Transmission Control Protocol

Introduction

- TCP is a connection-oriented protocol; it creates a virtual connection between two TCPs to send data.
- TCP uses flow and error control mechanisms at the transport level.

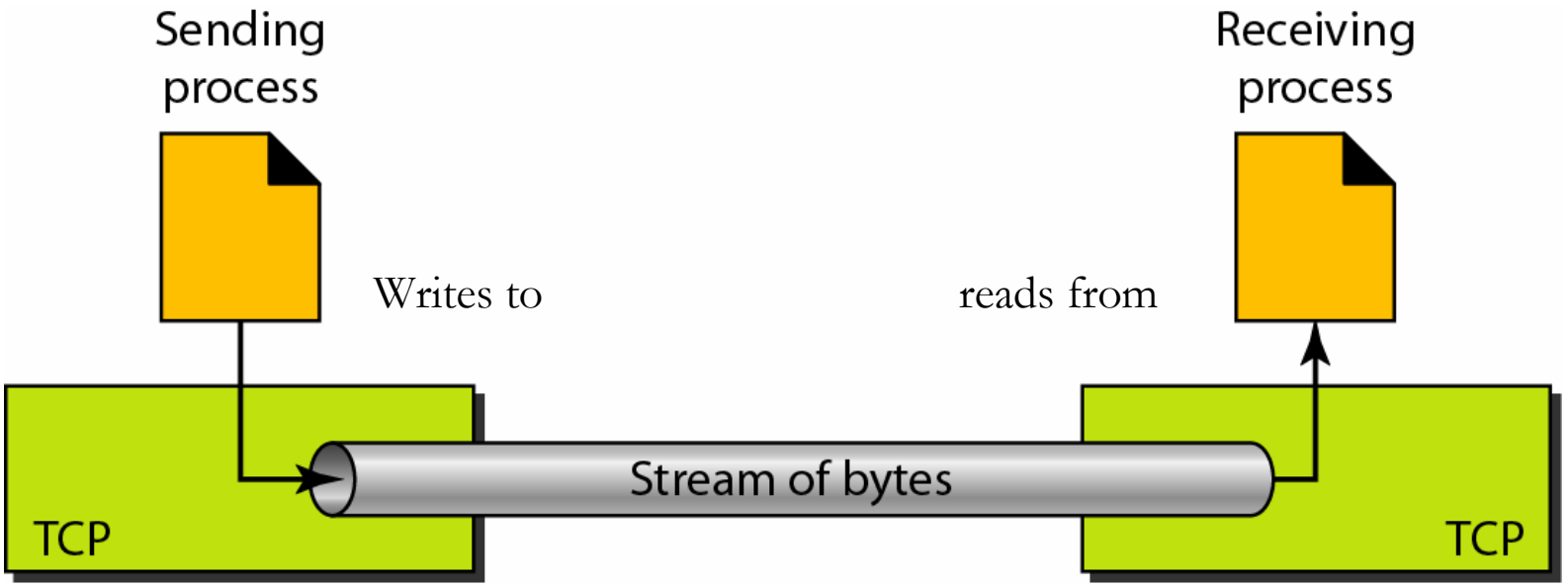
TCP Services

- Process to Process Communication
- Stream Delivery Service
- Sending and Receiving Buffers
- Segments
- Full Duplex communication
- Connection oriented service
- Reliable Service

