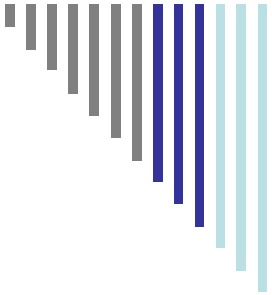
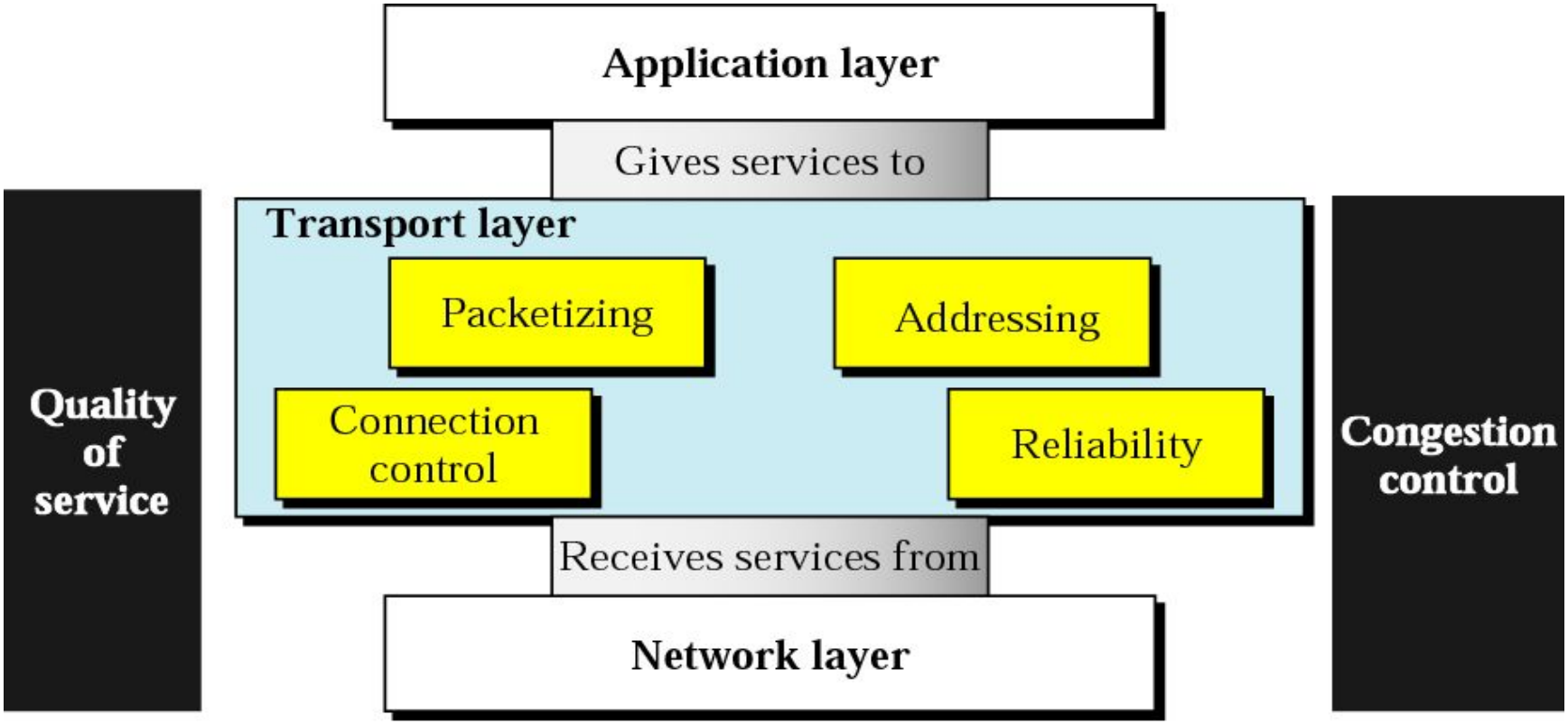
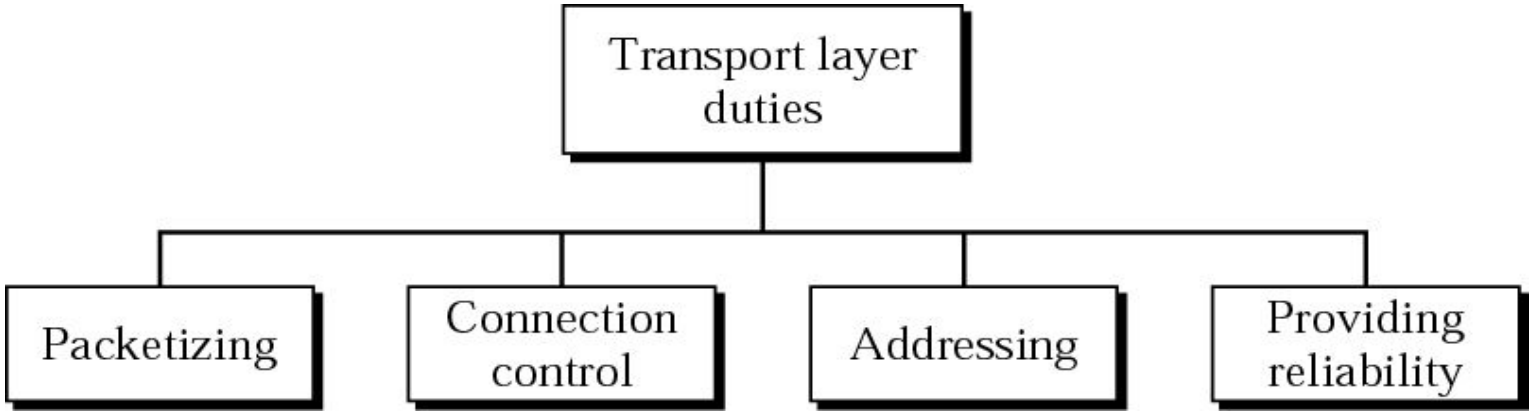


