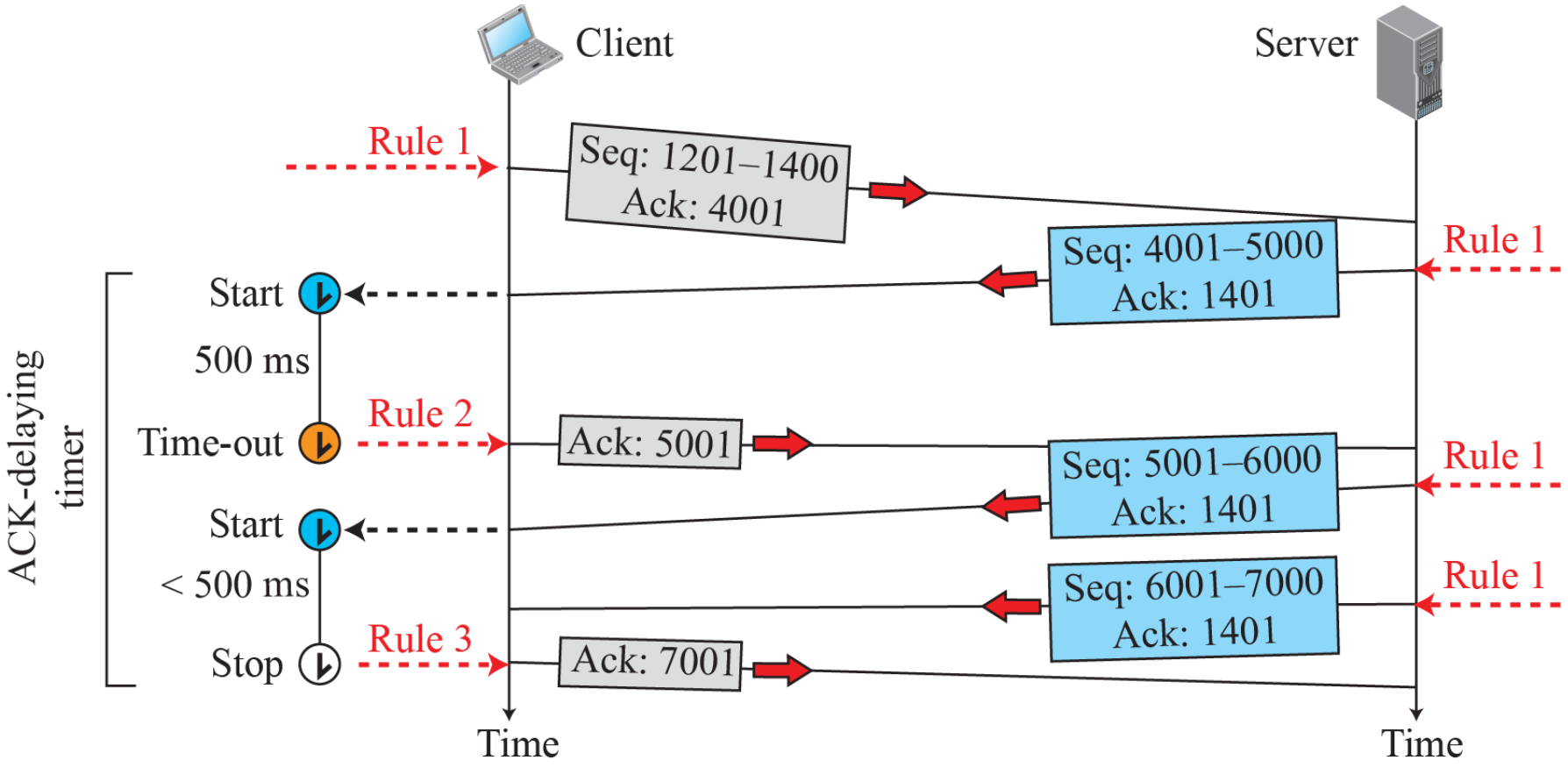


Error Control

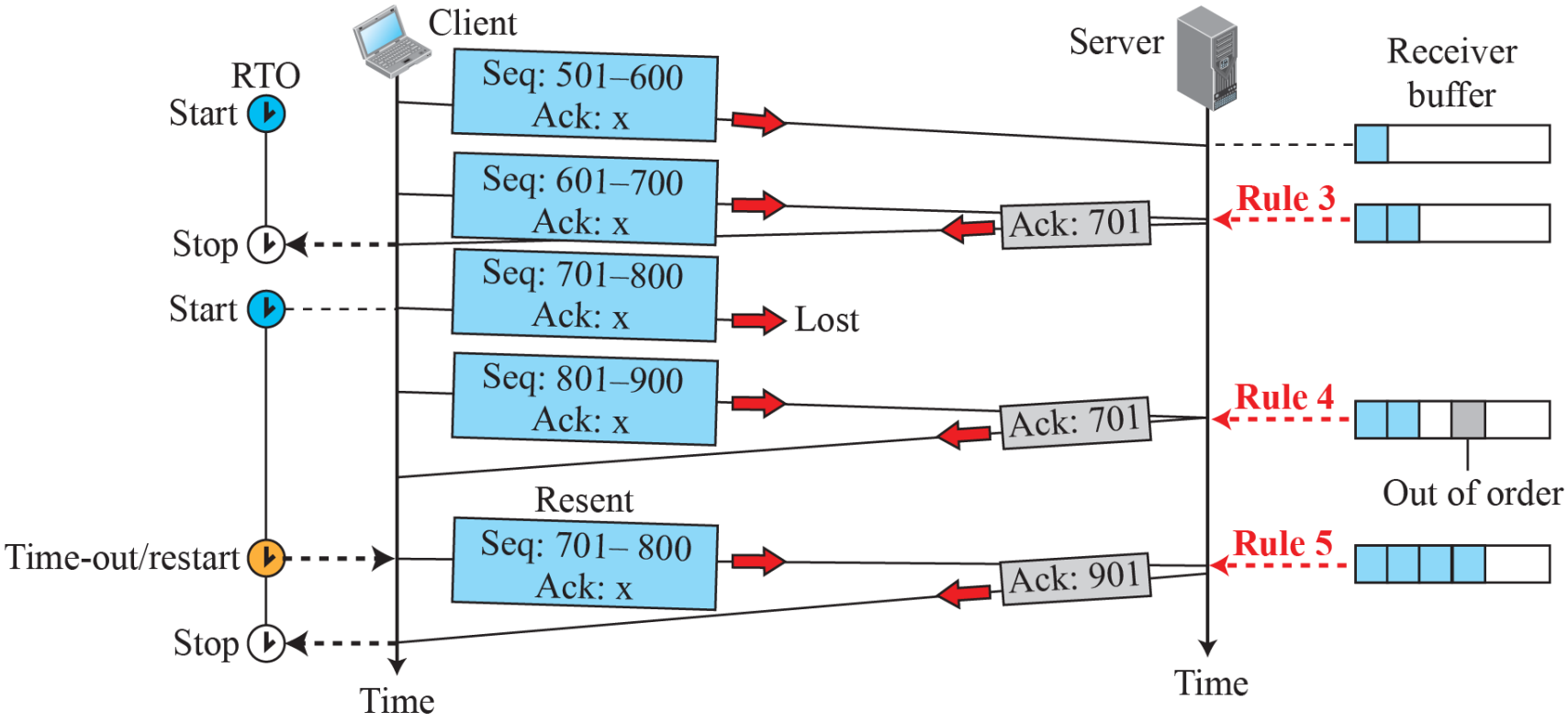
TCP is a reliable transport-layer protocol

→ This means that an application program that delivers a stream of data to TCP relies on TCP to deliver the entire stream to the application program on the other end in order, without error, and without any part lost or duplicated.

