Chapter 12

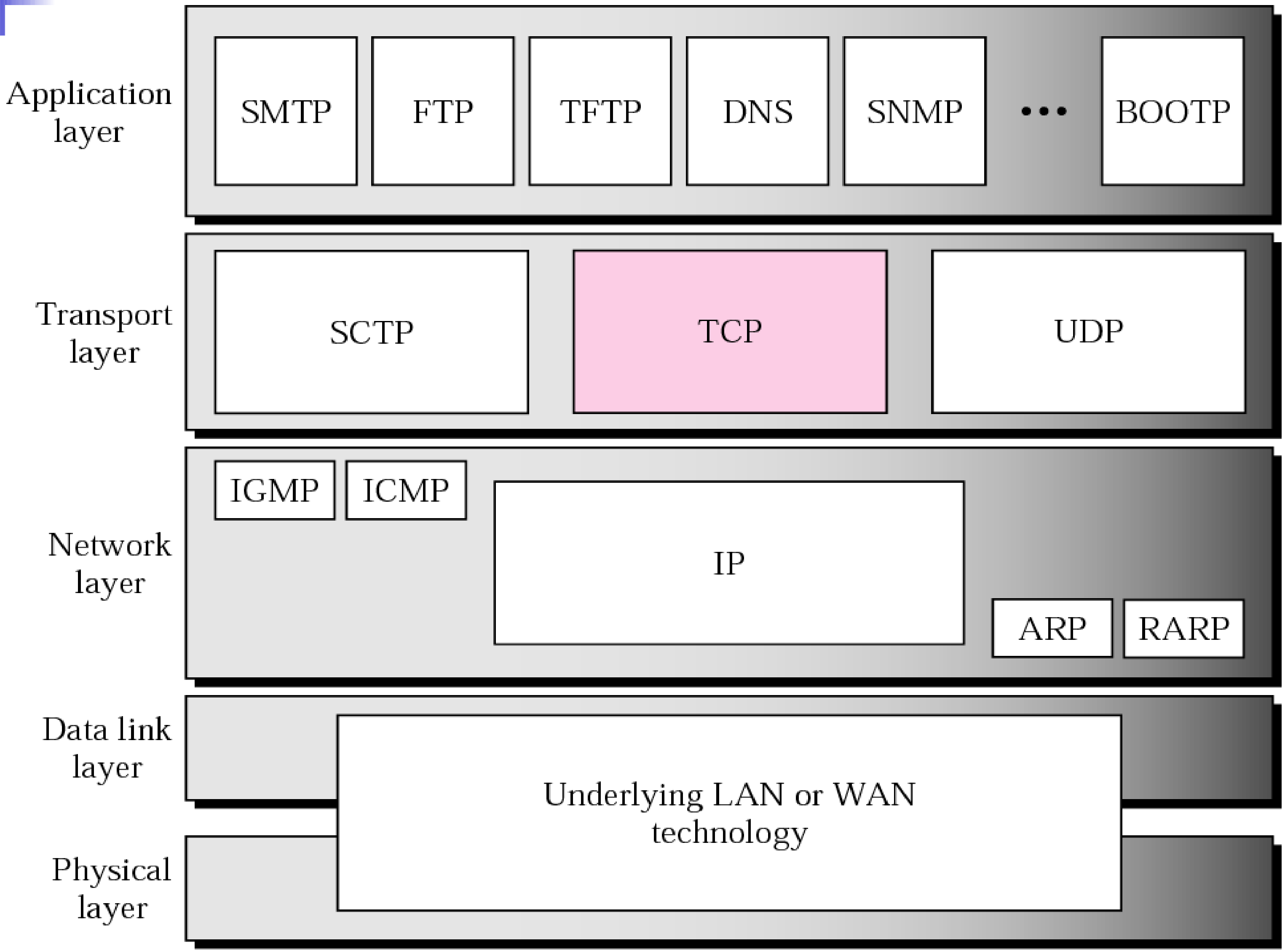
Transmission Control Protocol

Objectives

Upon completion you will be able to:

- *Be able to name and understand the services offered by TCP*
- *Understand TCP's flow and error control and congestion control*
- *Be familiar with the fields in a TCP segment*
- *Understand the phases in a connection-oriented connection*
- *Understand the TCP transition state diagram*
- *Be able to name and understand the timers used in TCP*
- *Be familiar with the TCP options*

Figure 12.1 *TCP/IP protocol suite*



12.1 TCP SERVICES

We explain the services offered by TCP to the processes at the application layer.

