Ch. 5 TCP/IP Protocol Suite
Part 2
Process-to-Process Delivery: UDP, TCP

Most slides are from the instructor resources of the following books:

1. Ch 23, Data communications and networking, 4th edition, Forouzan, McGrawHill
2. Computer networks, 4th edition, Tanenbaum, Prentice Hall
3. Textbook

PROCESS-TO-PROCESS DELIVERY

*The **transport layer** is responsible for process-to-process delivery—the delivery of a packet, part of a message, from one process to another. Two processes communicate in a client/server relationship, as we will see later.*

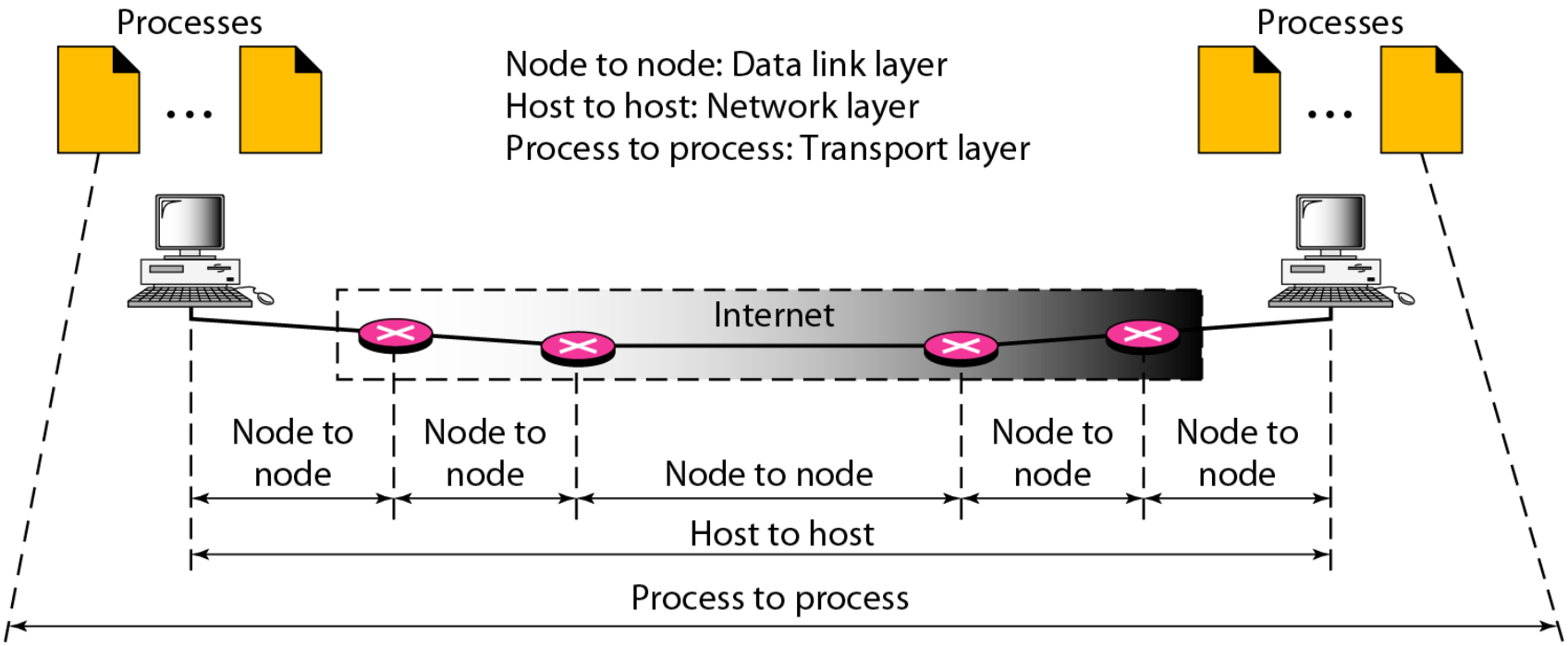
Topics discussed in this section:

Client/Server Paradigm
Multiplexing and Demultiplexing
Connectionless Versus Connection-Oriented Service
Reliable Versus Unreliable
Three Protocols

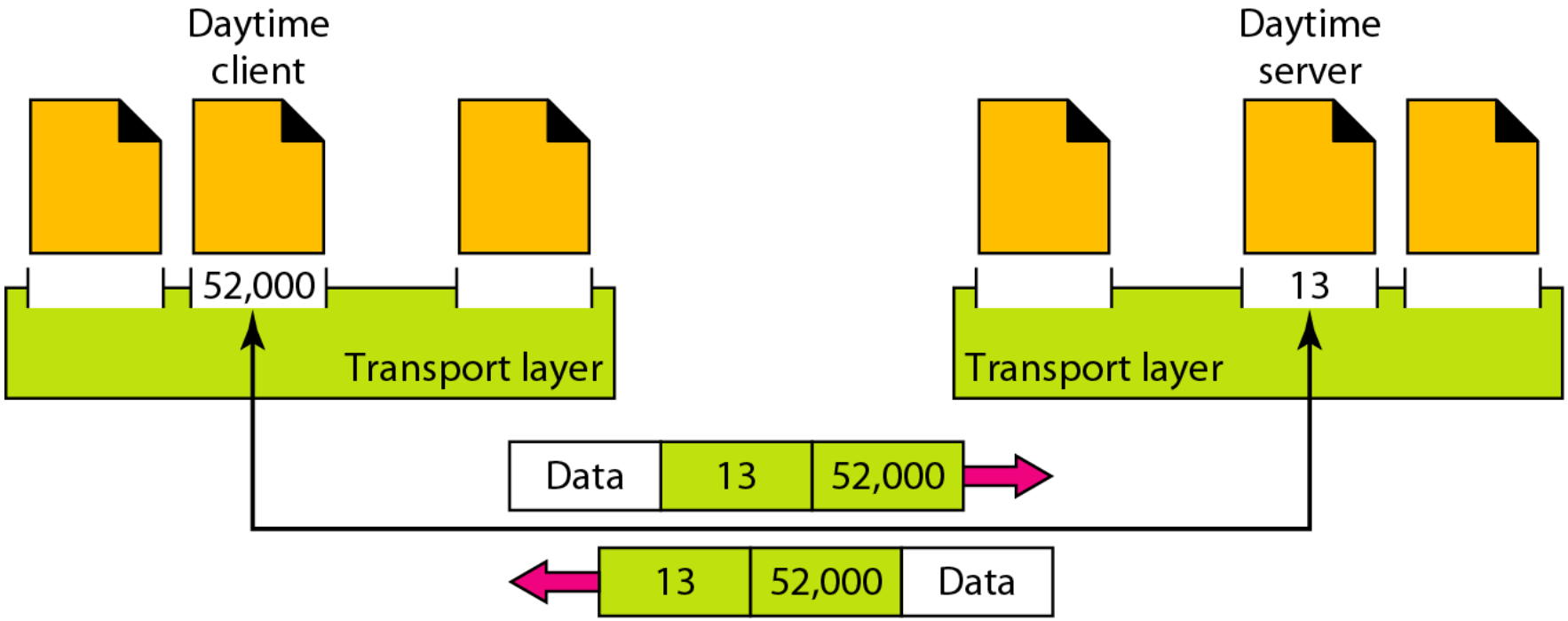
Note

The **transport layer** is responsible for **process-to-process delivery**.

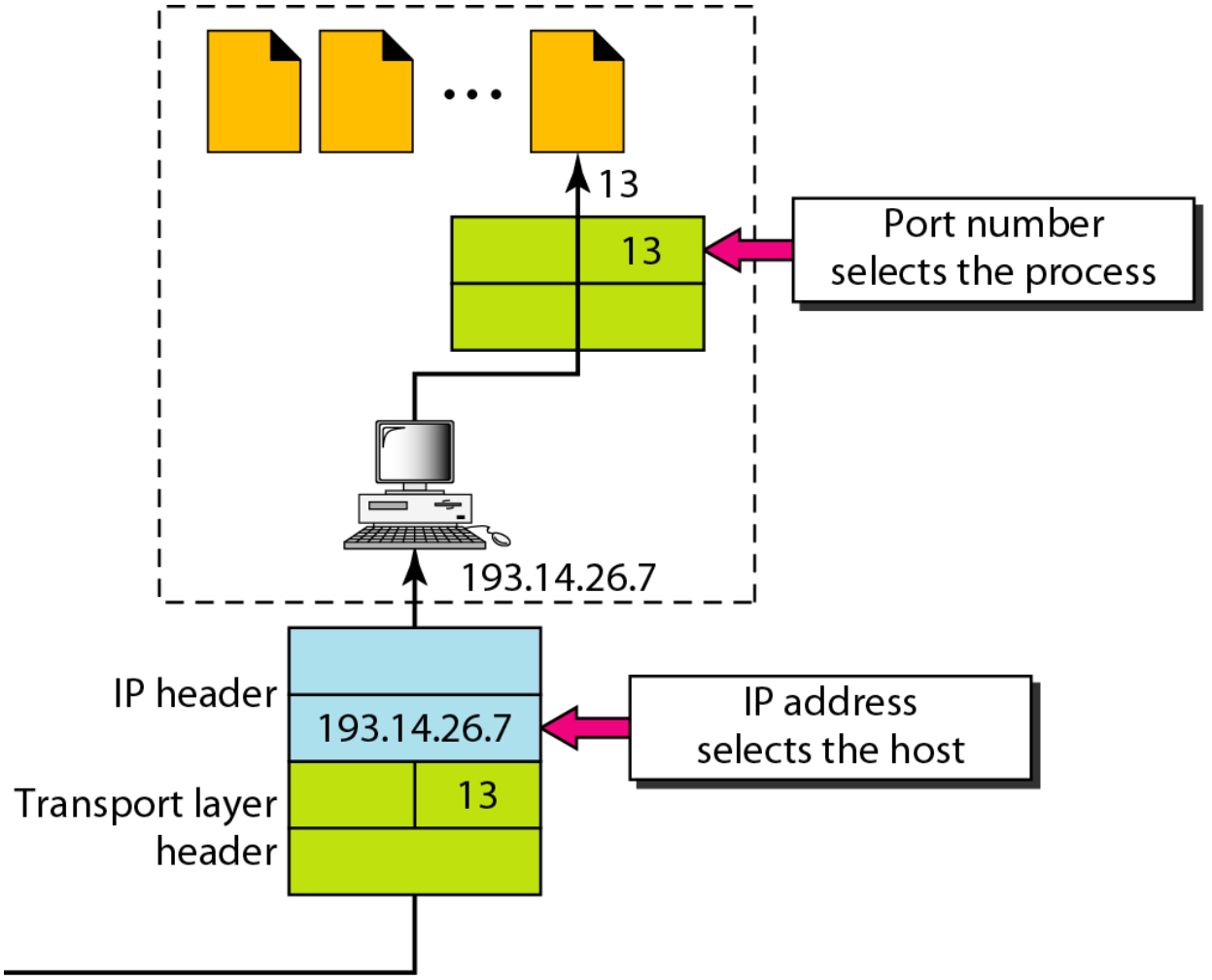
Types of data deliveries



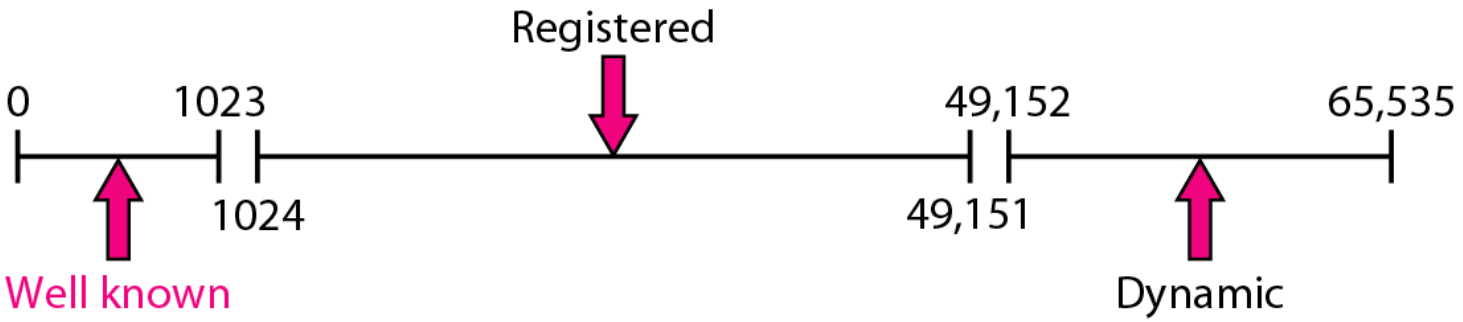
Port numbers (=transport layer address)