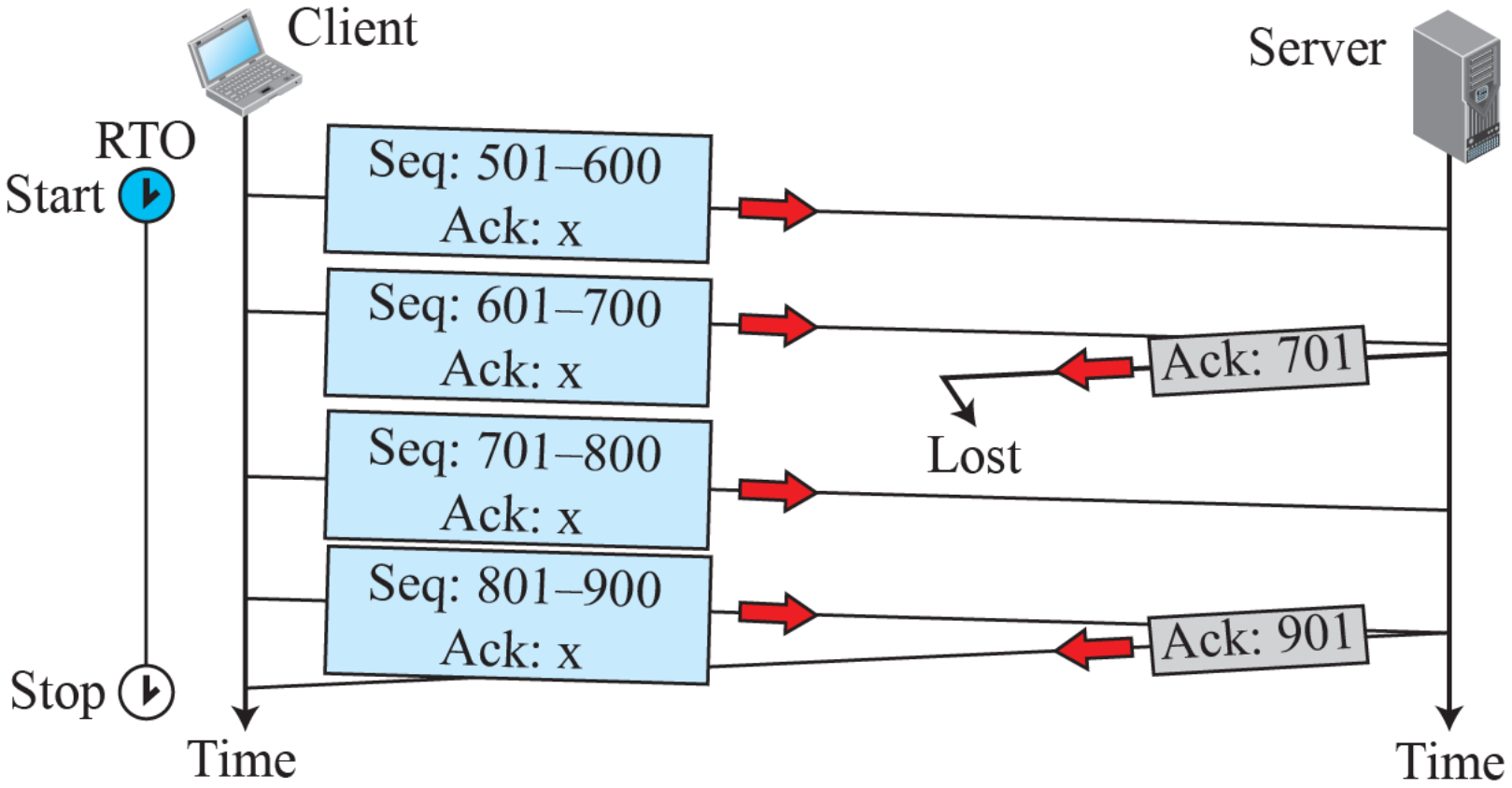
Normal operation