Transport Layer

Position of transport layer



Transport layer duties



Chapters

Chapter 22 ***Process-to-Process Delivery***

Chapter 23 ***Congestion Control and QoS***

Chapter 22

Process-to-Process Delivery:

UDP and TCP

22.1 Process-to-Process Delivery

Client-Server Paradigm

Addressing

Multiplexing and Demultiplexing

Connectionless/Connection-Oriented

Reliable/Unreliable

Note:

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

Figure 22.1 Types of data deliveries

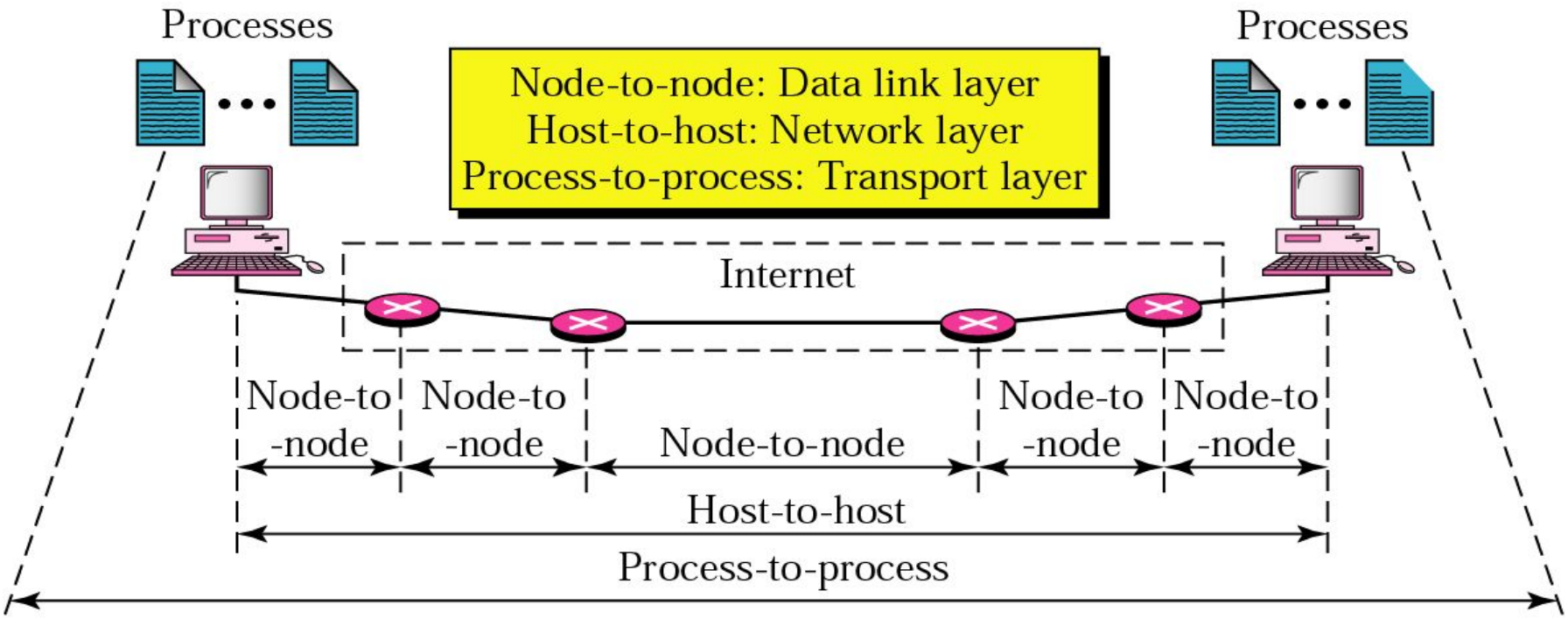


Figure 22.2 Port numbers