The topics discussed in this section include:

Process-to-Process Communication
Stream Delivery Service
Full-Duplex Communication
Connection-Oriented Service
Reliable Service

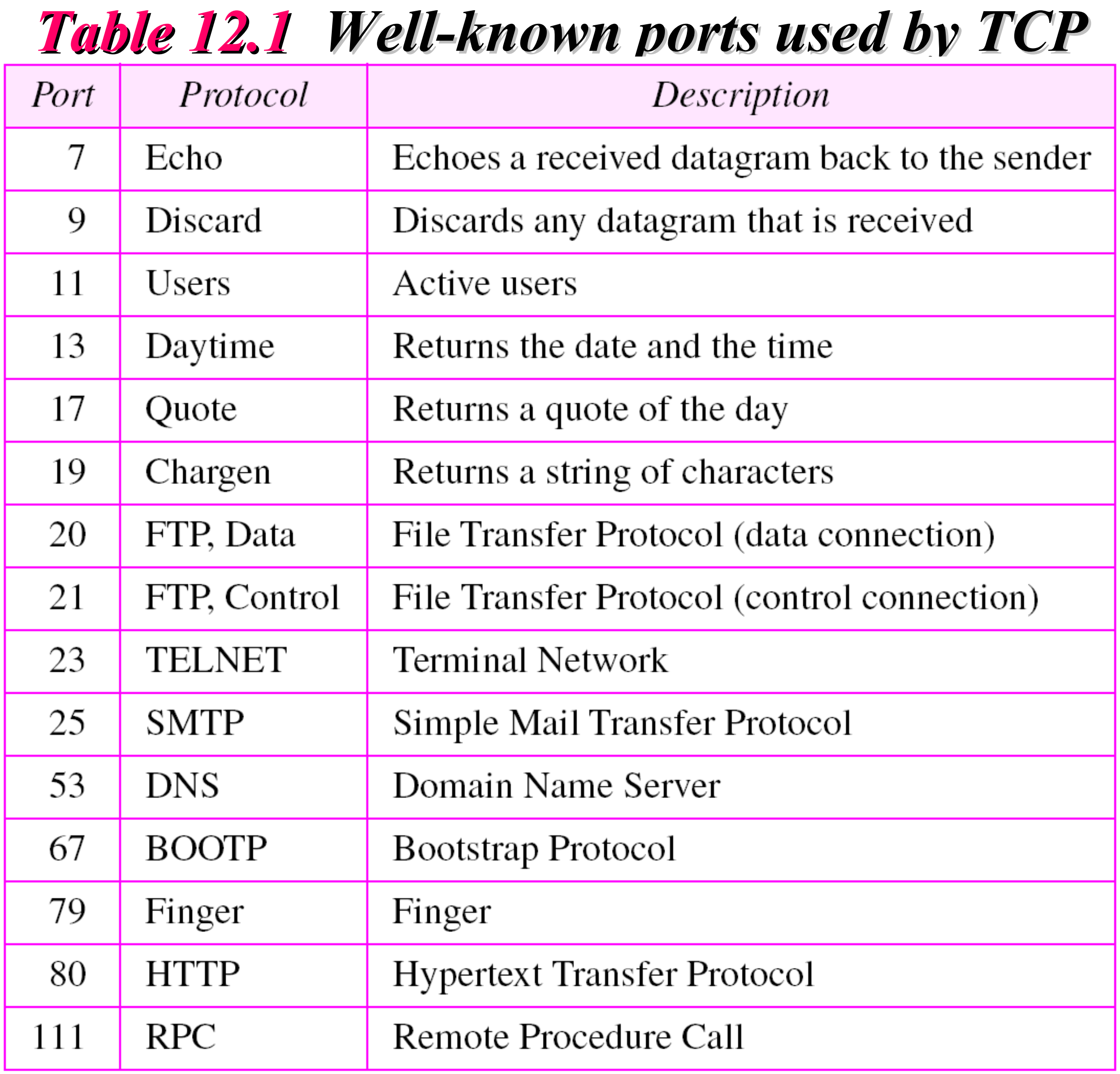


Table 12.1 Well-known ports used by TCP

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

Figure 12.2 *Stream delivery*