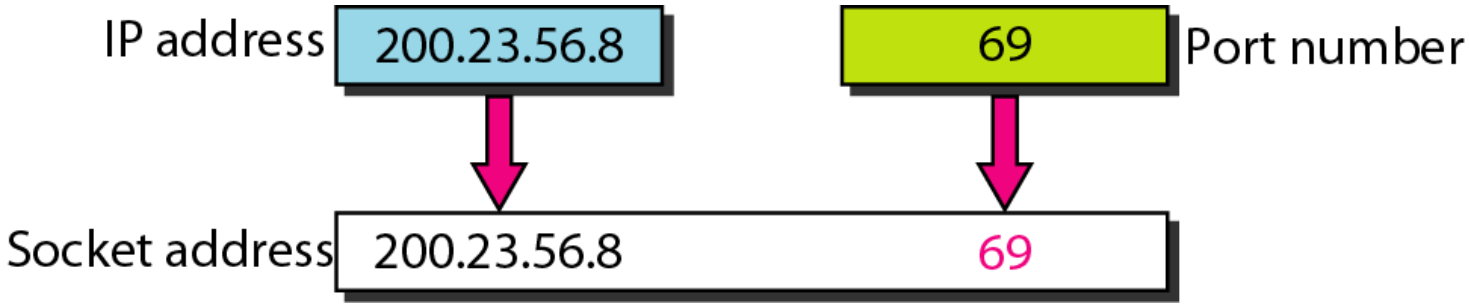
IP addresses versus port numbers



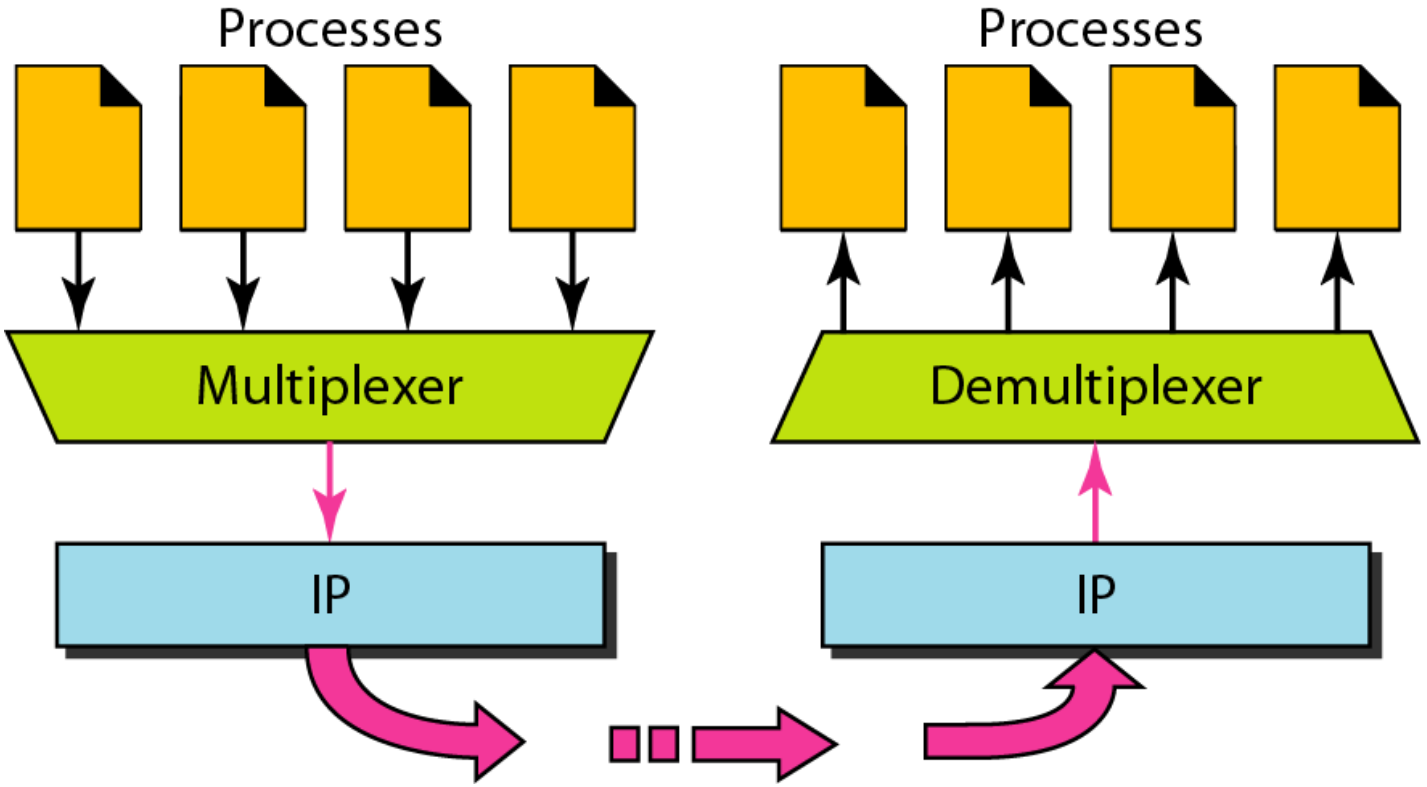
IANA (=internet assigned number authority) ranges