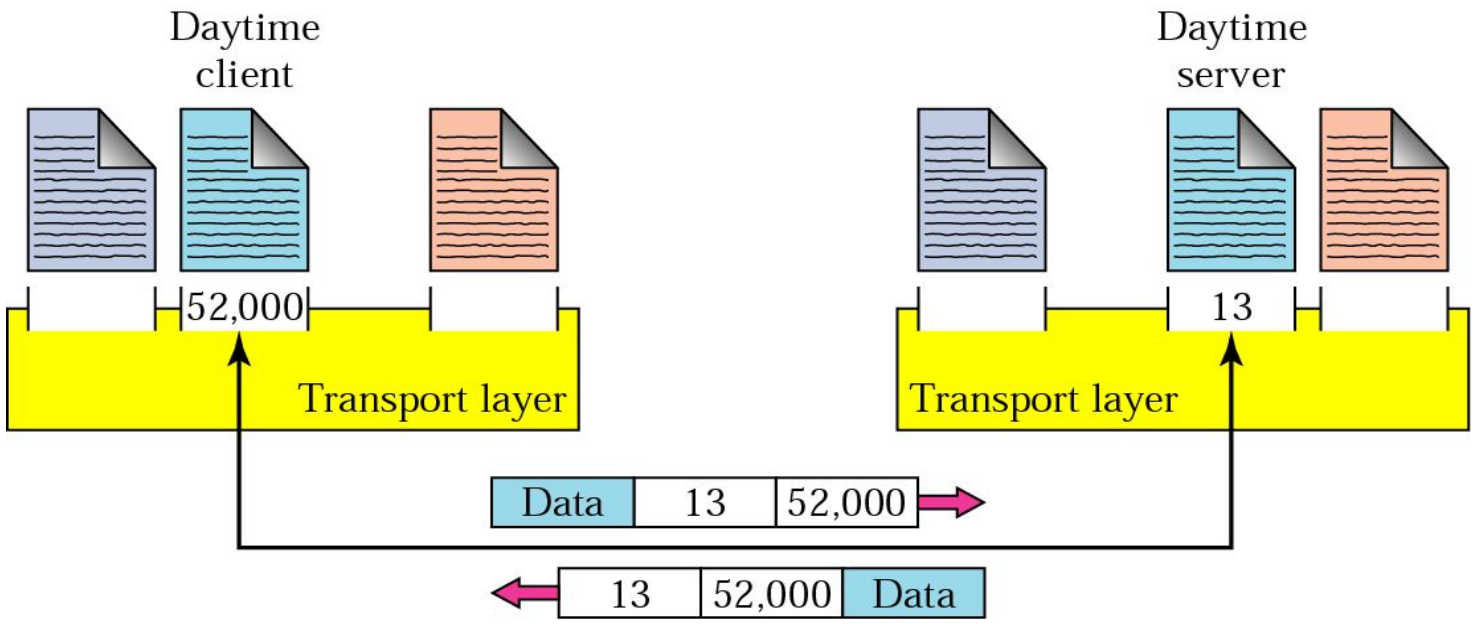


Figure 22.3 IP addresses versus port numbers

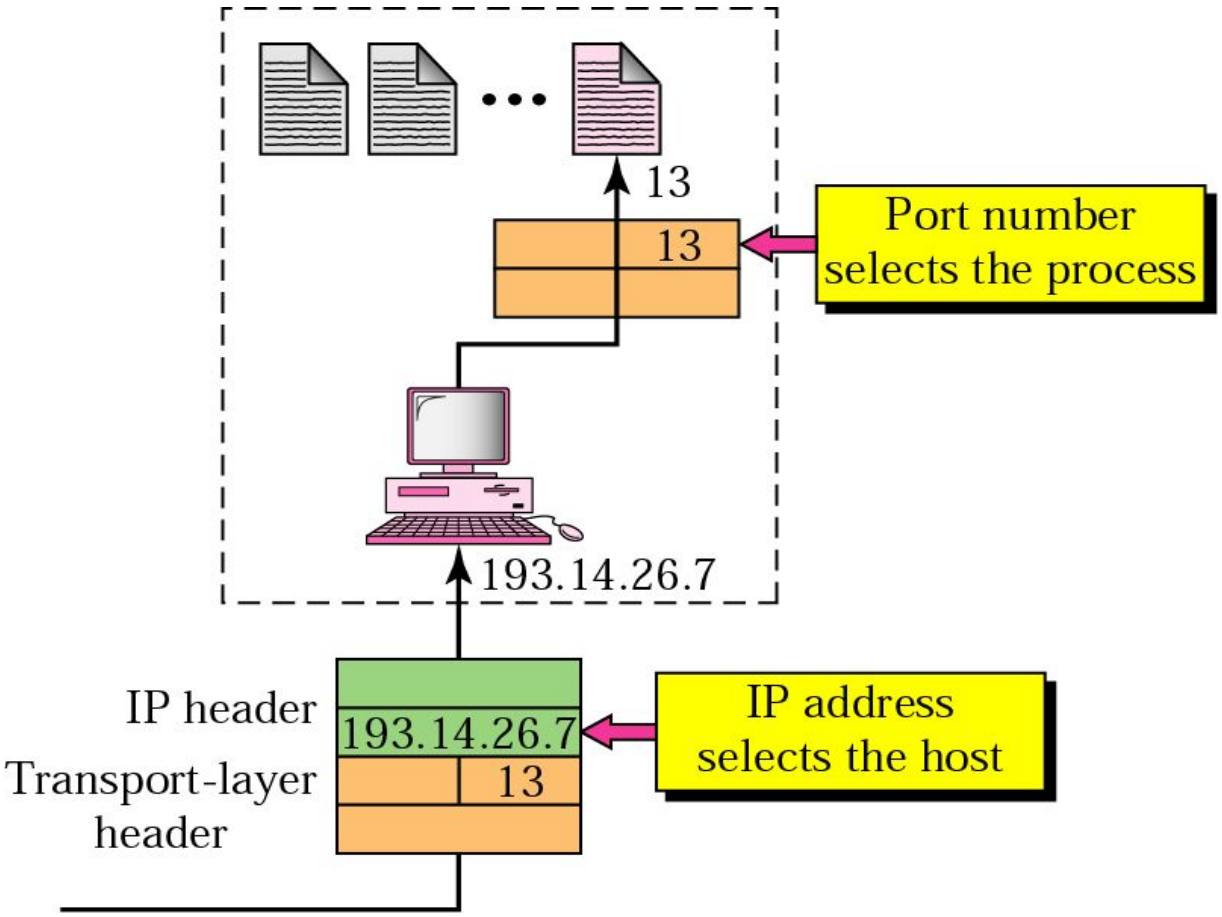


Figure 22.4 IANA ranges

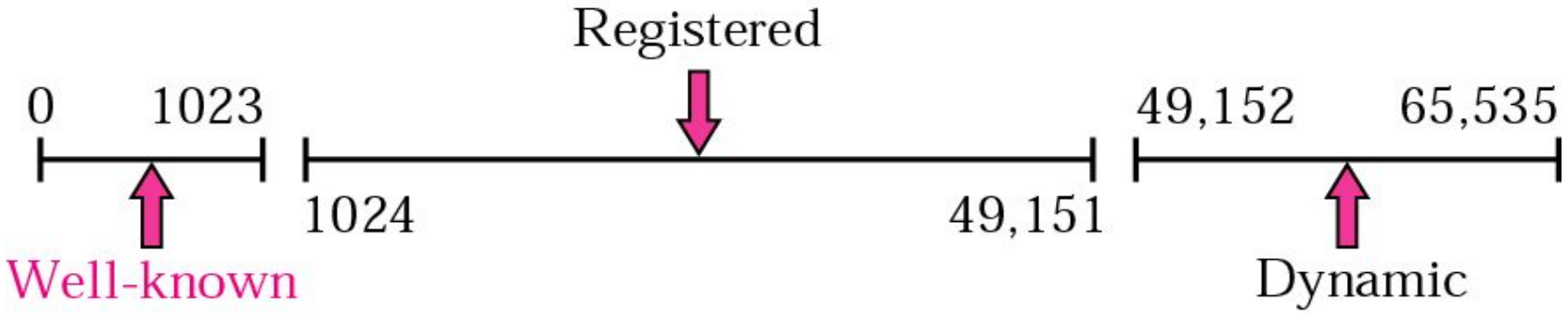