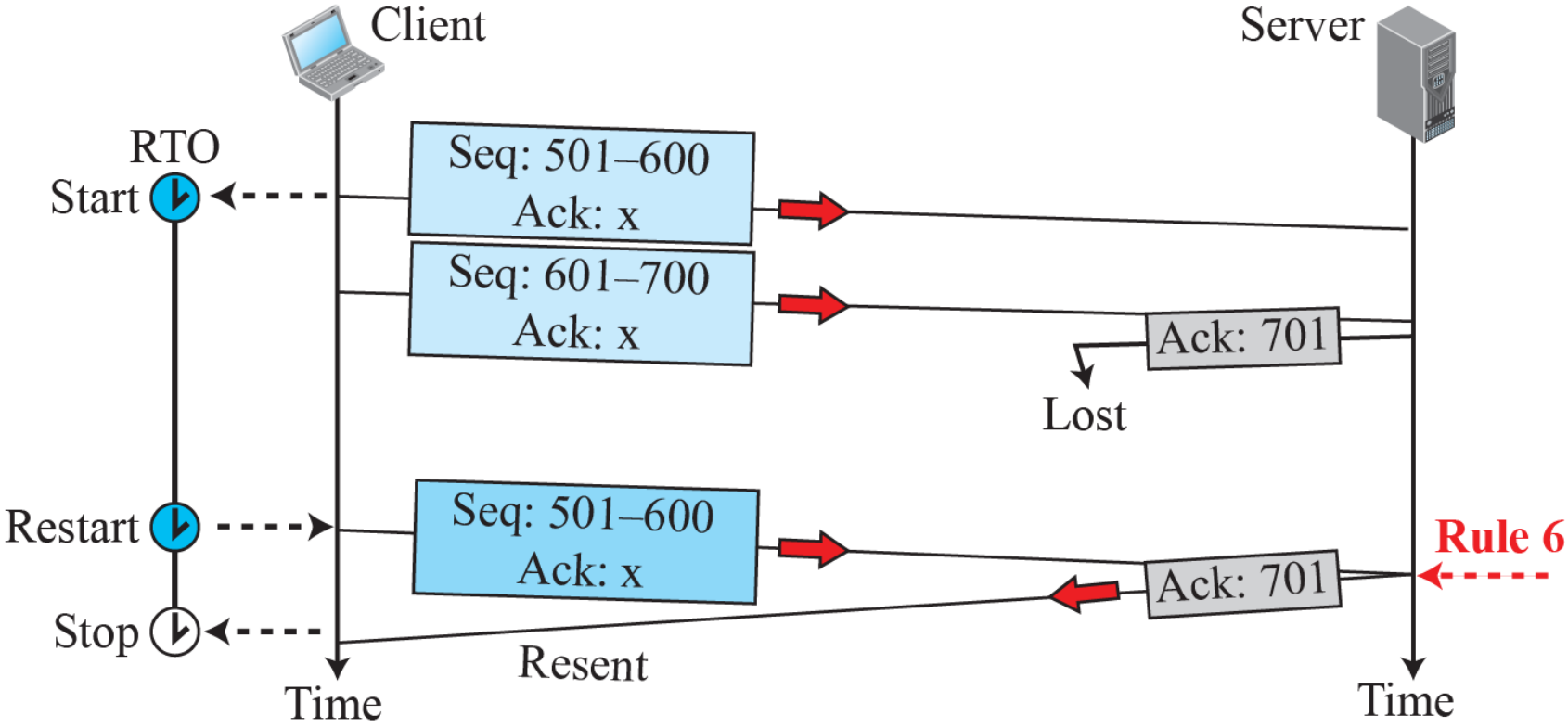
Lost segment



Lost acknowledgment