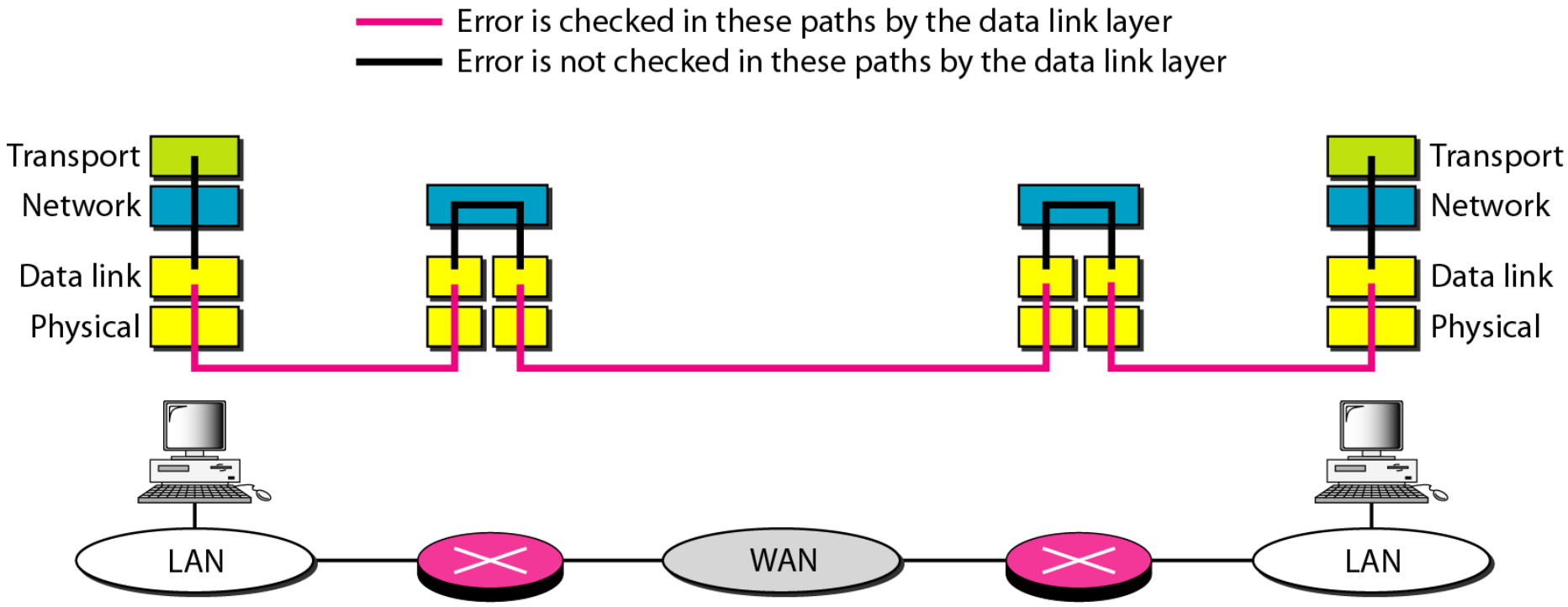
Socket address



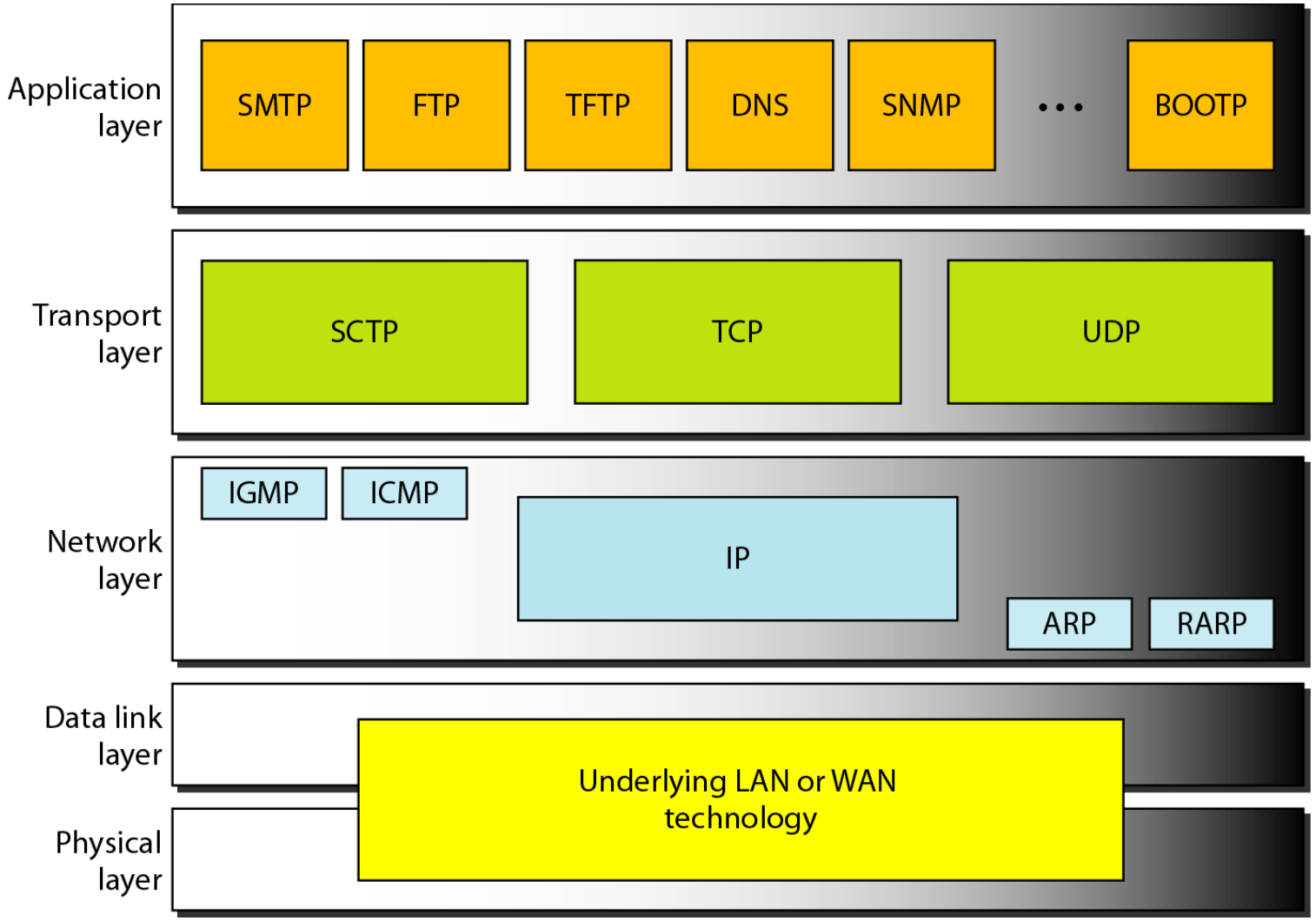
Multiplexing and demultiplexing



Error control



Position of UDP, TCP, and SCTP in TCP/IP suite



USER DATAGRAM PROTOCOL (UDP)

*The User Datagram Protocol (UDP) is called a **connectionless, unreliable transport protocol**. It does not add anything to the services of IP except to provide process-to-process communication instead of host-to-host communication.*

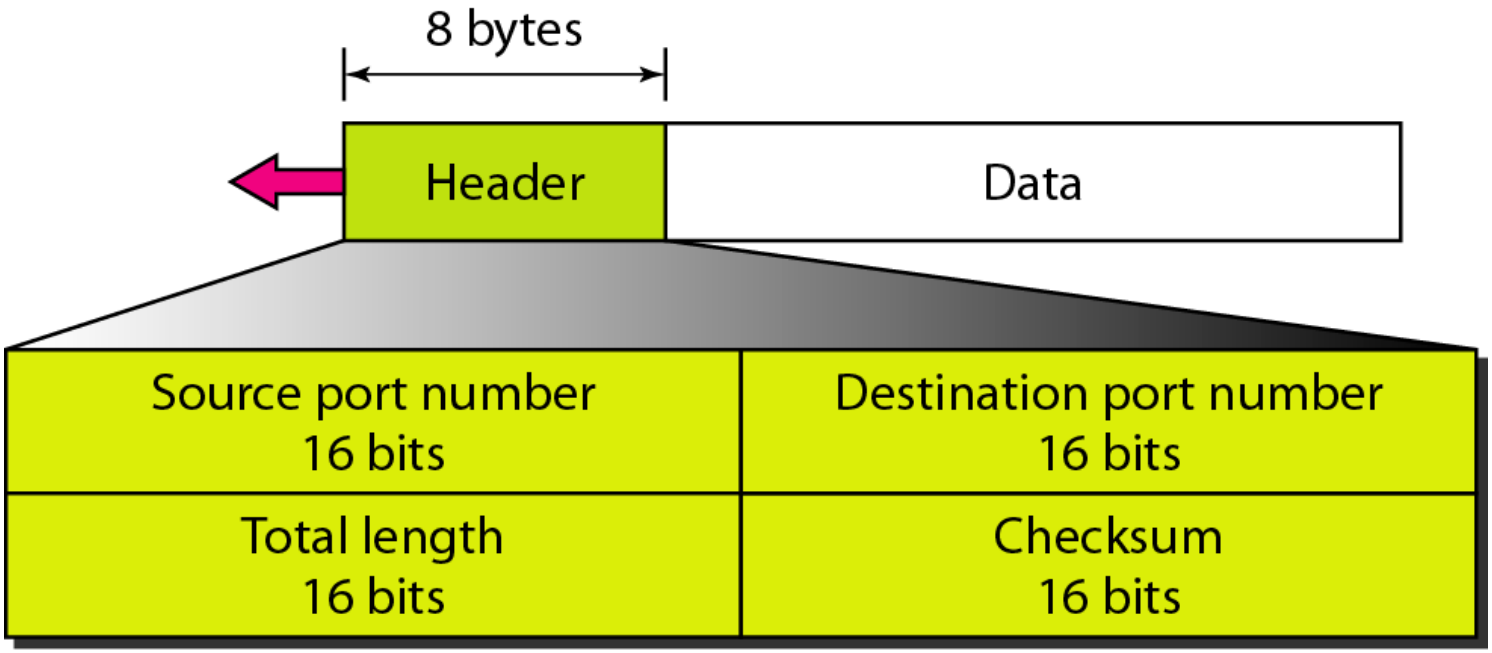
Topics discussed in this section:

Well-Known Ports for UDP
User Datagram
Checksum
UDP Operation
Use of UDP

Well-known ports used with UDP

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
53	Nameserver	Domain Name Service
67	BOOTPs	Server port to download bootstrap information
68	BOOTPc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

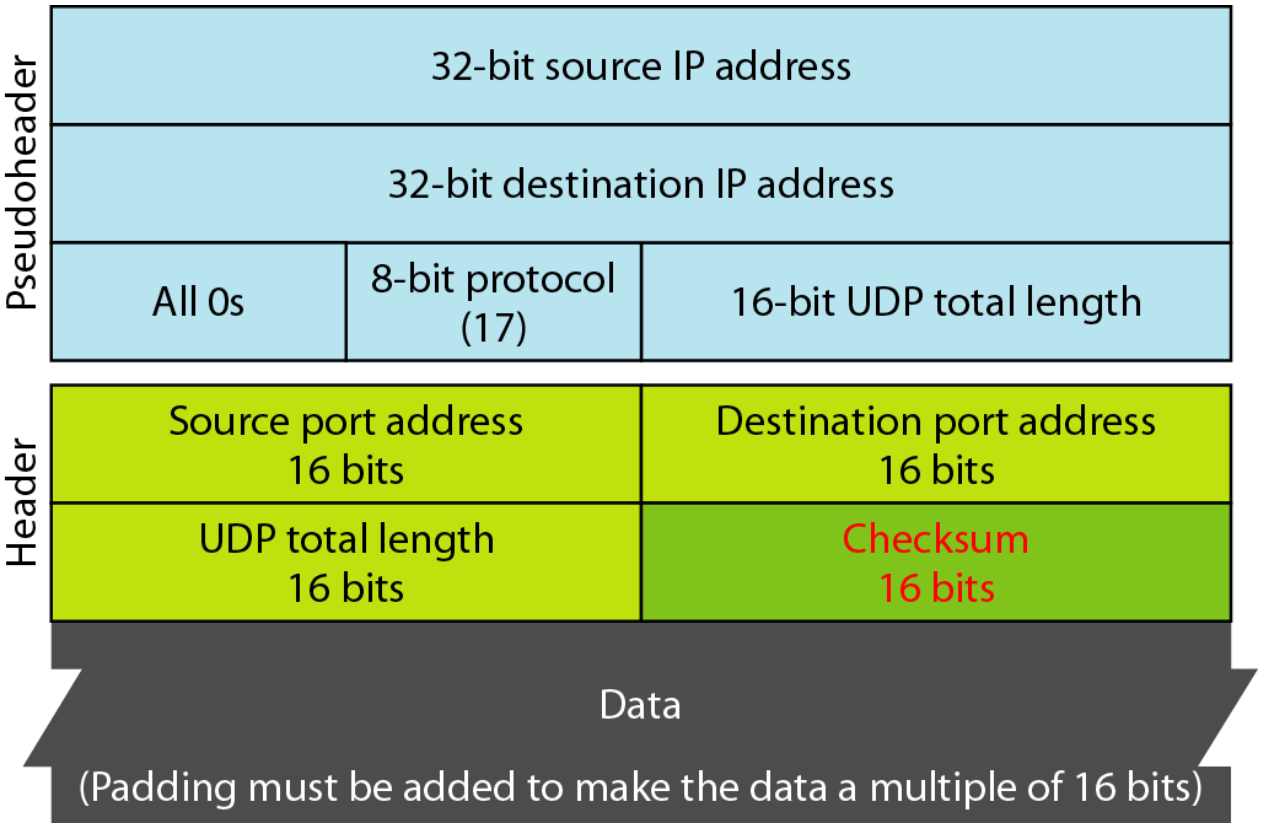
User datagram format



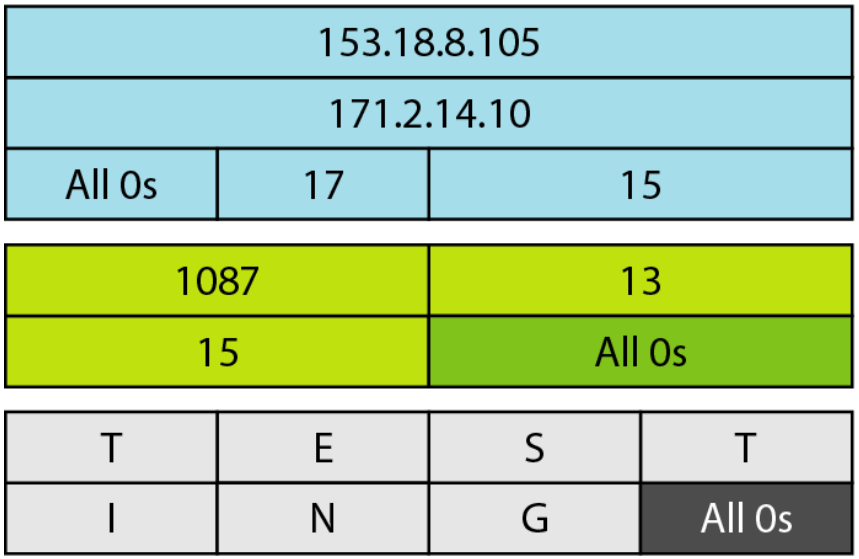
Note

UDP length
= IP length – IP header's length

Pseudoheader for checksum calculation



Checksum calculation of a simple UDP user datagram



10011001 00010010 → 153.18
00001000 01101001 → 8.105
10101011 00000010 → 171.2
00001110 00001010 → 14.10
00000000 00010001 → 0 and 17
00000000 00001111 → 15
00000100 00111111 → 1087
00000000 00001101 → 13
00000000 00001111 → 15
00000000 00000000 → 0 (checksum)
01010100 01000101 → T and E
01010011 01010100 → S and T
01001001 01001110 → I and N
01000111 00000000 → G and 0 (padding)