Figure 22.5 Socket address

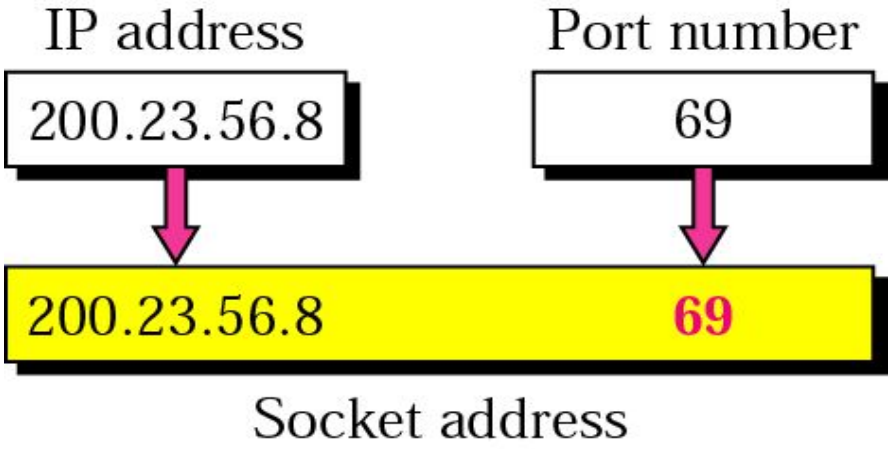


Figure 22.6 Multiplexing and demultiplexing

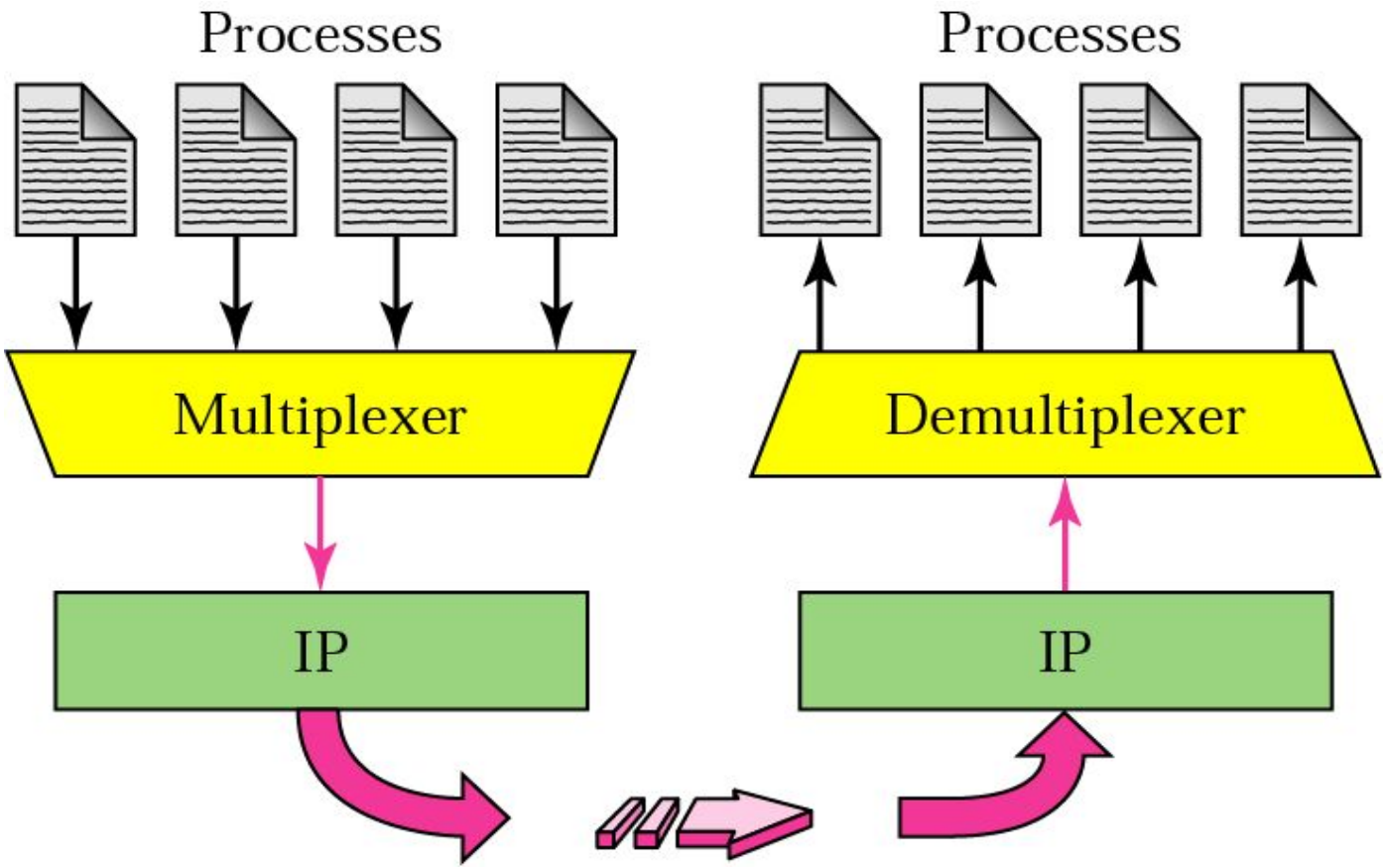