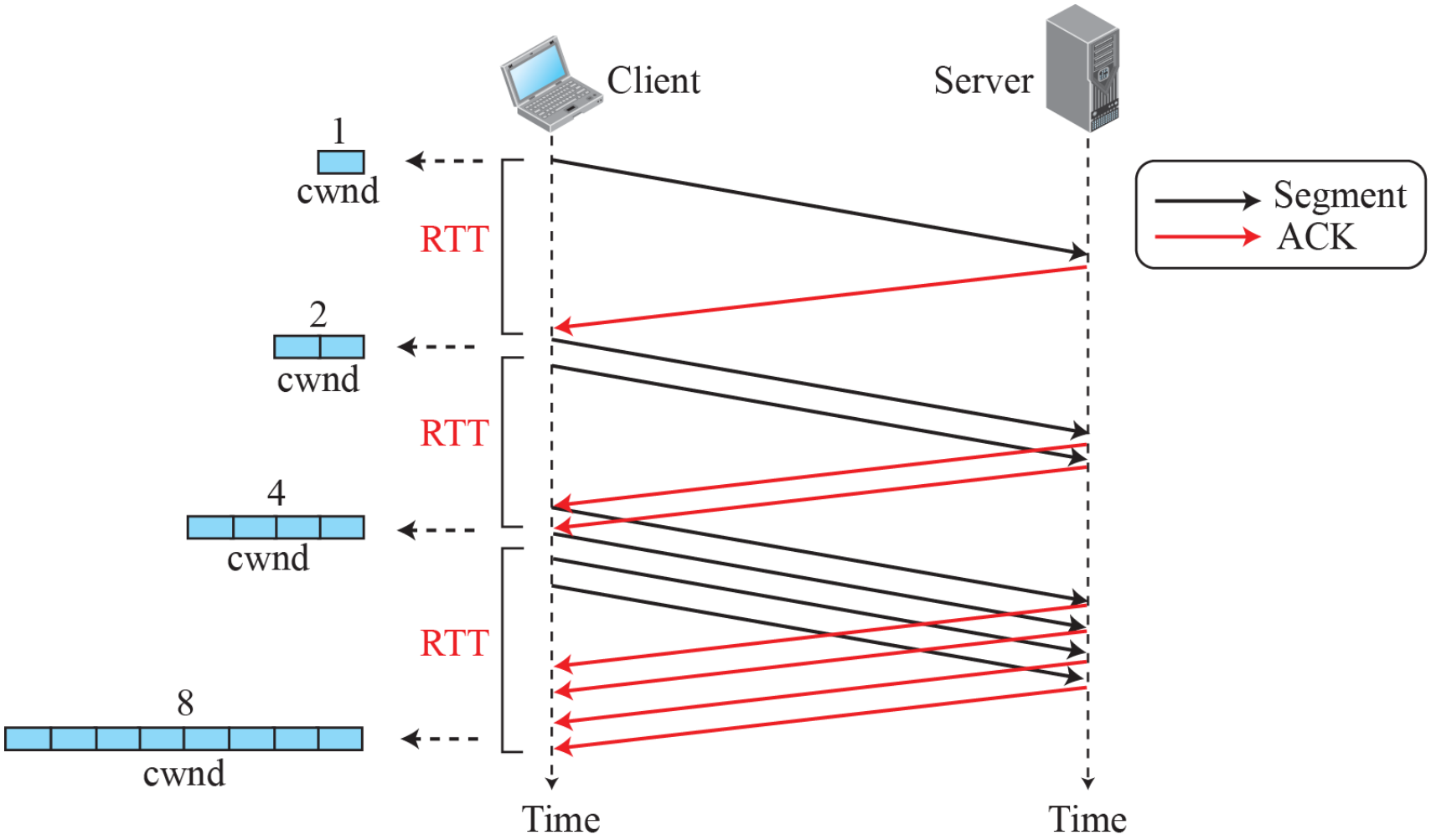
Lost acknowledgment corrected by resending a segment