10010110 11101011 → Sum
01101001 00010100 → Checksum

UDP operation

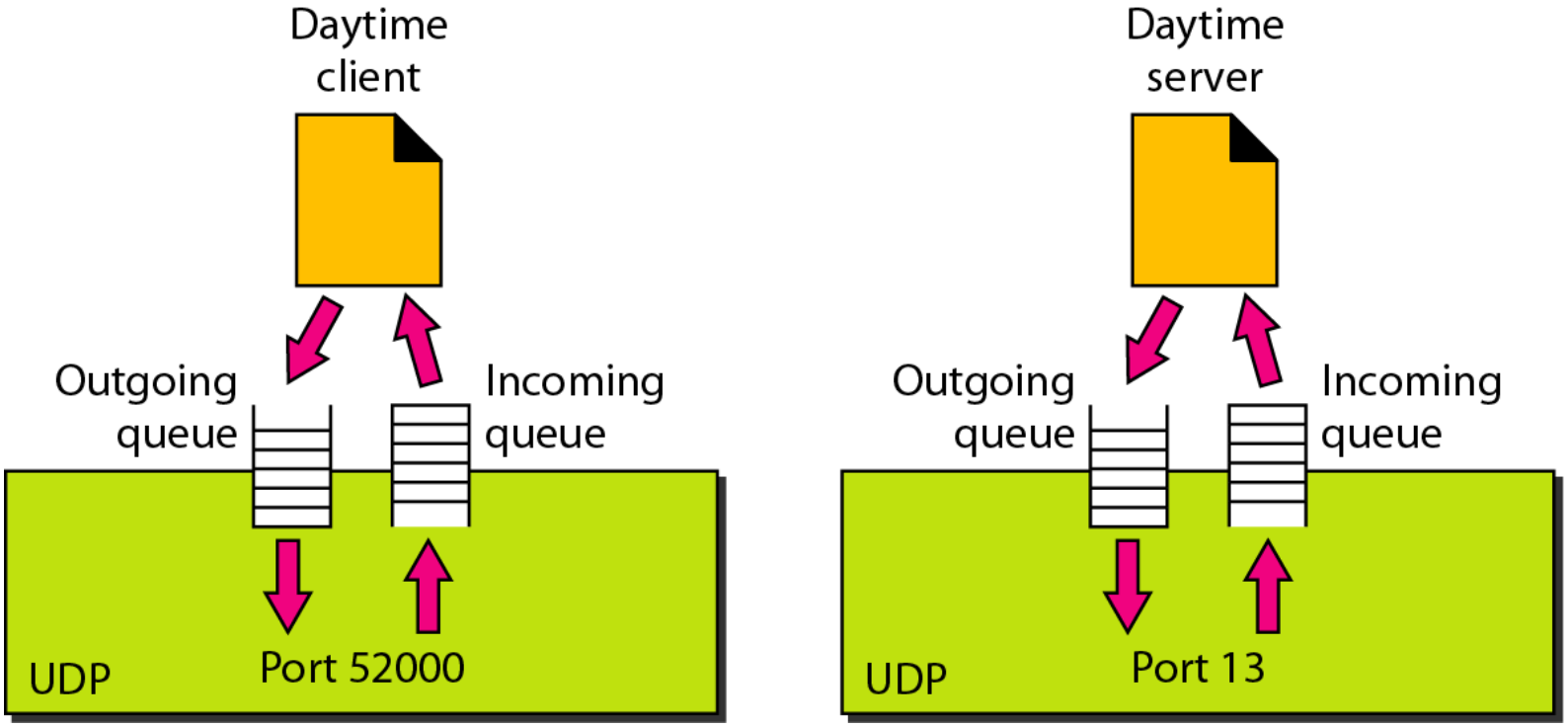
- Connectionless services
- No flow/error control
- Encapsulation and decapsulation
- Queuing

Use of UDP

- Suitable for a process that requires simple request-response communication. Ex. DNS
- Suitable for a process with internal flow and error control mechanisms. Ex. Trivial File Transfer Protocol
- A suitable transport protocol for multicasting
- Used for management processes such as SNMP
- Used for some route updating protocols such as RIP

UDP-Lite

Queues in UDP



TCP

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

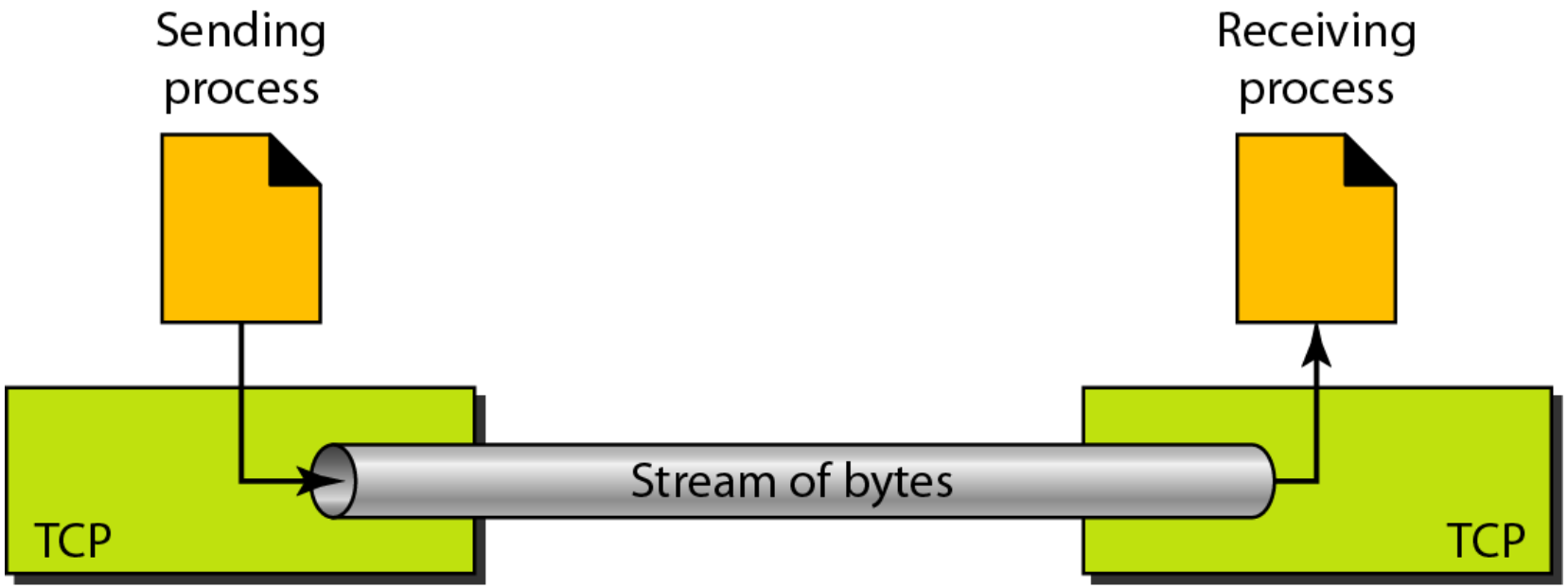
Topics discussed in this section:

TCP Services
TCP Features
Segment
A TCP Connection
Flow Control
Error Control

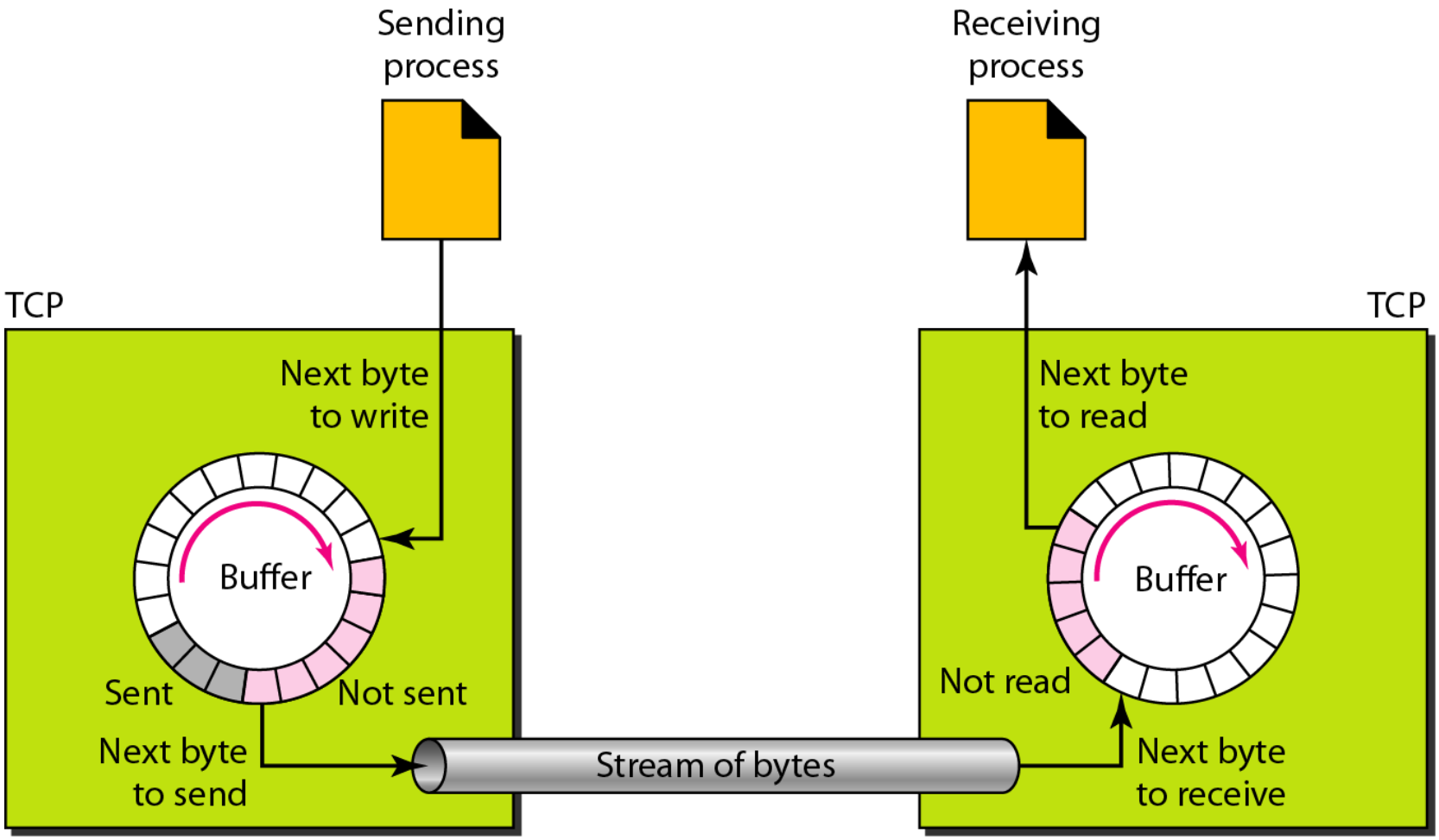
Well-known ports used by TCP

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

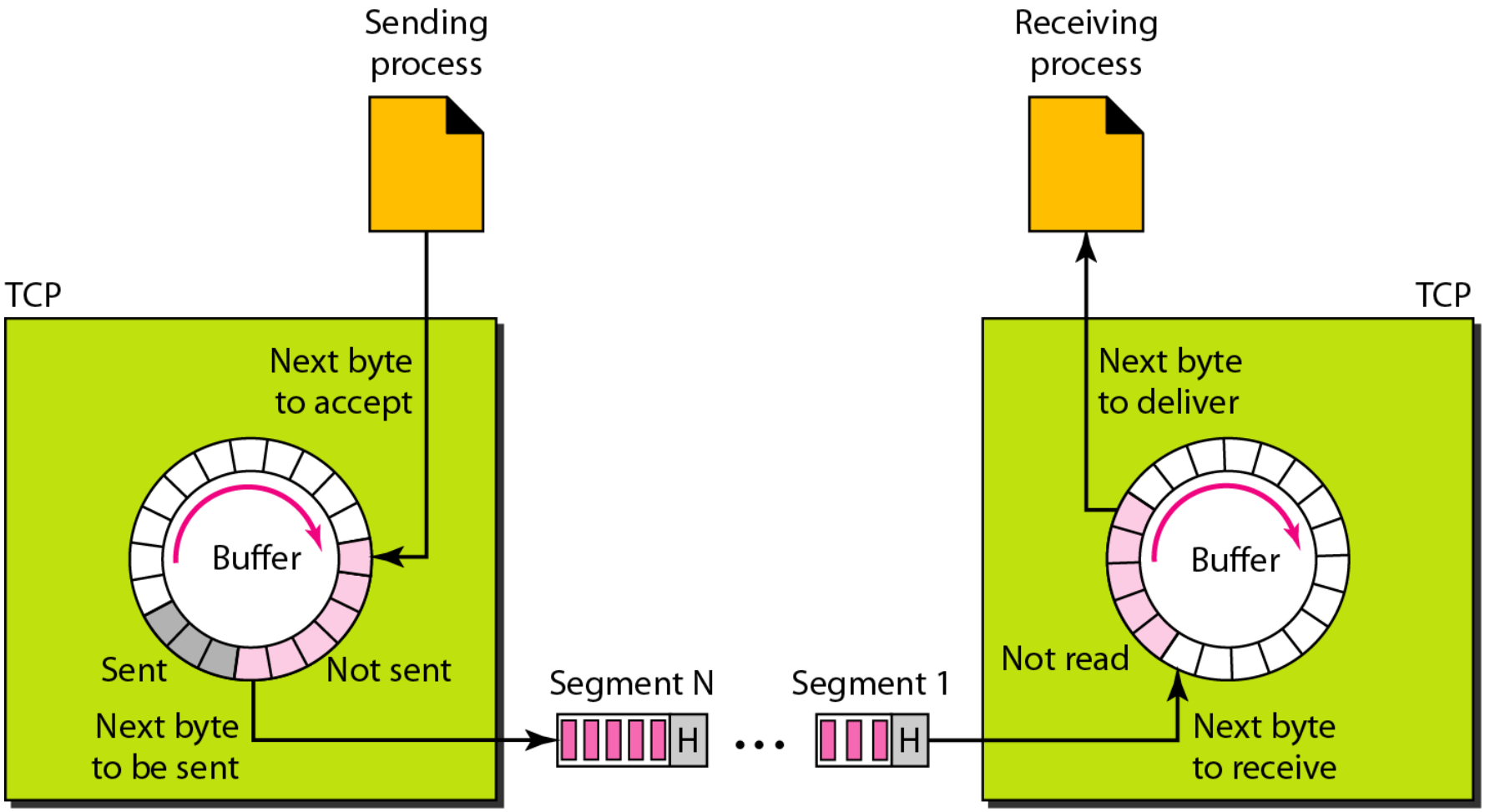
Stream delivery



Sending and receiving buffers



TCP segments



TCP adds a header to each segment (for control purpose) and delivers the segment to the _____ for transmission. The segments are encapsulated in _____ and transmitted.

Note

The bytes of data being transferred in each connection are numbered by TCP. The numbering starts with a randomly generated number.

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)**

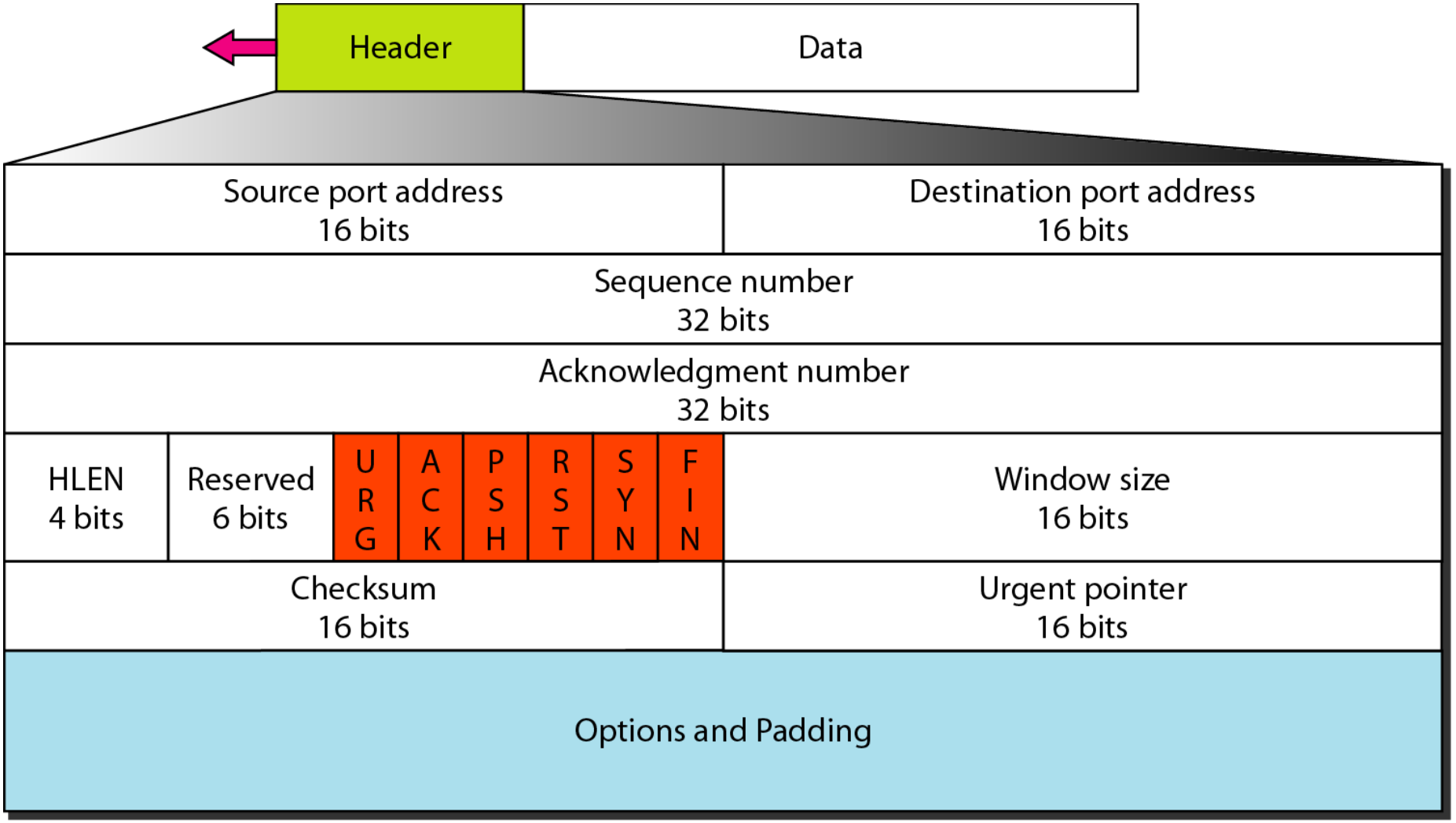
Note

The value in the **sequence number field** of a segment defines the number of the first data byte contained in that segment.

Note

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 segment format



Source port address (16-bit): port number of the application program in the host that is sending the segment

Destination port address (16-bit):

Sequence number (32-bit): defines the number assigned to the first byte of data contained in this segment

Acknowledgement number (32-bit): defines the byte number that the receiver of the segment is expecting to receive from the other party

Header length (4-bit): between 5(5*4=20) and 15 (15*4=60)

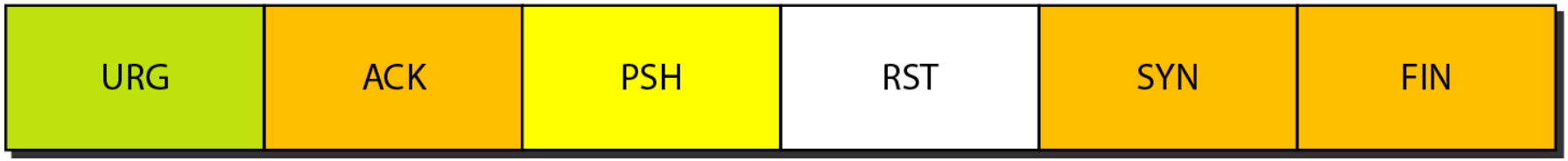
Reserved (6-bit)

Control (6-bit)

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



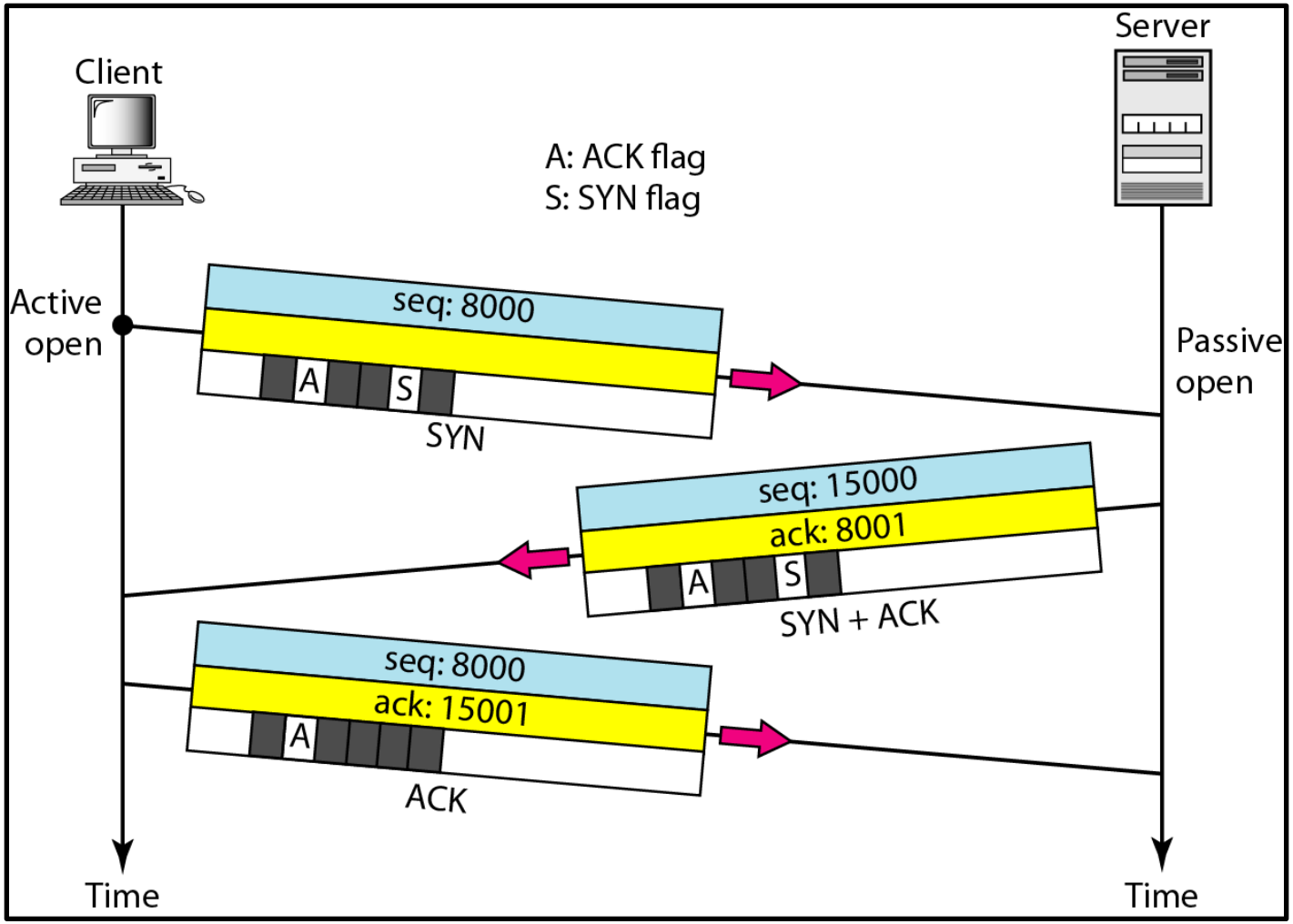
Window size (16-bit): defines the size of window, in bytes

Checksum (16-bit):

Urgent pointer (16-bit): valid only if urgent flag is set; used when the segment contains urgent data

Options (up to 40 bytes)

Connection establishment using three-way handshaking



Note

A SYN segment cannot carry data, but it consumes one sequence number.