Figure 22.7 Connection establishment

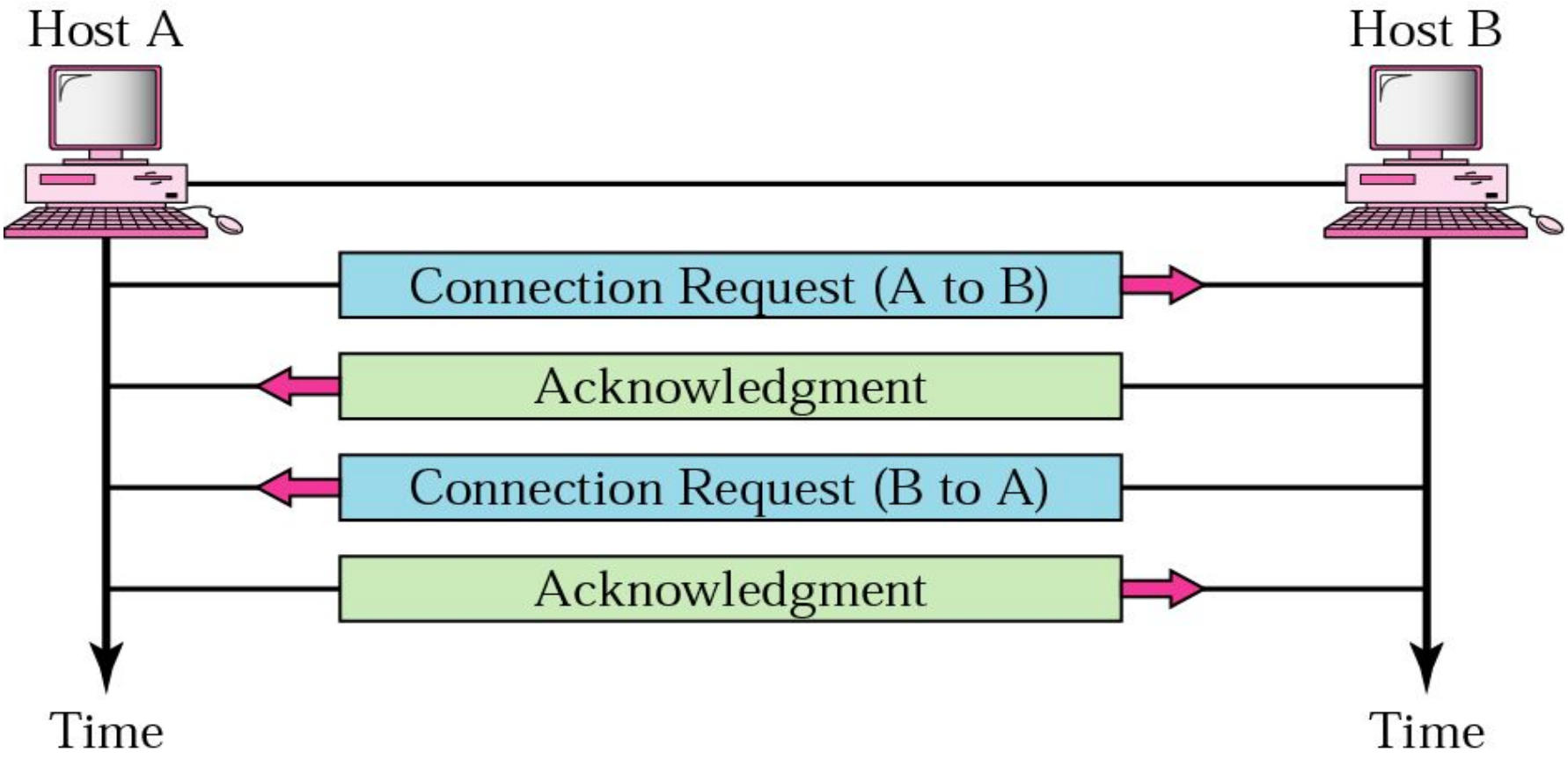


Figure 22.8 Connection termination

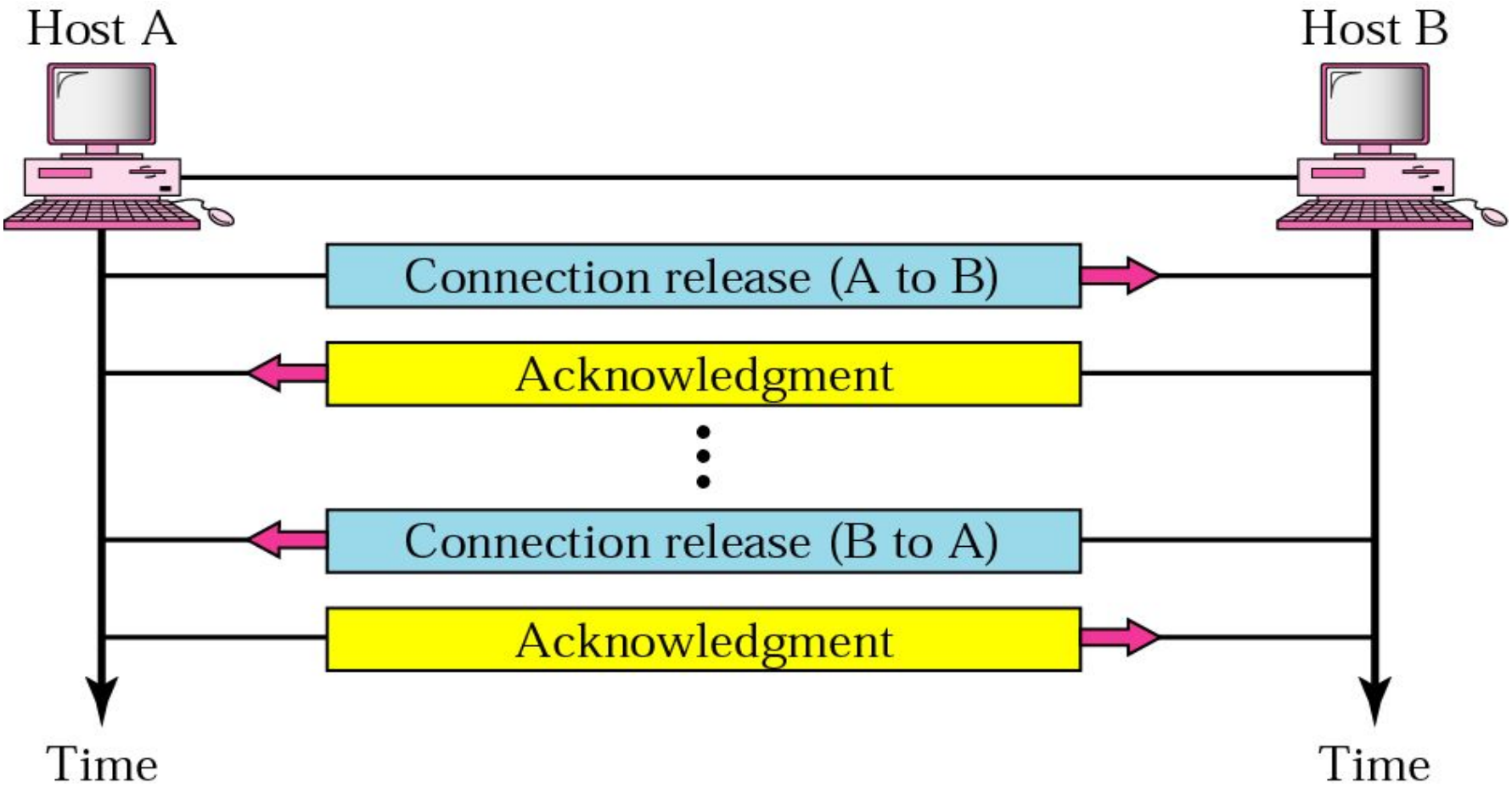
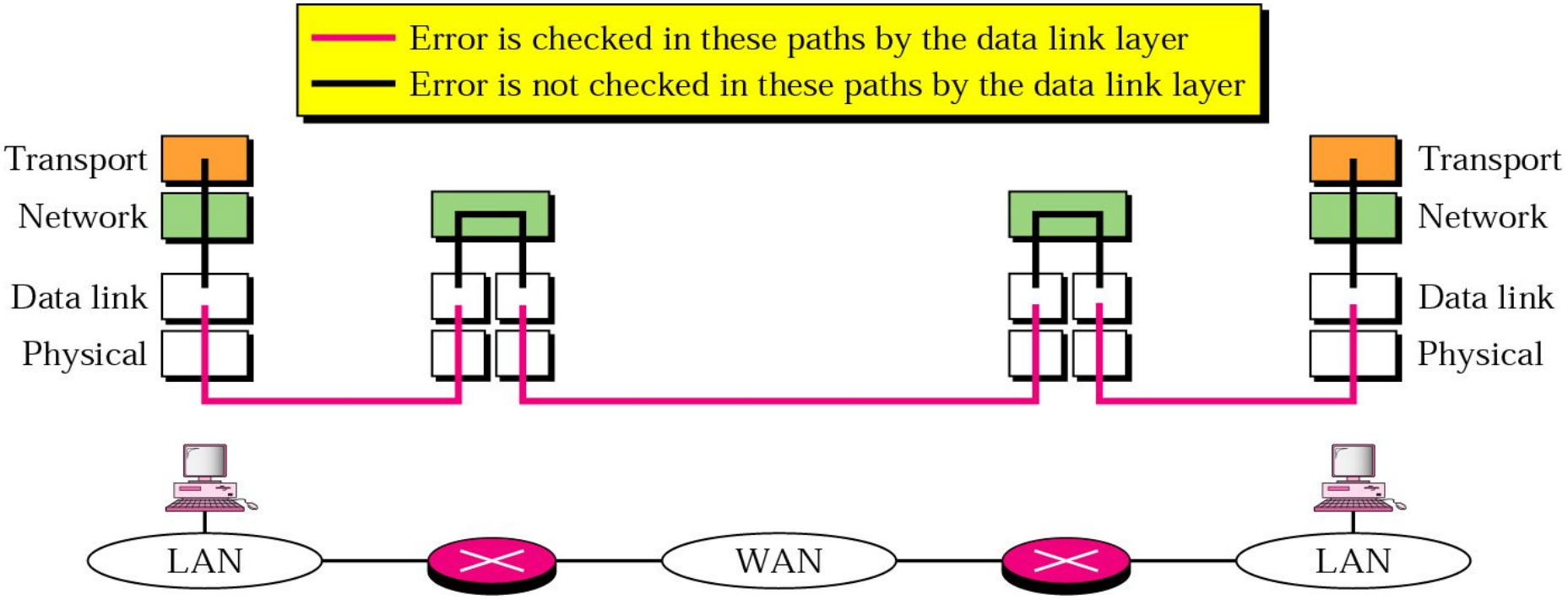