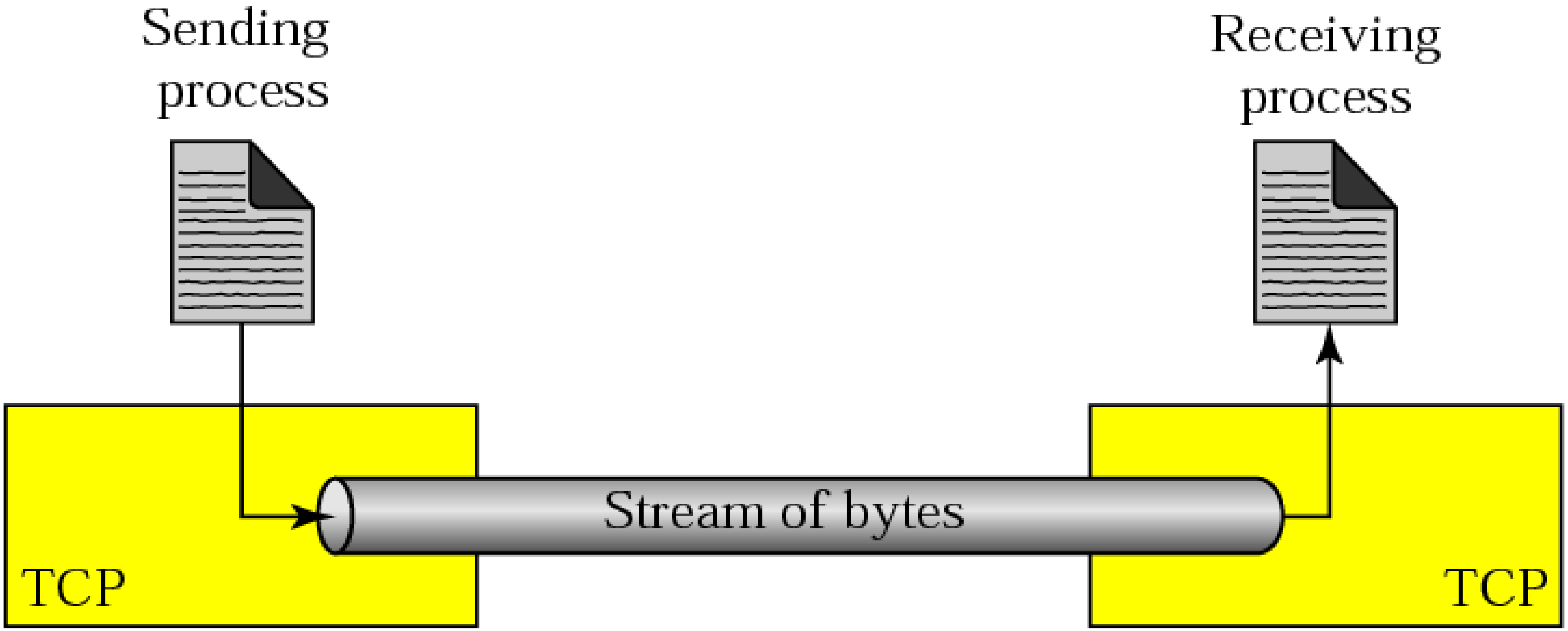


Figure 12.3 *Sending and receiving buffers*

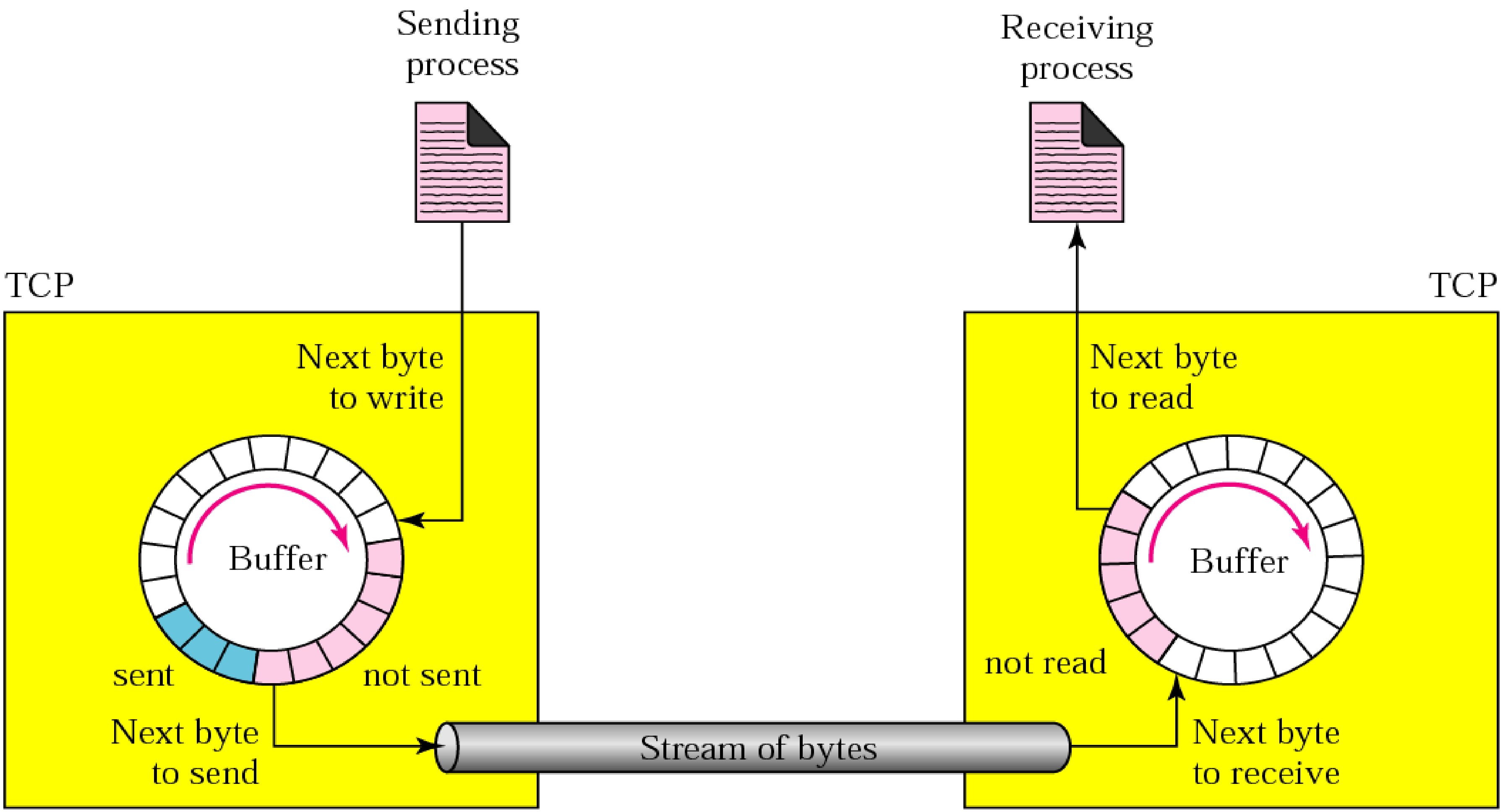
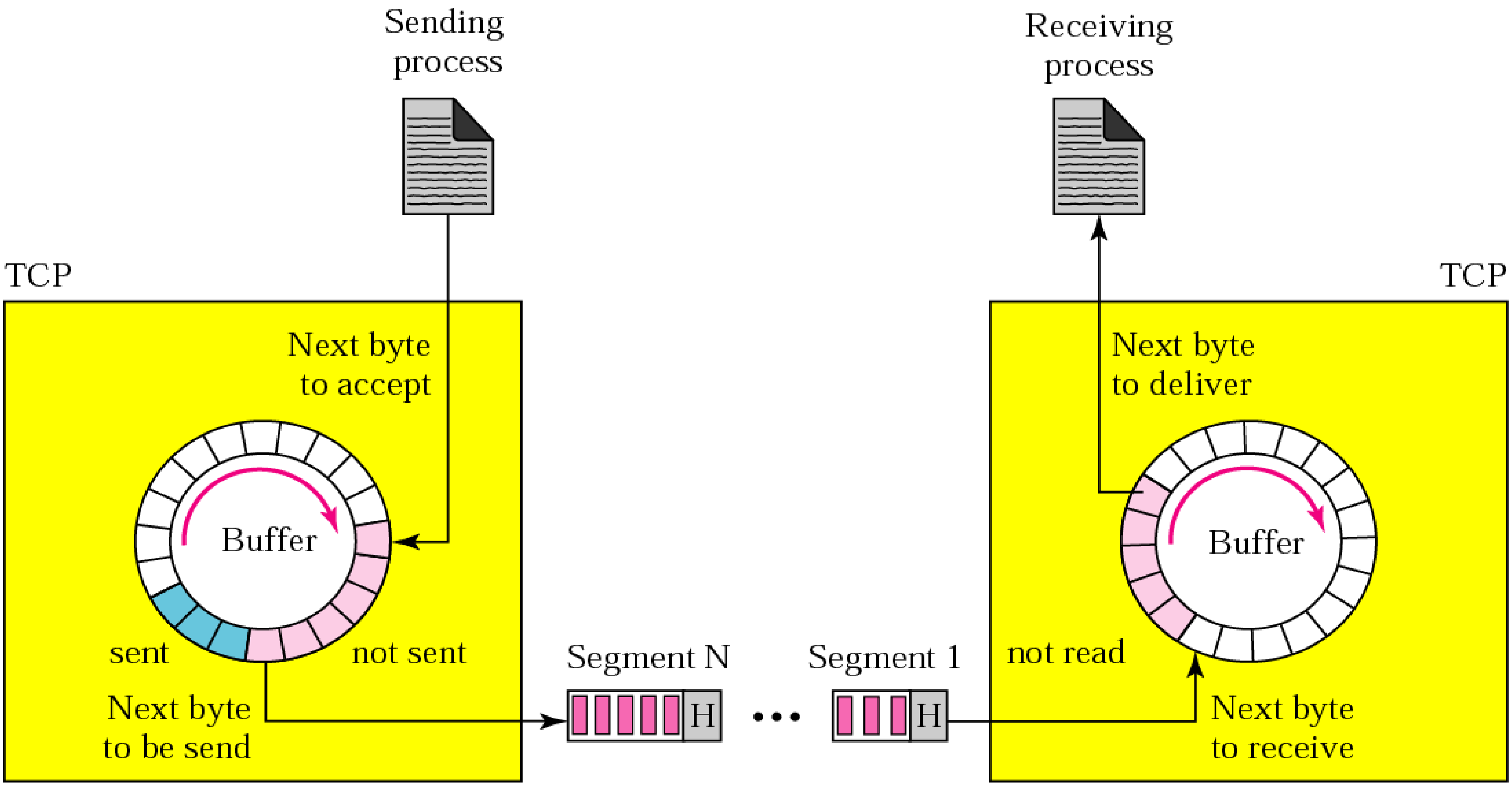


Figure 12.4 *TCP segments*



12.2 TCP FEATURES

To provide the services mentioned in the previous section, TCP has several features that are briefly summarized in this section.

The topics discussed in this section include:

Numbering System
Flow Control
Error Control
Congestion Control

Note:

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

Example 2

Suppose a TCP connection is transferring a file of 5000 bytes. The first byte is numbered 10001. What are the sequence numbers for each segment if data is sent in five segments, each carrying 1000 bytes?