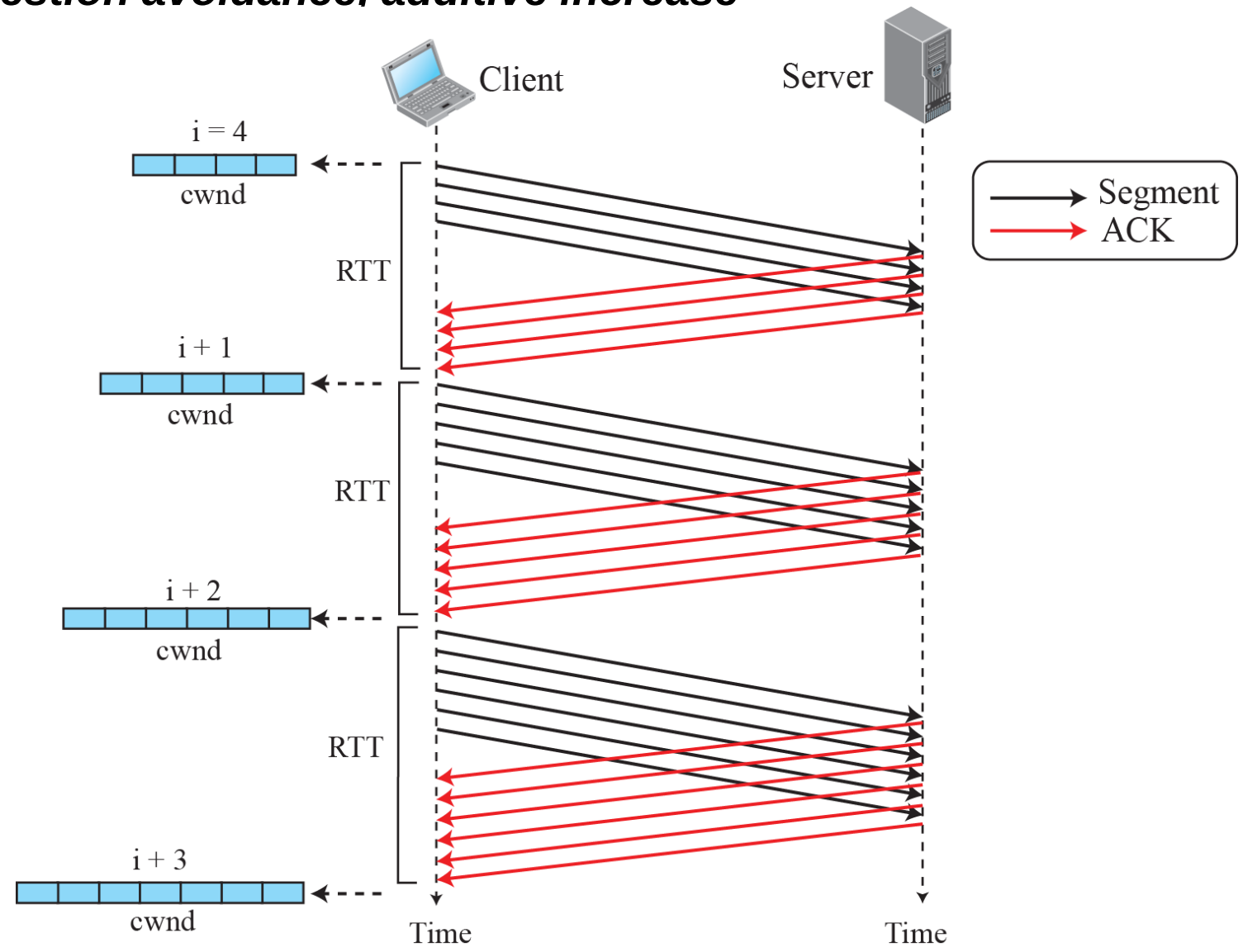
TCP Congestion Control

TCP uses different policies to handle the congestion in the network. We describe these policies in this section.