Process to Process Communication

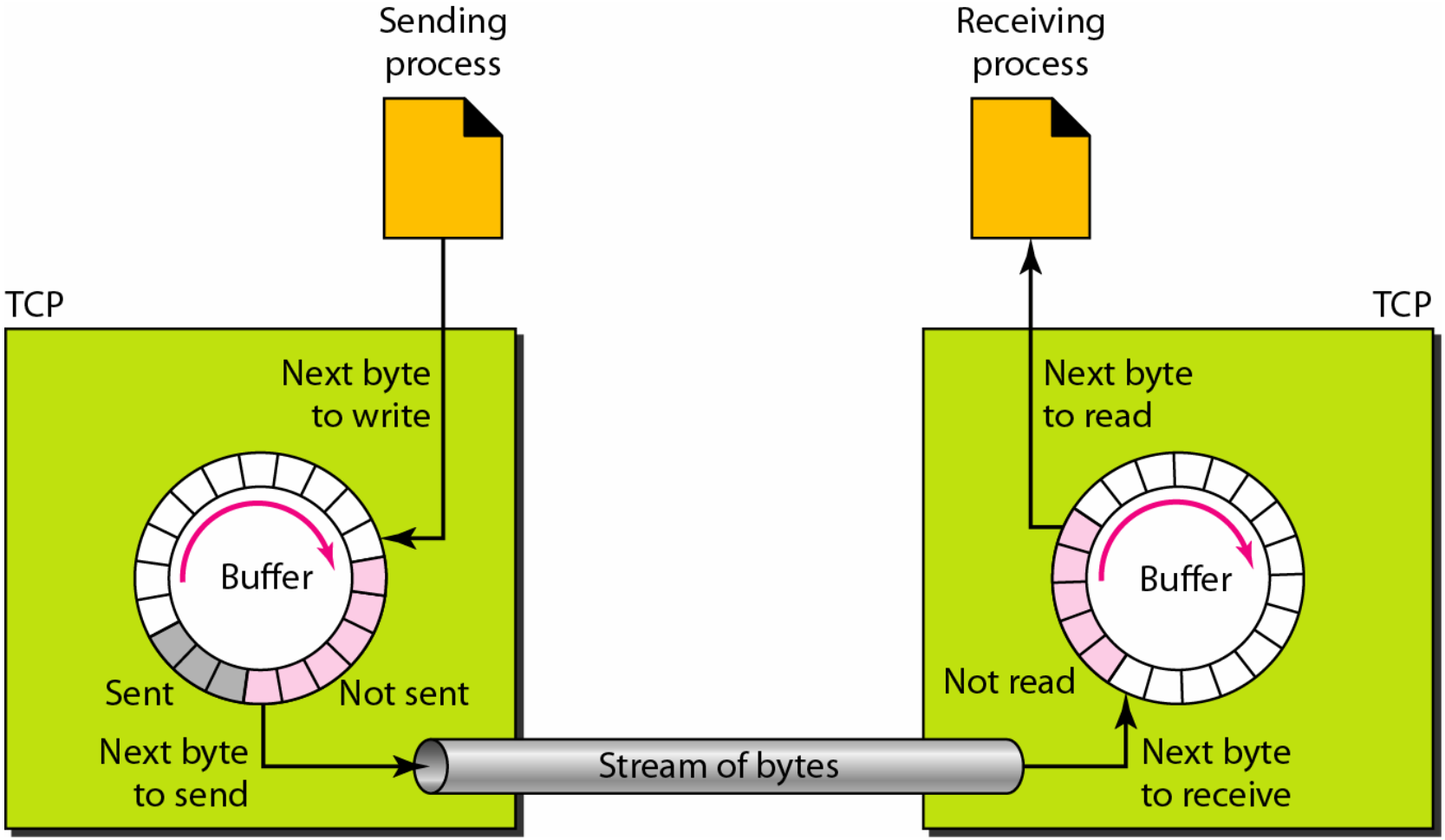
Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20	FTP, Data	File Transfer Protocol (data connection)
21	FTP, Control	File Transfer Protocol (control connection)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote Procedure Call