Solution

The following shows the sequence number for each segment:

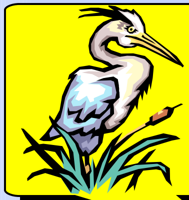
Segment 1 ➡ *Sequence Number: 10,001 (range: 10,001 to 11,000)*

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)*



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.

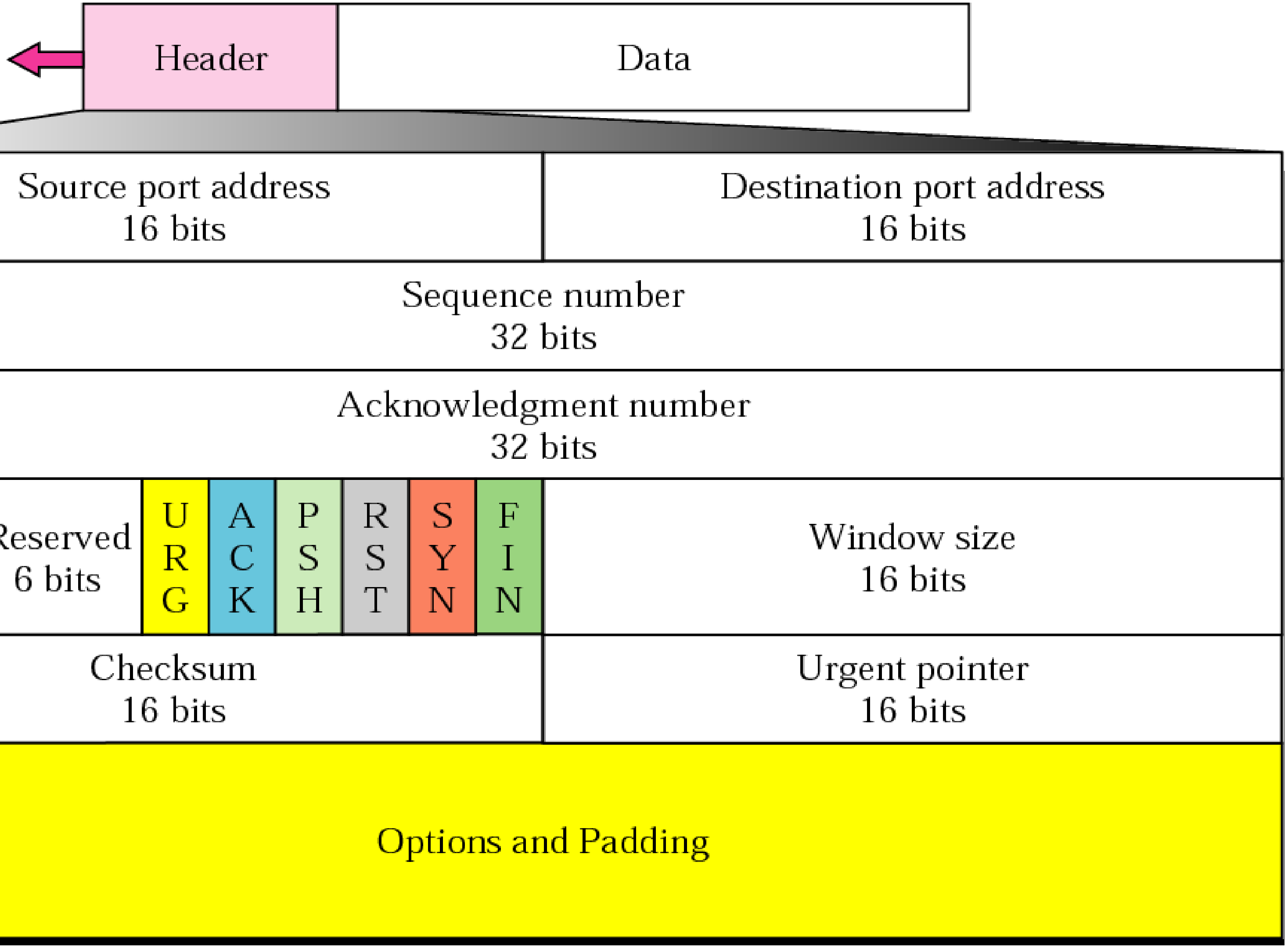
12.3 SEGMENT

A packet in TCP is called a segment

The topics discussed in this section include:

Format
Encapsulation

Figure 12.5 *TCP segment format*



Header

Data

Source port address
16 bits

Destination port address
16 bits

Sequence number
32 bits

Acknowledgment number
32 bits

HLEN
4 bits

Reserved
6 bits

URG

ACK

PSH

RST

SYN

FIN

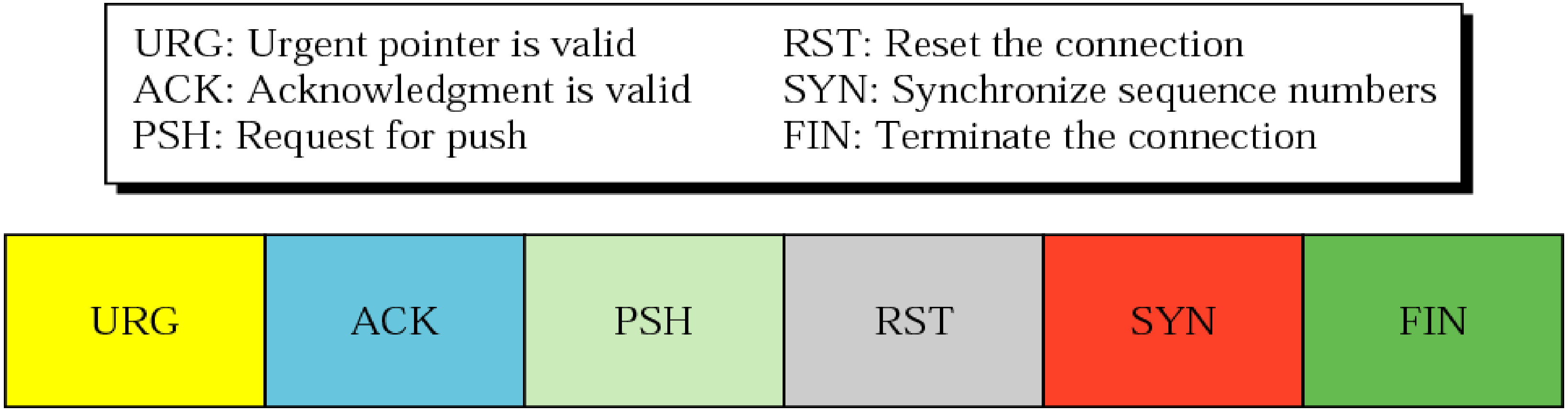
Window size
16 bits

Checksum
16 bits

Urgent pointer
16 bits

Options and Padding

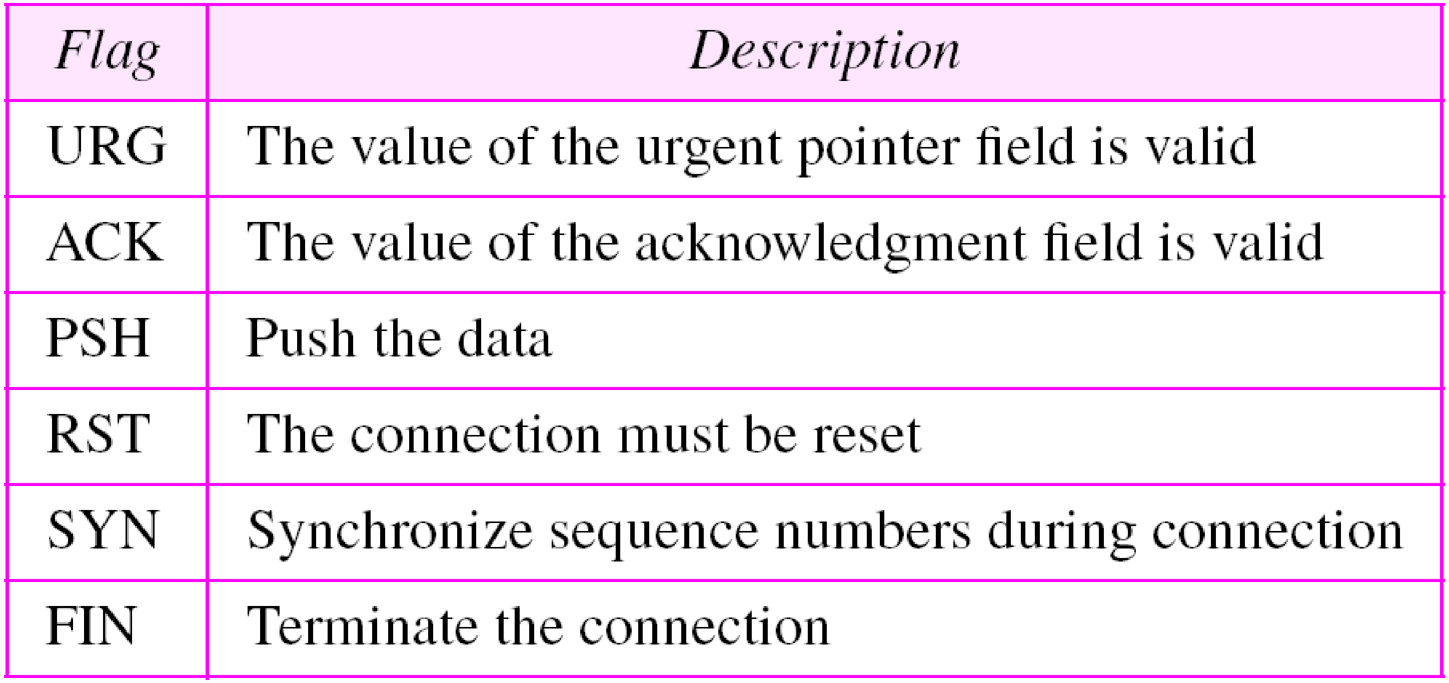
Figure 12.6 *Control field*



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

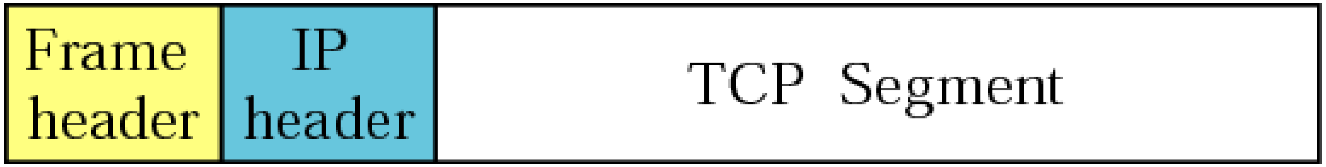
URG	ACK	PSH	RST	SYN	FIN

Table 12.2 Description of flags in the control field



Flag	*Description*
URG	The value of the urgent pointer field is valid