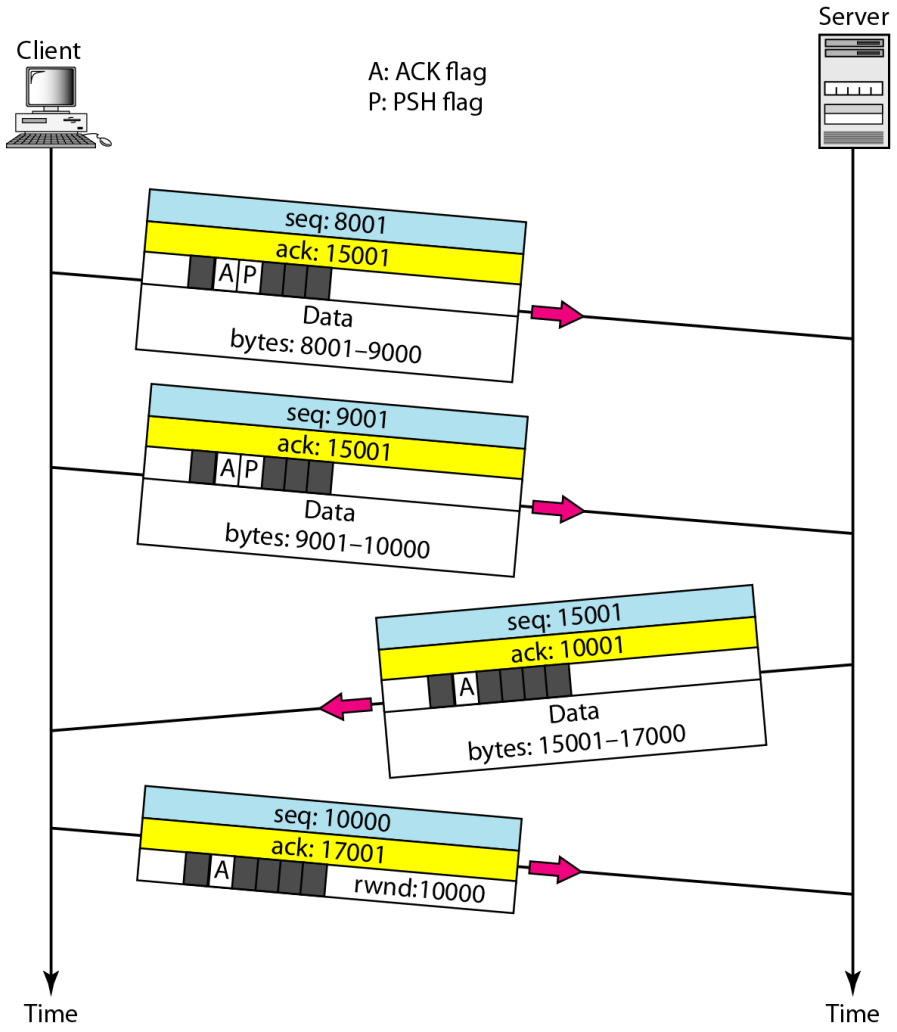
Note

A SYN + ACK segment cannot carry data, but does consume one sequence number.

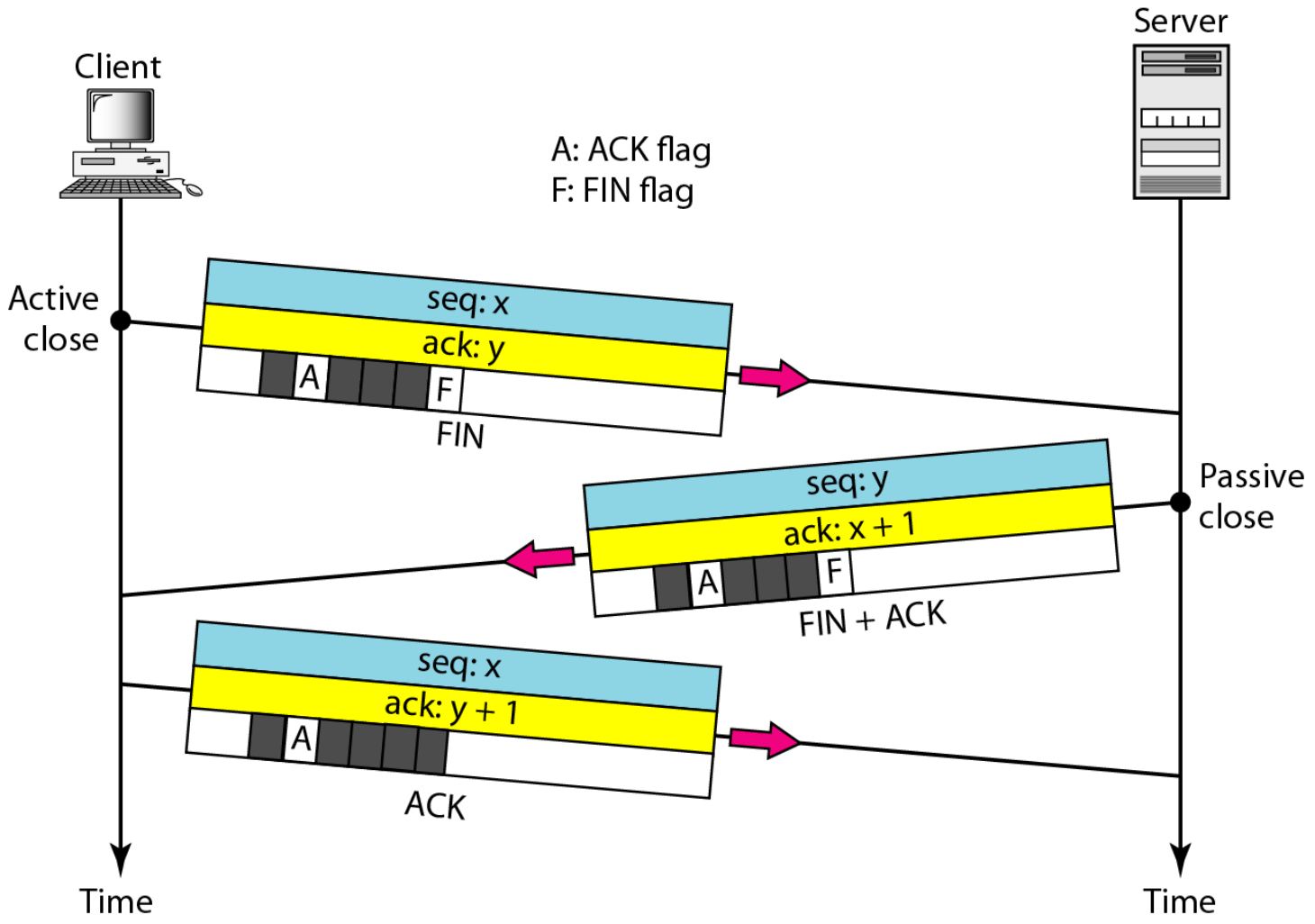
Note

An ACK segment, if carrying no data, consumes no sequence number.

Data transfer



Connection termination using 1)three-way handshaking



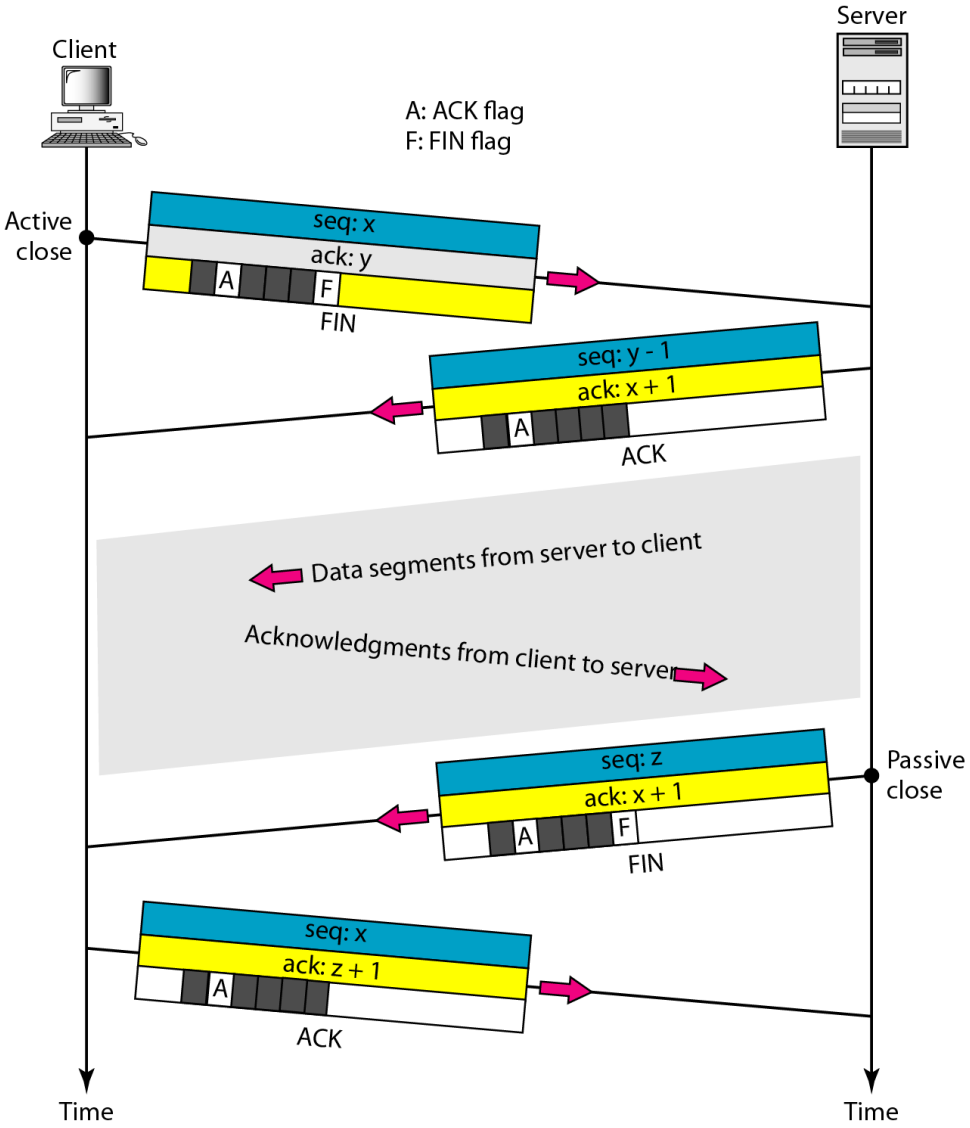
Note

The FIN segment consumes one sequence number if it does not carry data.