Stream Delivery Service

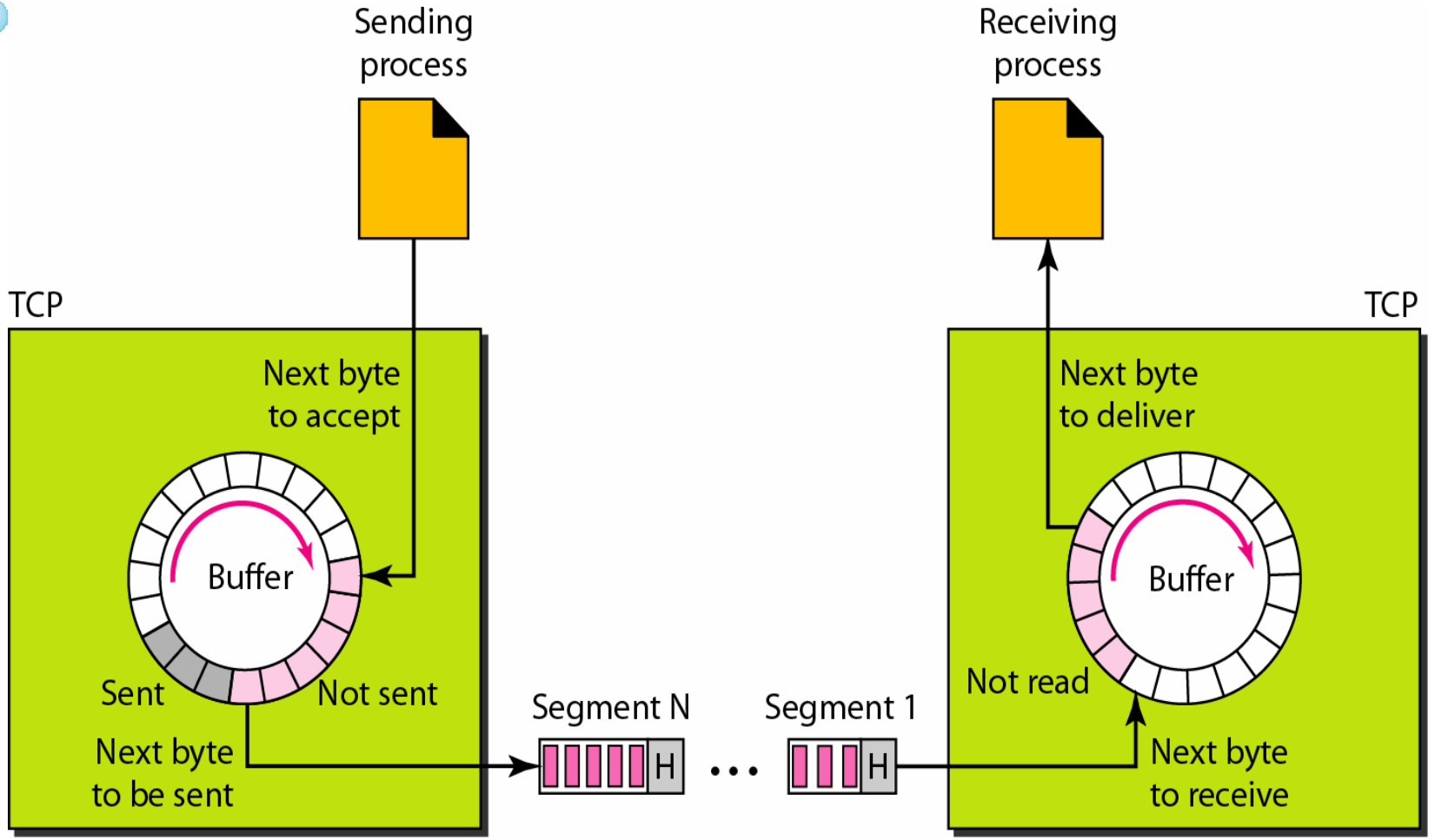


Sending and Receiving Buffers

- Reading and writing are not at the same speed.