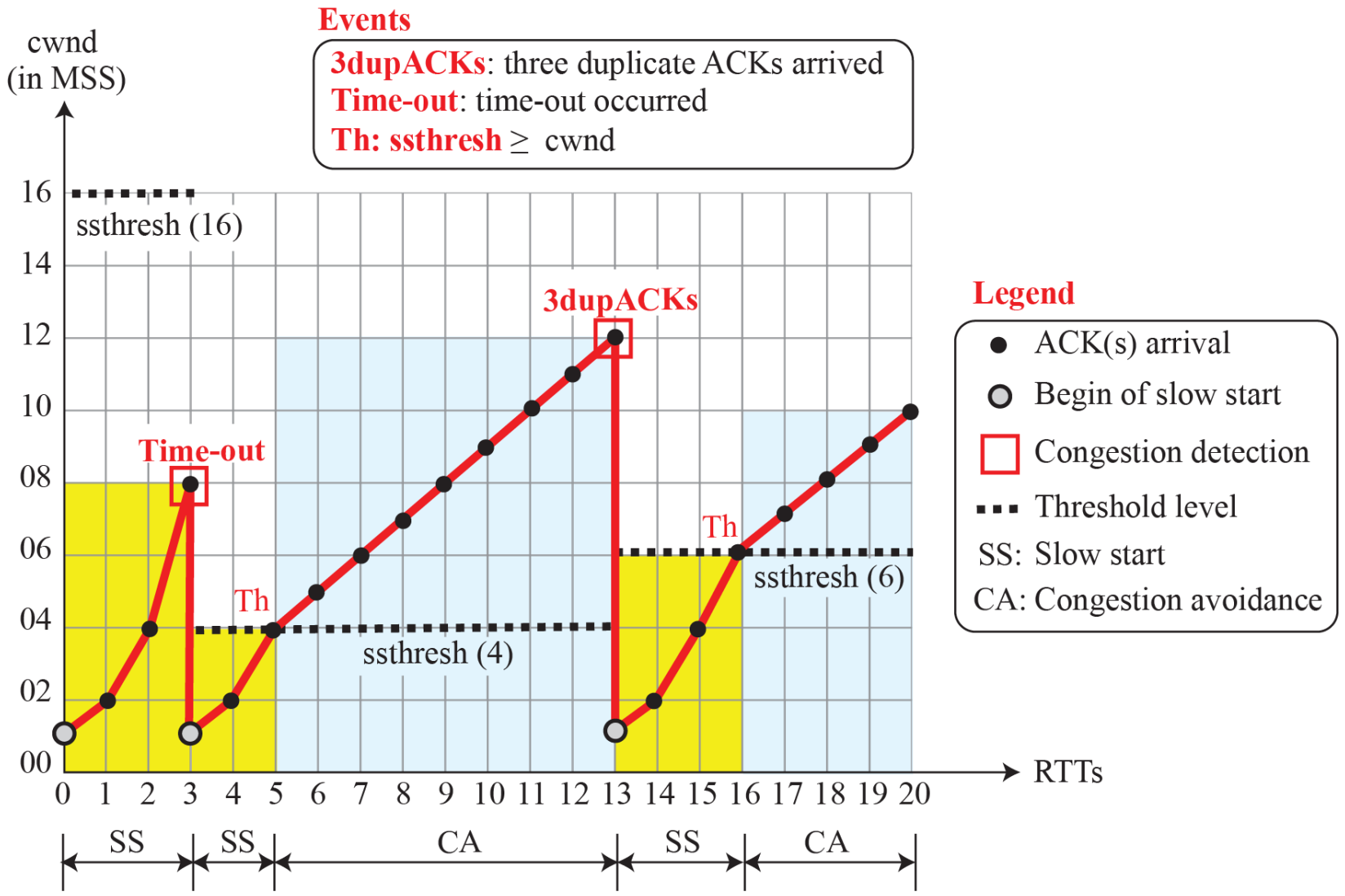
Slow start, exponential increase



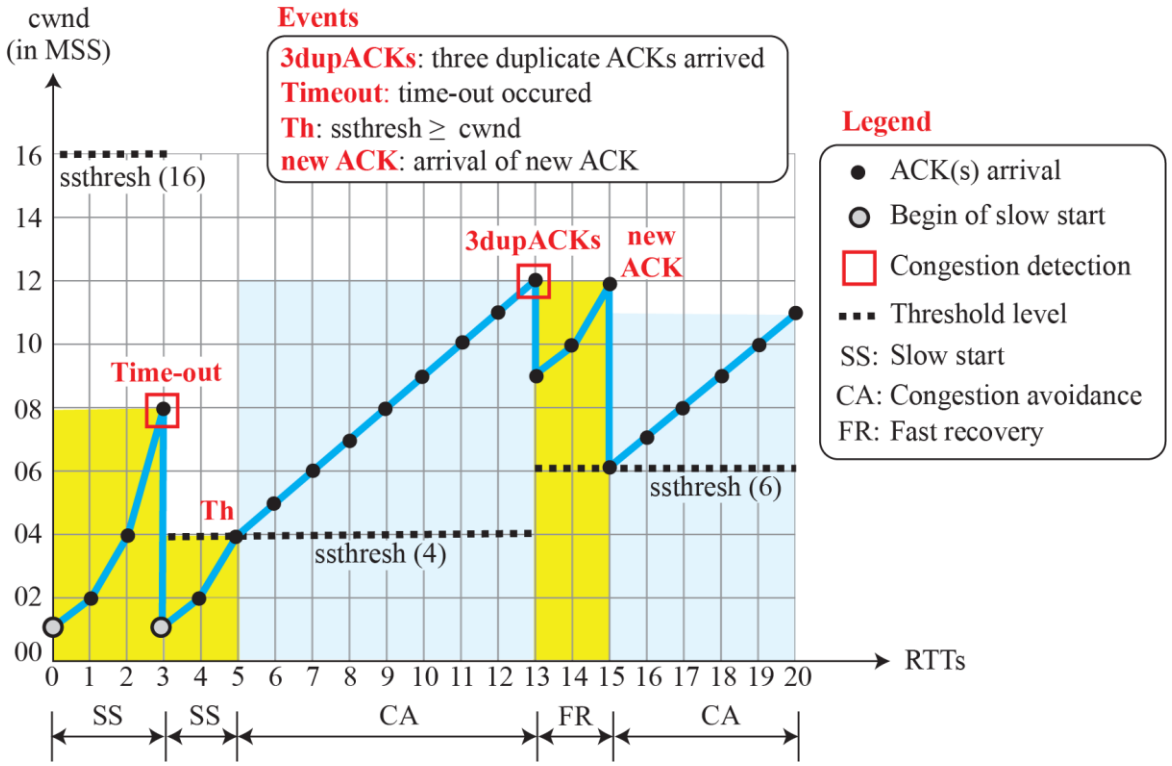
Congestion avoidance, additive increase



Example of Taho TCP



Example of a Reno TCP



Additive increase, multiplicative decrease (AIMD)