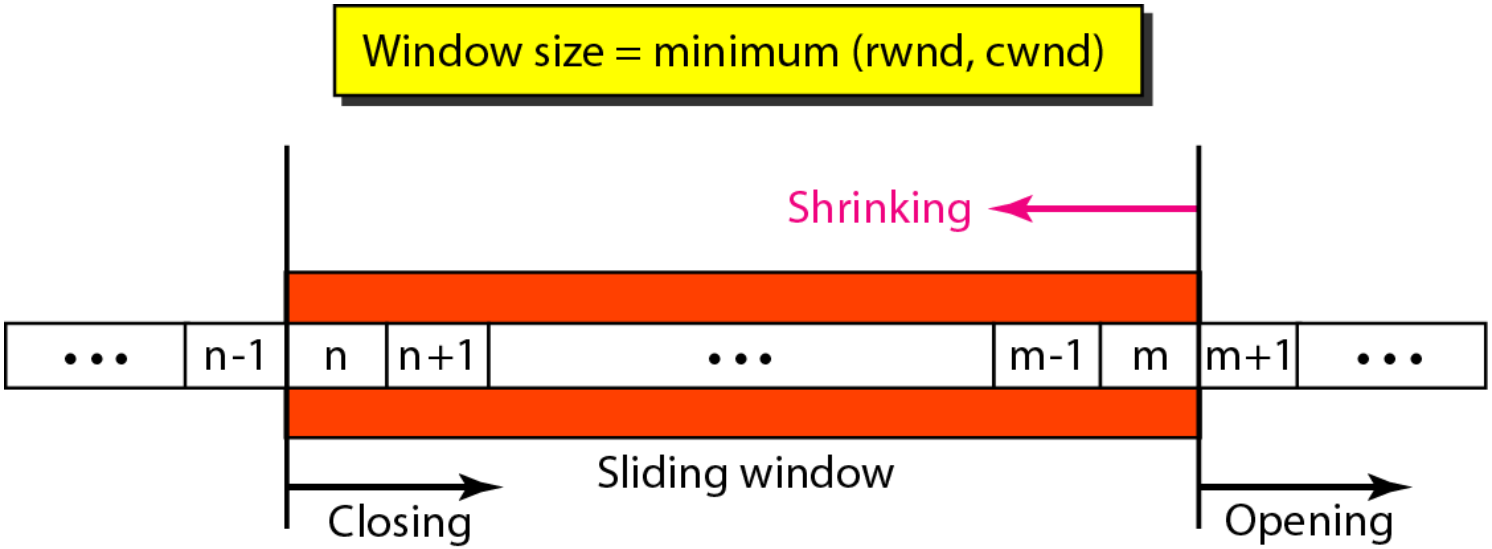
Note

The FIN + ACK segment consumes one sequence number if it does not carry data.

Connection termination using 2) Half-close



Receiver-based **Flow Control**: ***Sliding window***



rwnd: receiver window, **cwnd**: congestion window

Note

A sliding window is used to make transmission more efficient as well as to control the flow of data so that the destination does not become overwhelmed with data. TCP sliding windows are byte-oriented.

What is the value of the receiver window (rwnd) for host A if the receiver, host B, has a buffer size of 5000 bytes and 1000 bytes of received and unprocessed data?

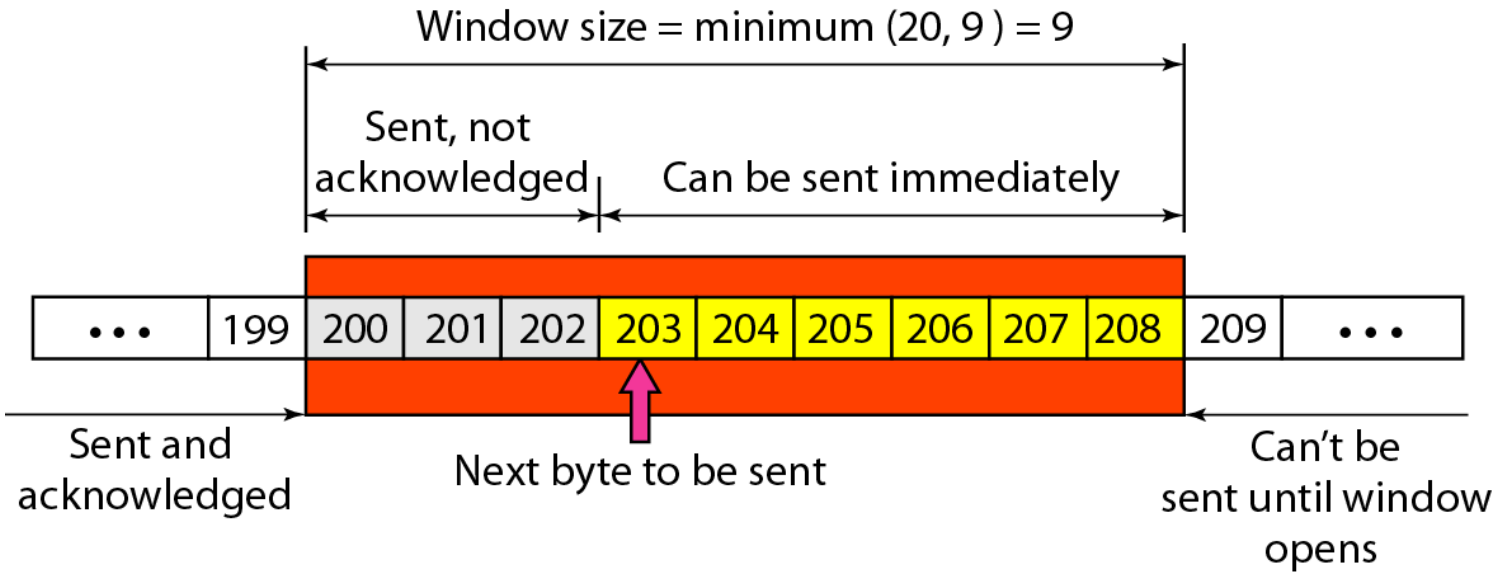
Solution

The value of rwnd = 5000 − 1000 = 4000. Host B can receive only 4000 bytes of data before overflowing its buffer. Host B advertises this value in its next segment to A.

What is the size of the window for host A if the value of rwnd is 3000 bytes and the value of cwnd is 3500 bytes?

Solution

The size of the window is the smaller of rwnd and cwnd, which is 3000 bytes.



- Sender has sent bytes up to 202. cwnd: 20 bytes.
- Receiver has sent an ack number of 200 with an rwnd of 9 bytes. Size of the sender window: 9 bytes.
- Bytes 200 to 202 are sent, but not acknowledged.
- Bytes (203-208) can be sent without worrying about acknowledgment.
- Bytes 209 and above cannot be sent.

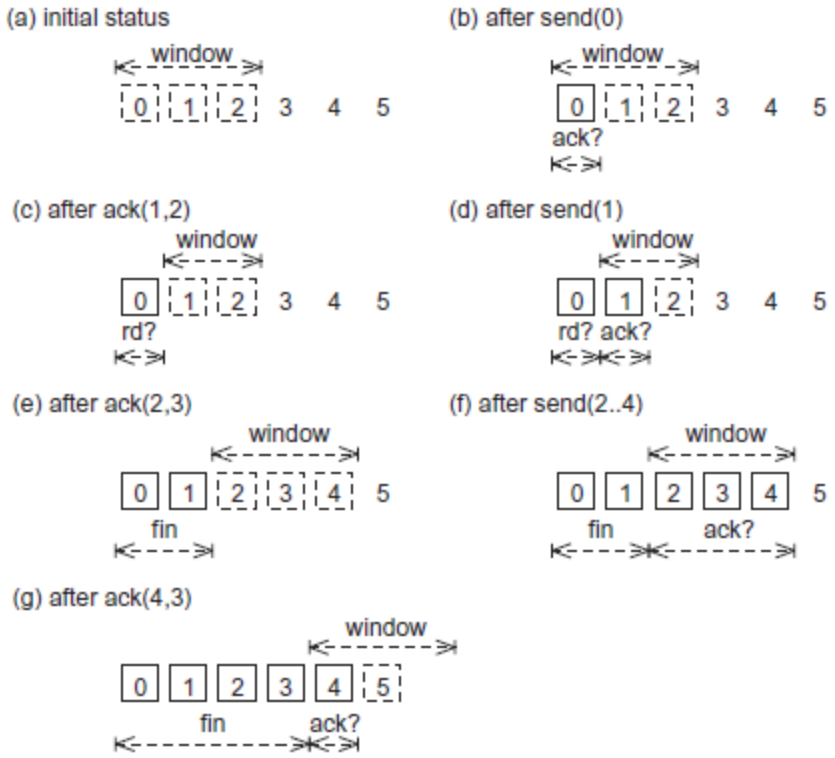


FIGURE 5.14 TCP receiver window progress.

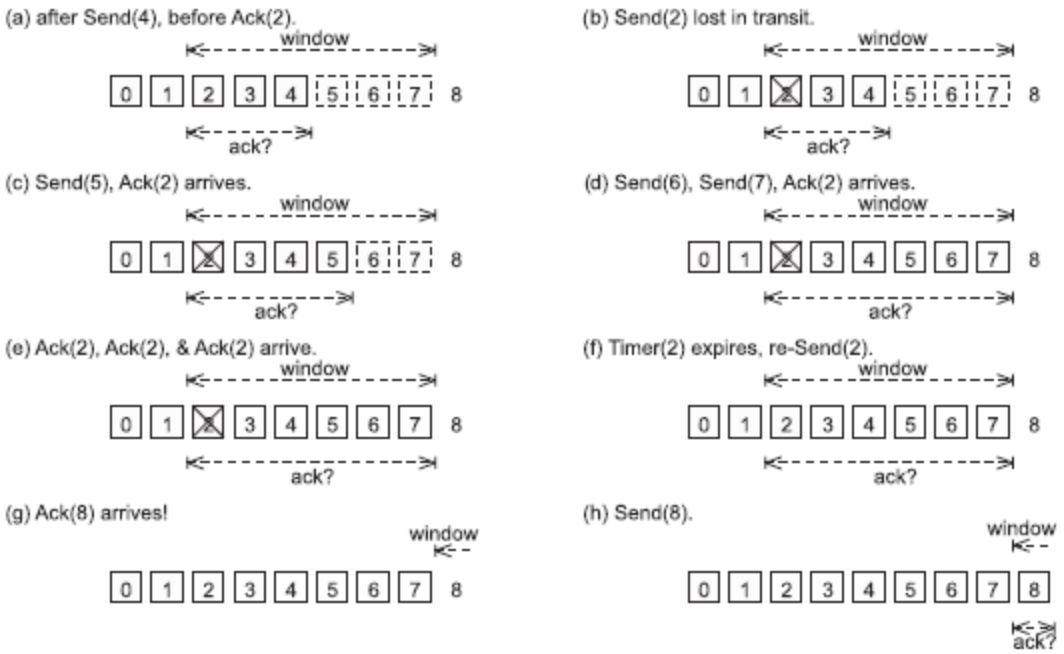


FIGURE 5.15 TCP window progress in response to lost segment.