Segments

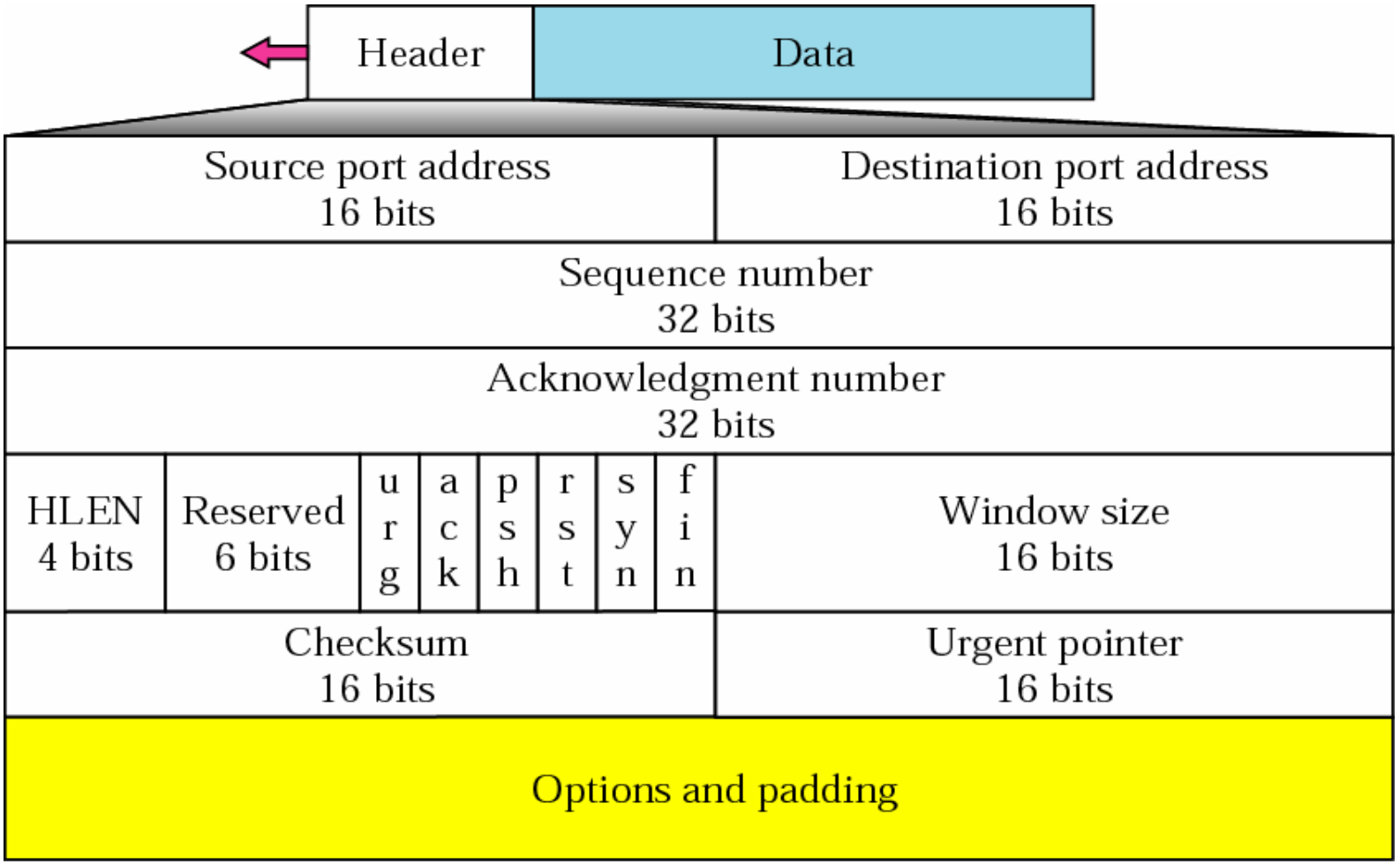

Sending process
Receiving process
TCP
TCP
Next byte to accept
Next byte to deliver
Buffer
Buffer
Sent
Not sent
Not read
Segment N
Segment 1
H
H
Next byte to be sent
Next byte to receive

Full Duplex communication

- Data flows in both direction at the same time.
- ie segments can move in both directions at the same time

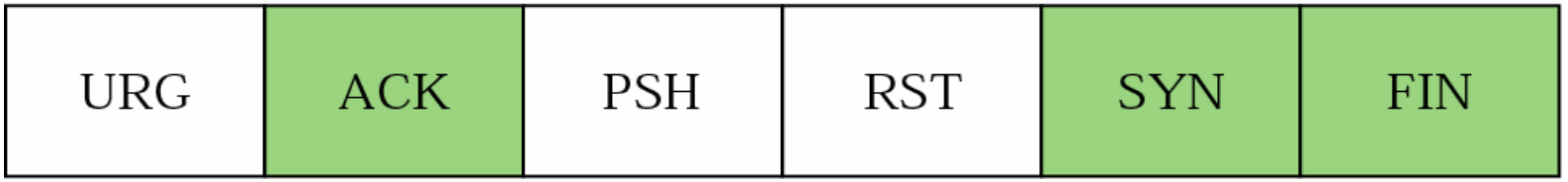
Segment

- 20 or 60 byte header



Control Fields

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



Numbering System

- Byte Number
- Sequence Number
- Acknowledgment Number

Byte Number

- Each byte should be numbered
- 0 –6 Random number – 1057
- Data contains 6000 bytes
- 1057 – 7056

Sequence Number

- Sequence number for each segment is the number of the first byte carried in that segment
- Imagine a TCP connection is transferring a file of 6000 bytes. The first byte is numbered 10010. What are the sequence numbers for each segment if data are sent in five segments with the first four segments carrying 1000 bytes and the last segment carrying 2000 bytes?