ACK	The value of the acknowledgment field is valid
PSH	Push the data
RST	The connection must be reset
SYN	Synchronize sequence numbers during connection
FIN	Terminate the connection

Figure 12.8 *Encapsulation and decapsulation*



Frame header

IP header

TCP Segment

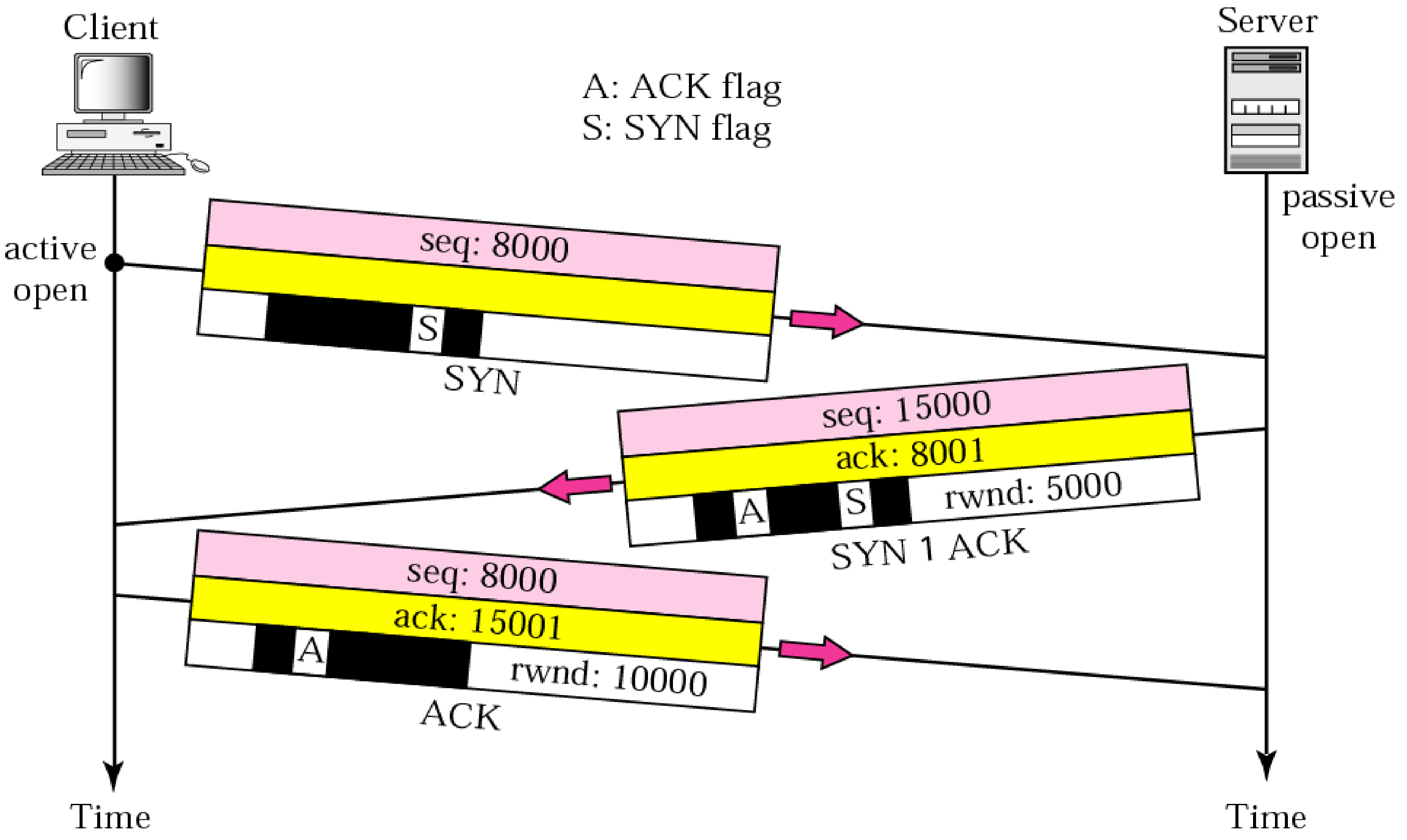
12.4 A TCP CONNECTION

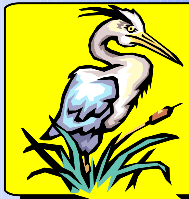
TCP is connection-oriented. A connection-oriented transport protocol establishes a virtual path between the source and destination. All of the segments belonging to a message are then sent over this virtual path. A connection-oriented transmission requires three phases: connection establishment, data transfer, and connection termination.