Transmitter-based Flow Control

Textbook:

- Fig 5.16: TCP transmitter flow control mode
- Fig 5.17: TCP transmitter flow control recovery modes

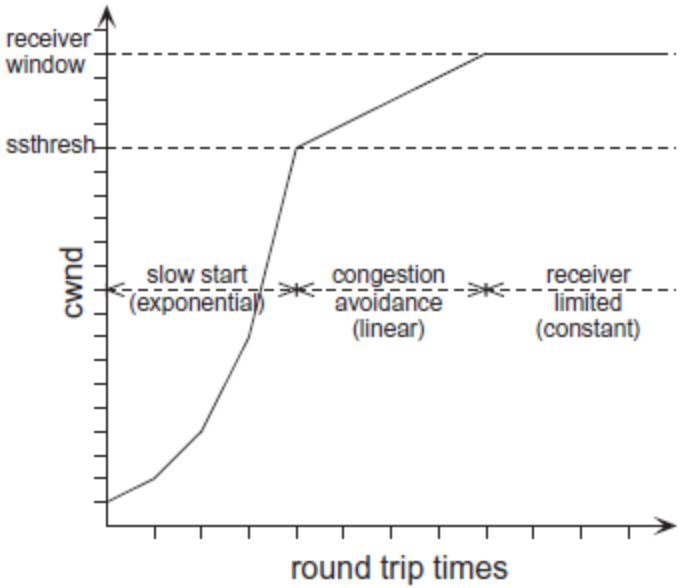


FIGURE 5.16 TCP transmitter flow control modes.

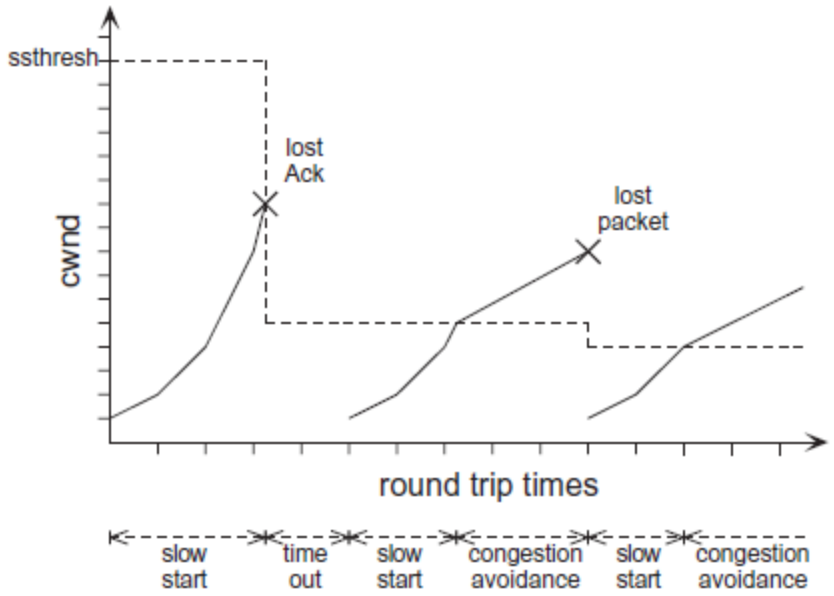


FIGURE 5.17 TCP transmitter flow control recovery modes.

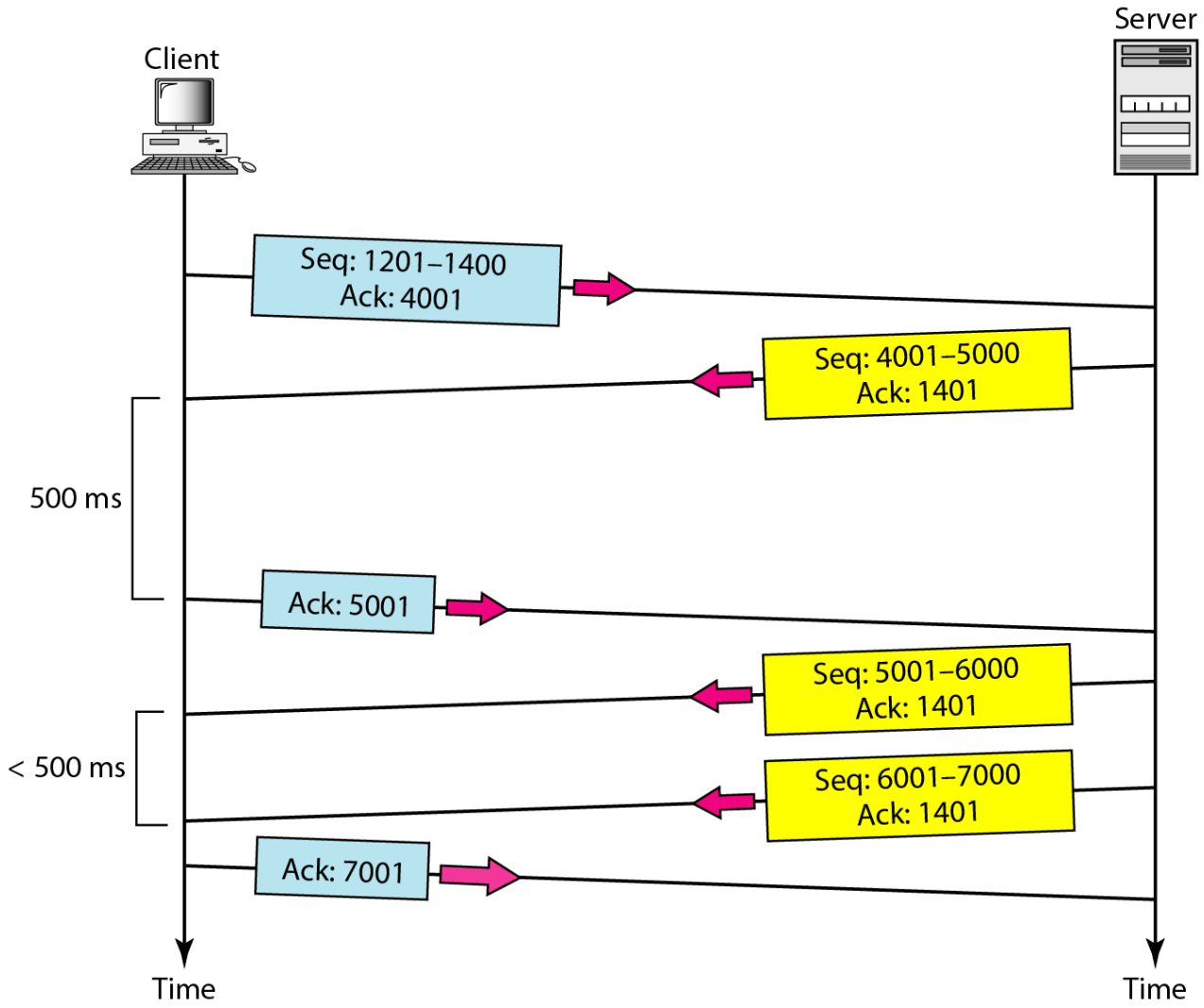
Note

In modern implementations, a retransmission occurs if the retransmission timer expires or three duplicate ACK segments have arrived.

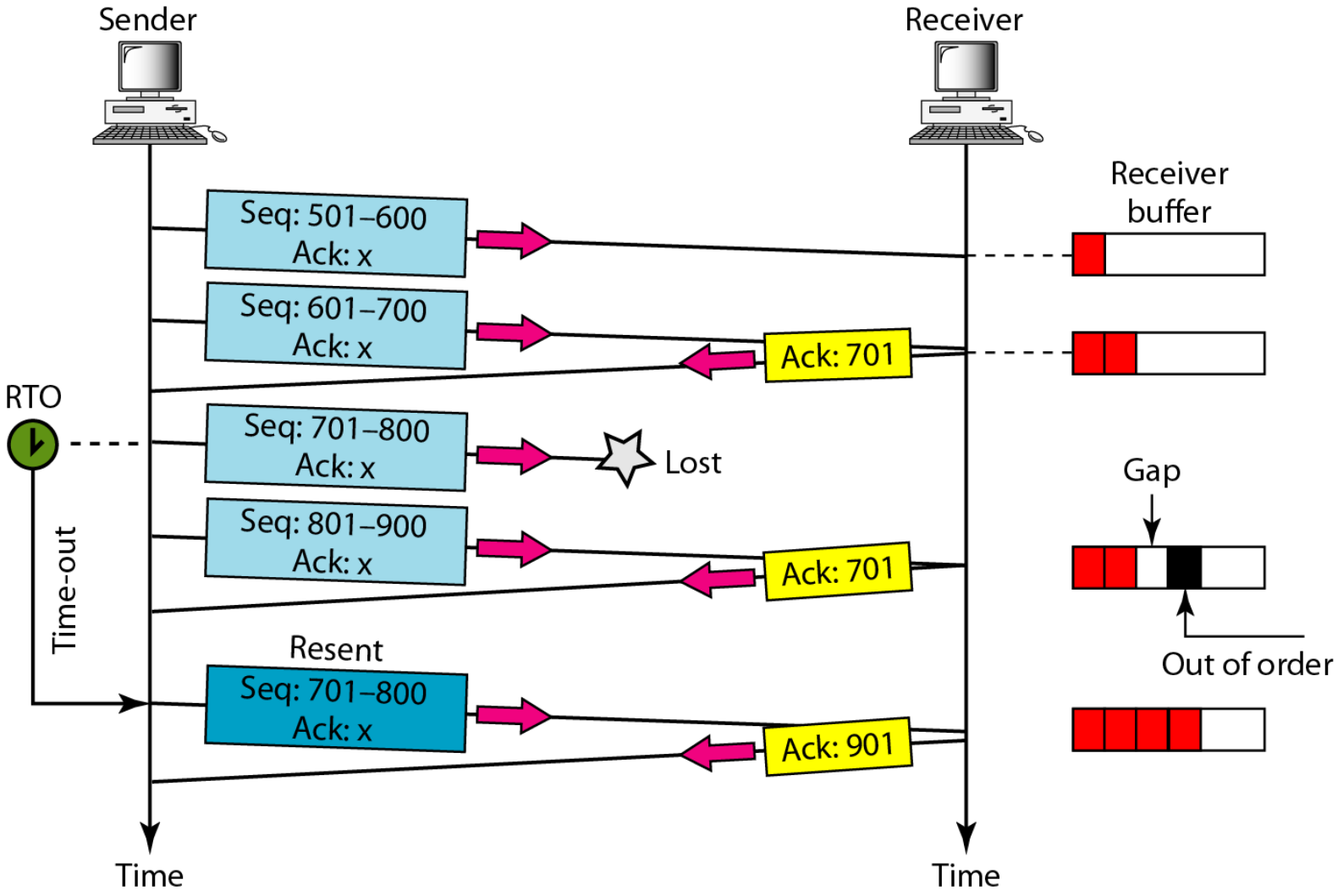
Note

No retransmission timer is set for an ACK segment.

Normal operation



Lost segment



Note

The receiver TCP delivers only ordered data to the process.

Fast retransmission (and Fast recovery)