Sequence Number

- The following shows the sequence number for each segment:

Segment 1 ==> sequence number: 10,010 (range: 10,010 to 11,009)

Segment 2 ==> sequence number: 11,010 (range: 11,010 to 12,009)

Segment 3 ==> sequence number: 12,010 (range: 12,010 to 13,009)

Segment 4 ==> sequence number: 13,010 (range: 13,010 to 14,009)

Segment 5 ==> sequence number: 14,010 (range: 14,010 to 16,009)

Acknowledgment Number

- The value of the acknowledgment field in a segment defines the number of the next byte a party expects to receive.
- The acknowledgment number is cumulative.

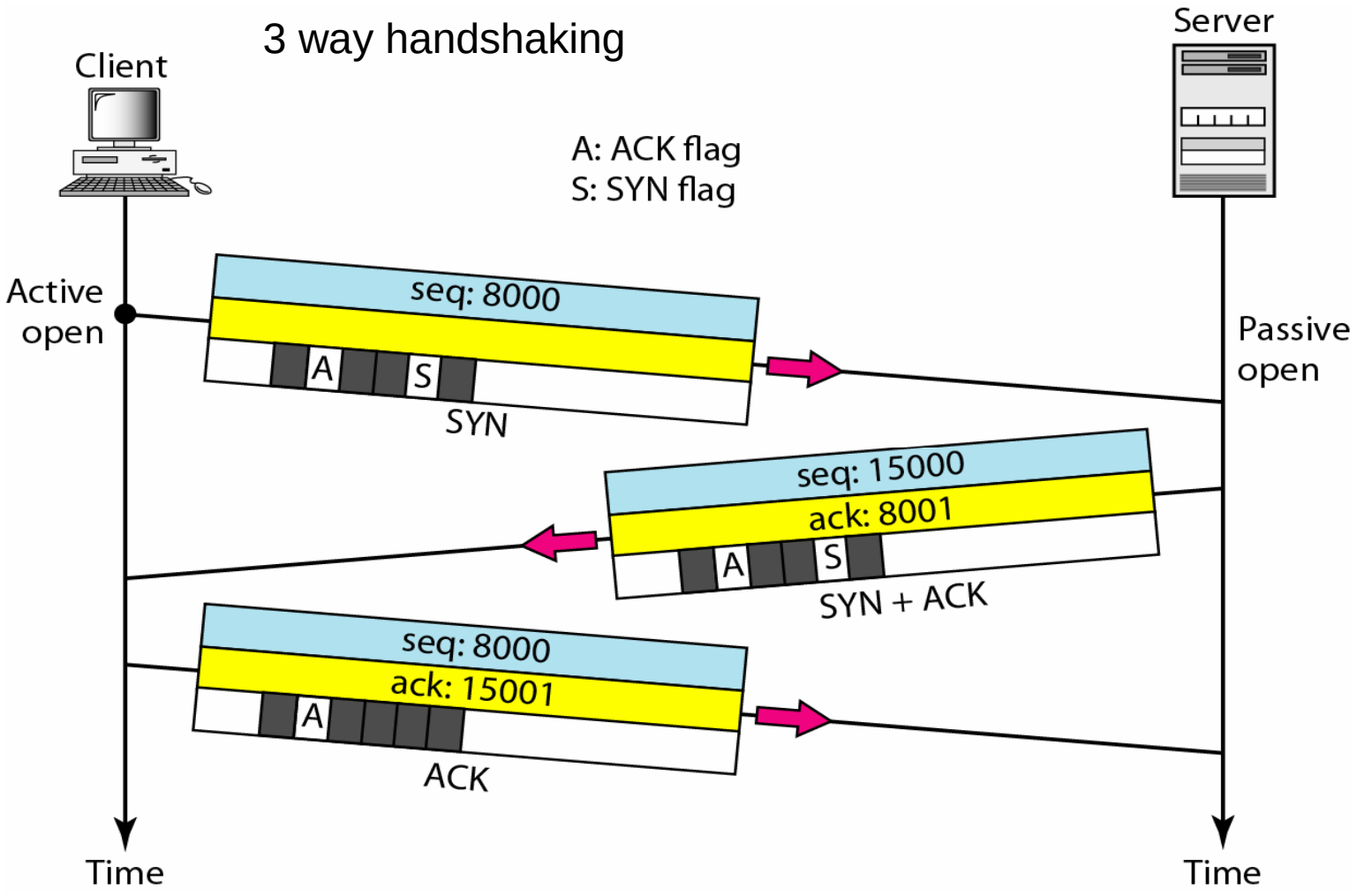
TCP Features

- Numbering system
- Flow control
- Error Control
- Congestion Control

TCP Connection

- 3 phases
  - Connection Establishment
  - Data transfer
  - Connection Termination