The topics discussed in this section include:

Connection Establishment
Data Transfer
Connection Termination
Connection Reset

Figure 12.9 *Connection establishment using three-way handshaking*





Note:

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

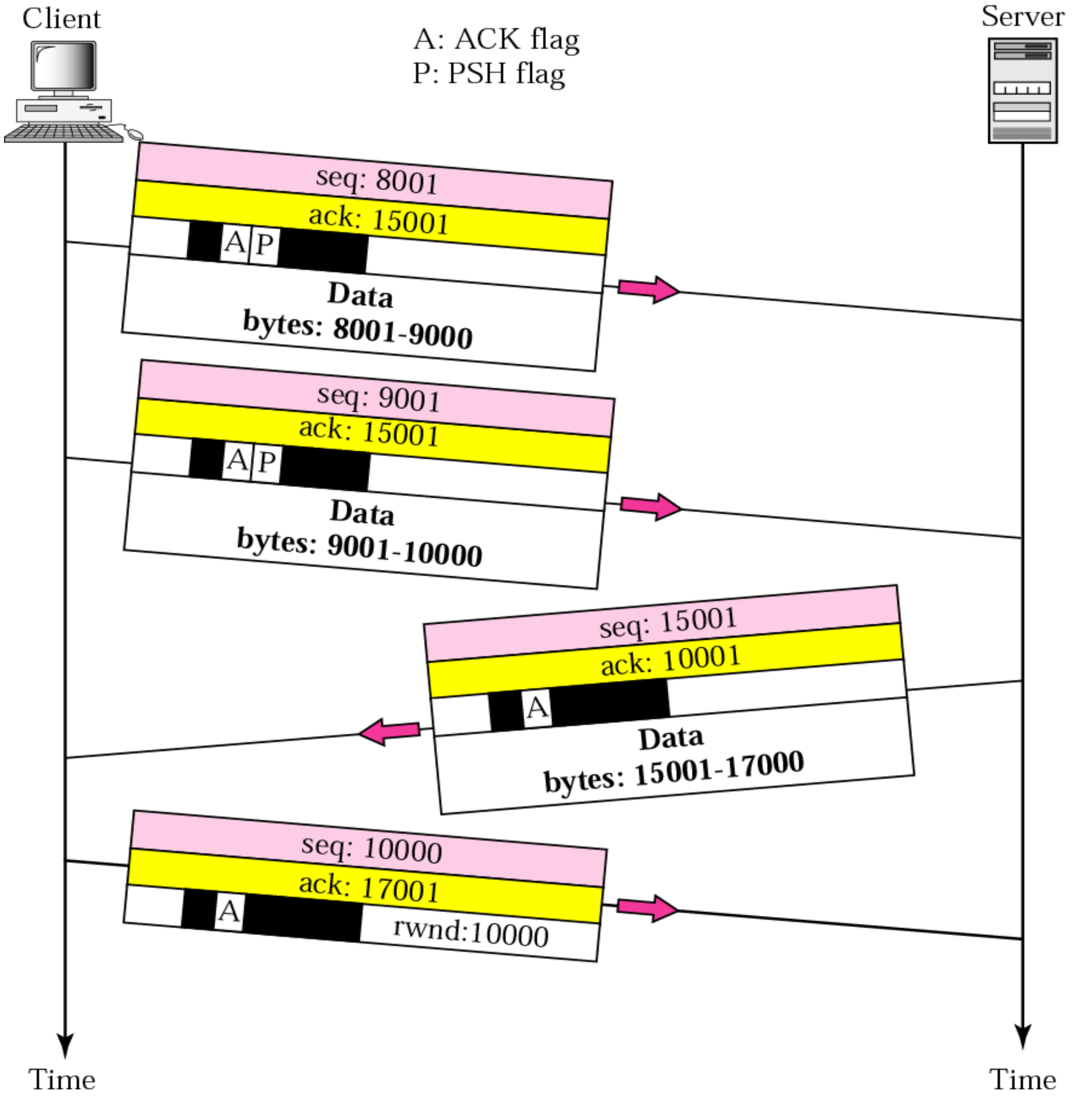
Note:

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.

Figure 12.10 *Data transfer*


Client
A: ACK flag
P: PSH flag
Server
seq: 8001
ack: 15001
A
P
Data
bytes: 8001-9000
seq: 9001
ack: 15001
A
P
Data
bytes: 9001-10000
seq: 15001
ack: 10001
A
Data
bytes: 15001-17000
seq: 10000
ack: 17001
A
rwnd:10000
Time
Time