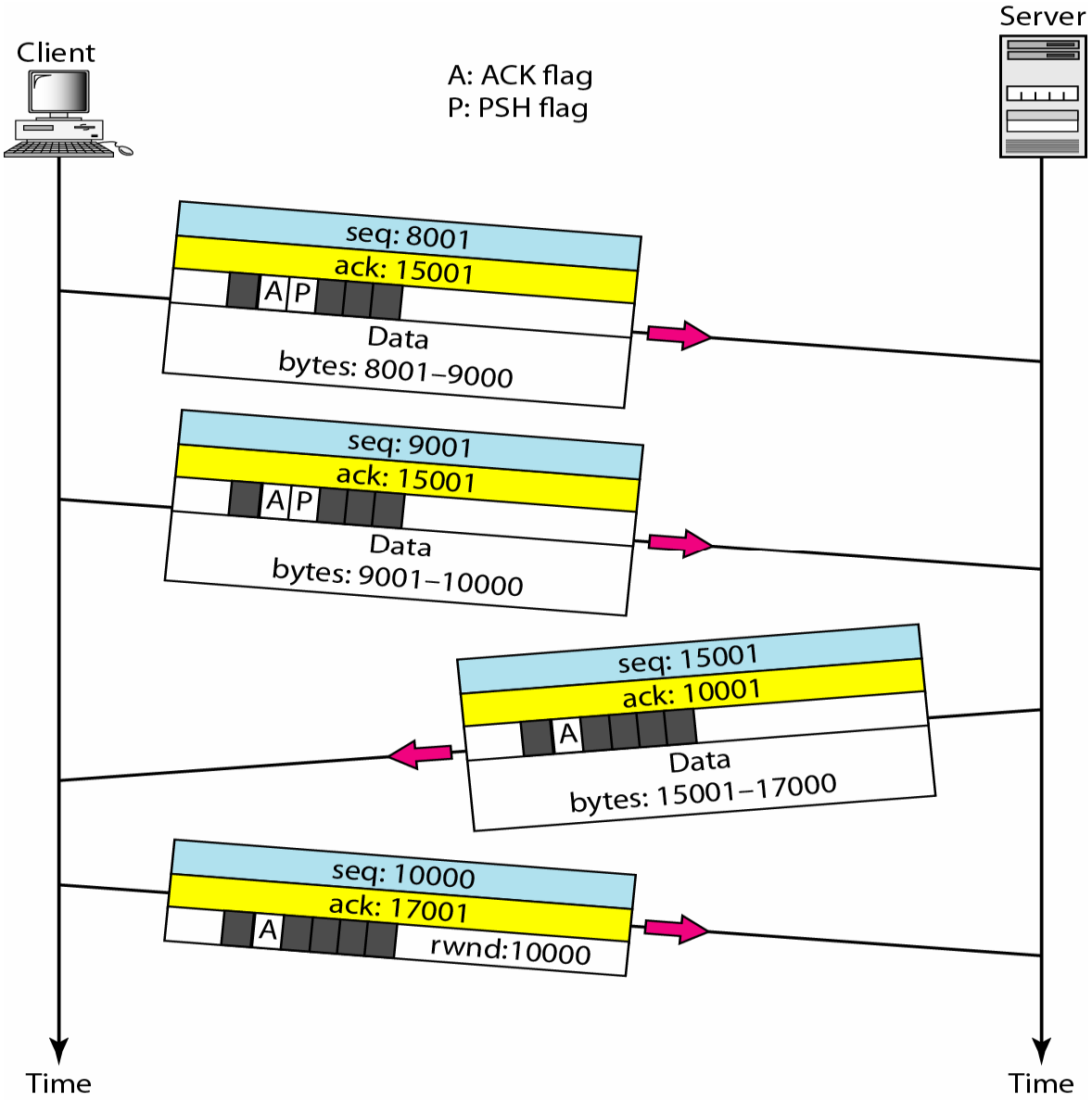
Connection Establishment



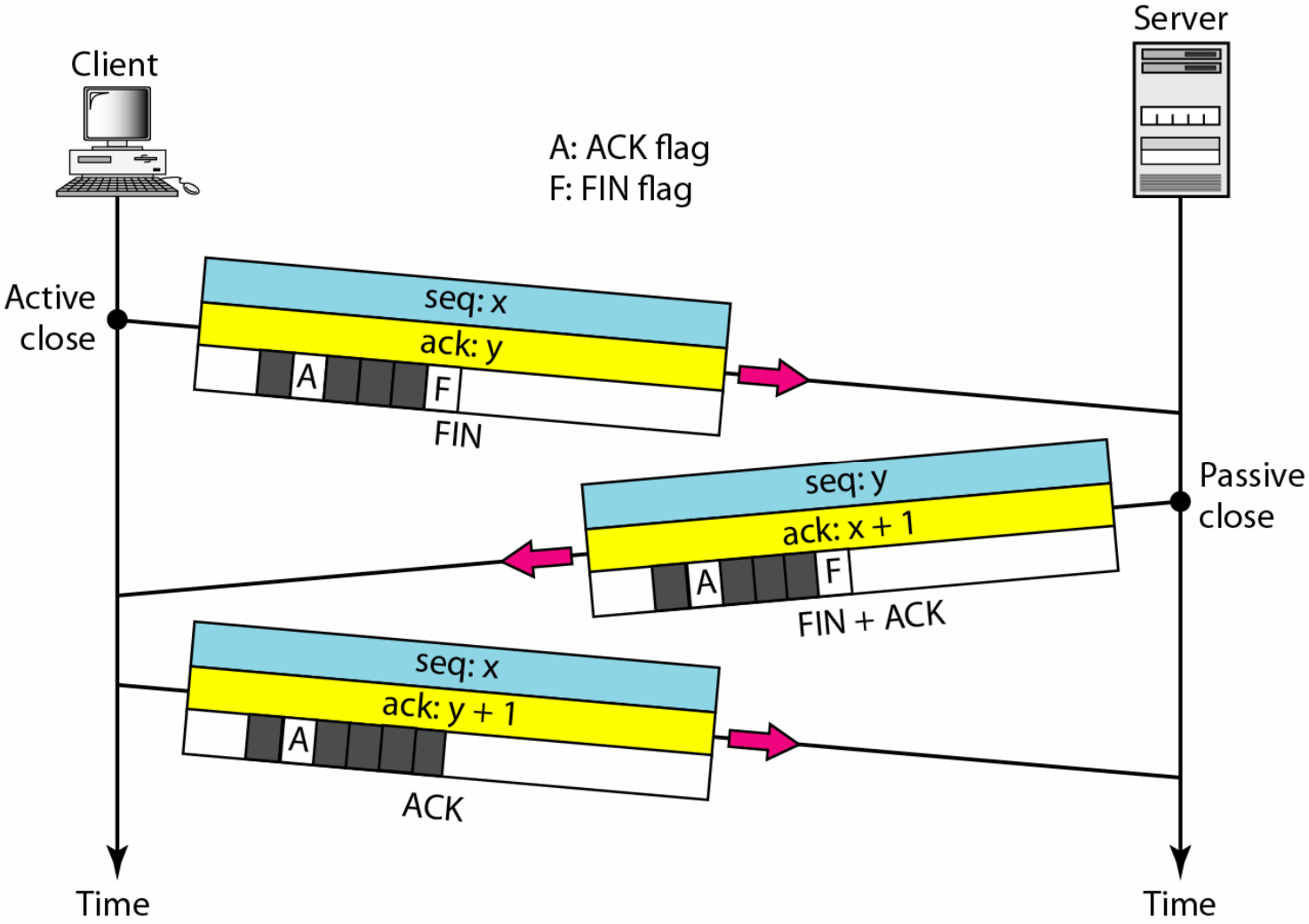
Connection Establishment

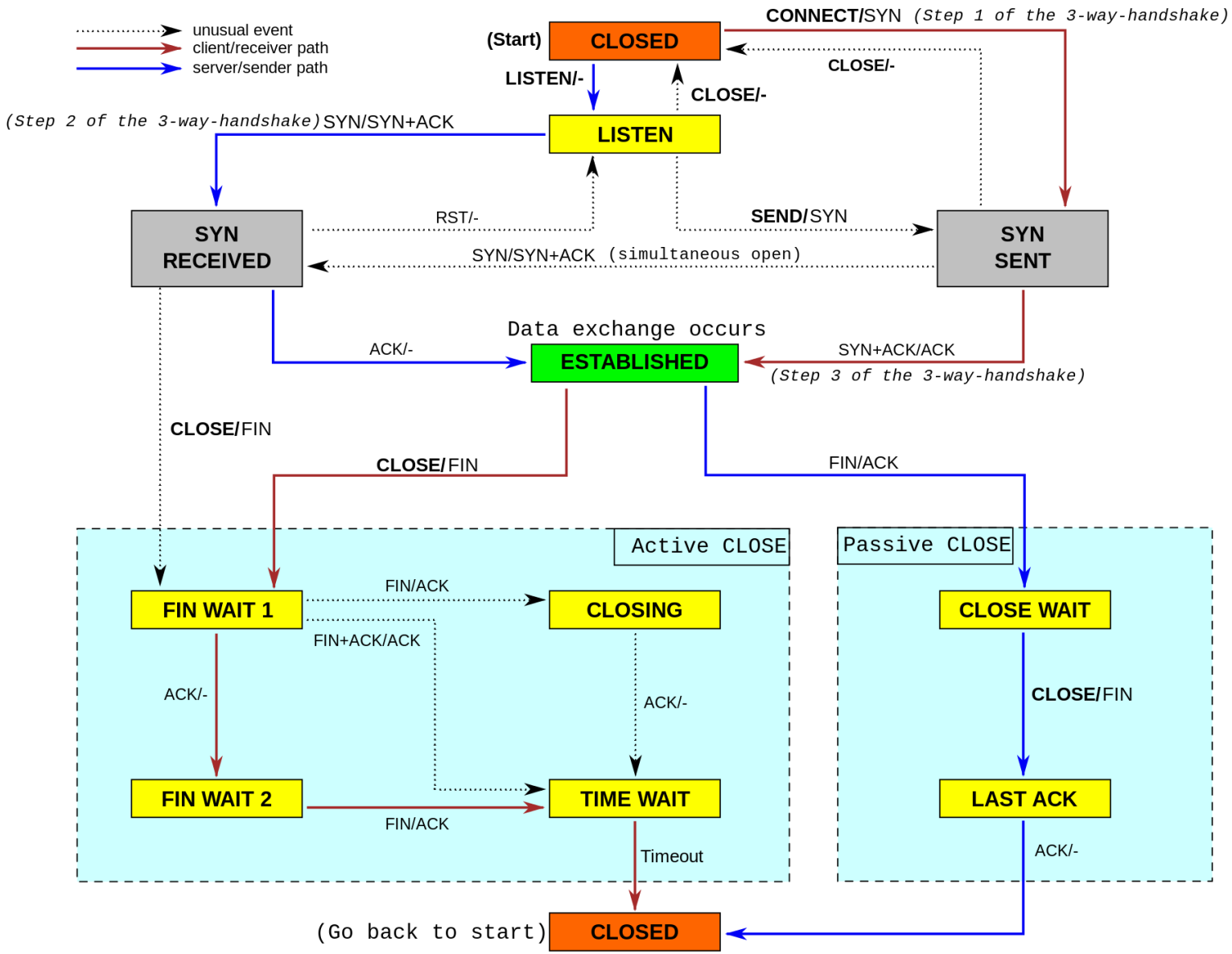
- A SYN segment cannot carry data, but it consumes one sequence number.
- A SYN + ACK segment cannot carry data, but does consume one sequence number.
- An ACK segment, if carrying no data, consumes no sequence number

Data Transfer



Connection Termination




unusual event
client/receiver path
server/sender path
(Start)
CLOSED
CONNECT/SYN (Step 1 of the 3-way-handshake)
CLOSE/-
LISTEN/-
CLOSE/-
LISTEN
(Step 2 of the 3-way-handshake) SYN/SYN+ACK
SYN RECEIVED
RST/-
SEND/SYN
SYN SENT
SYN/SYN+ACK (simultaneous open)
Data exchange occurs
ACK/-
ESTABLISHED
SYN+ACK/ACK
(Step 3 of the 3-way-handshake)
CLOSE/FIN
CLOSE/FIN
FIN/ACK
Active CLOSE
Passive CLOSE
FIN WAIT 1
FIN/ACK
CLOSING
CLOSE WAIT
FIN+ACK/ACK
ACK/-
ACK/-
CLOSE/FIN
FIN WAIT 2
TIME WAIT
LAST ACK
FIN/ACK
Timeout
ACK/-
(Go back to start)
CLOSED