Figure 22.9 Error control



22.2 UDP

Port Numbers

User Datagram

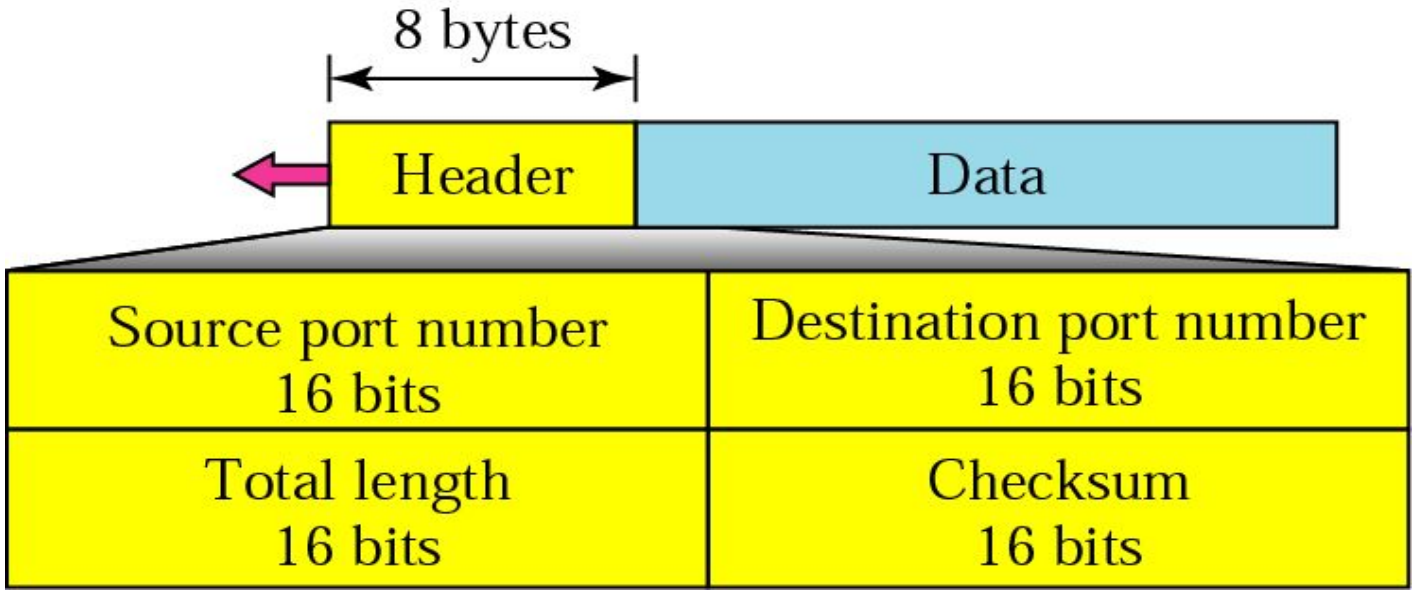
Applications

UDP is a connectionless, unreliable protocol that has no flow and error control. It uses port numbers to multiplex data from the application layer.

Table 22.1 Well-known ports used by 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)

Figure 22.10 User datagram format



Note:

The calculation of checksum and its inclusion in the user datagram are optional.

UDP is a convenient transport-layer protocol for applications that provide flow and error control. It is also used by multimedia applications.

22.3 TCP

Port Numbers

Services

Sequence Numbers

Segments

Connection

Transition Diagram

Flow and Error Control

Silly Window Syndrome

Table 22.2 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 22.11 Stream delivery

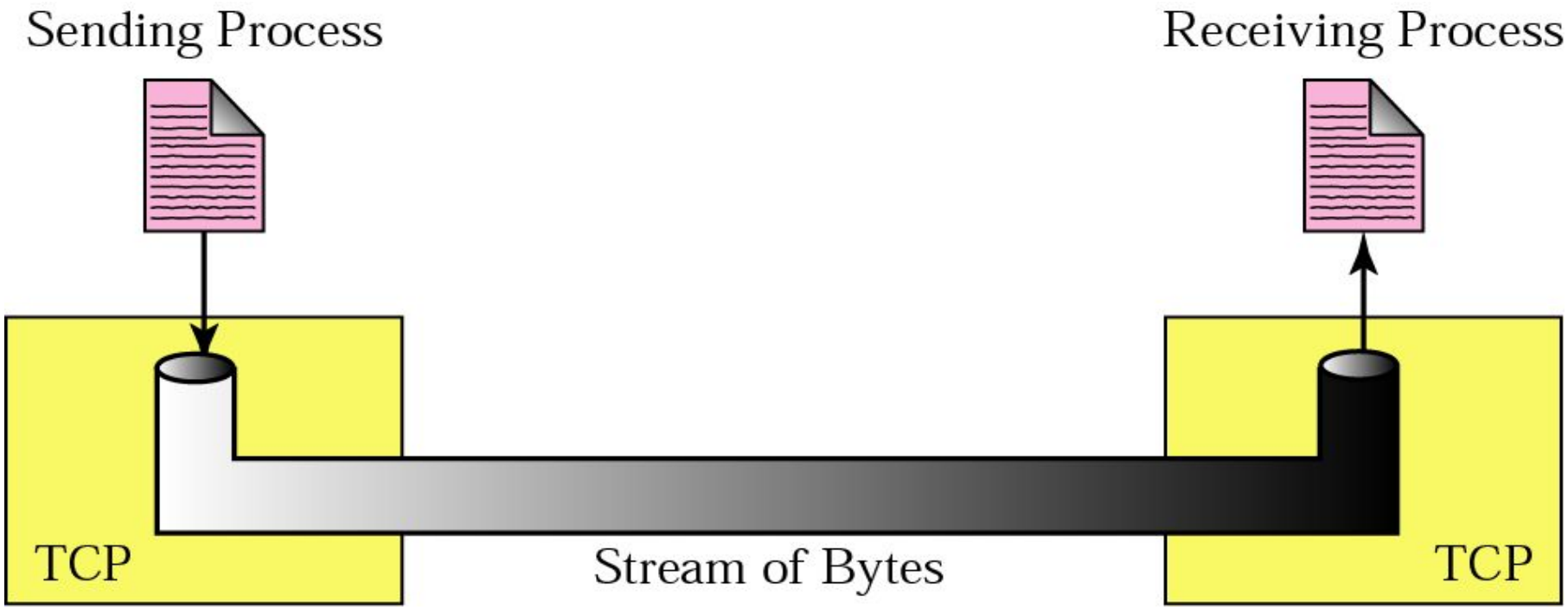


Figure 22.12 Sending and receiving buffers

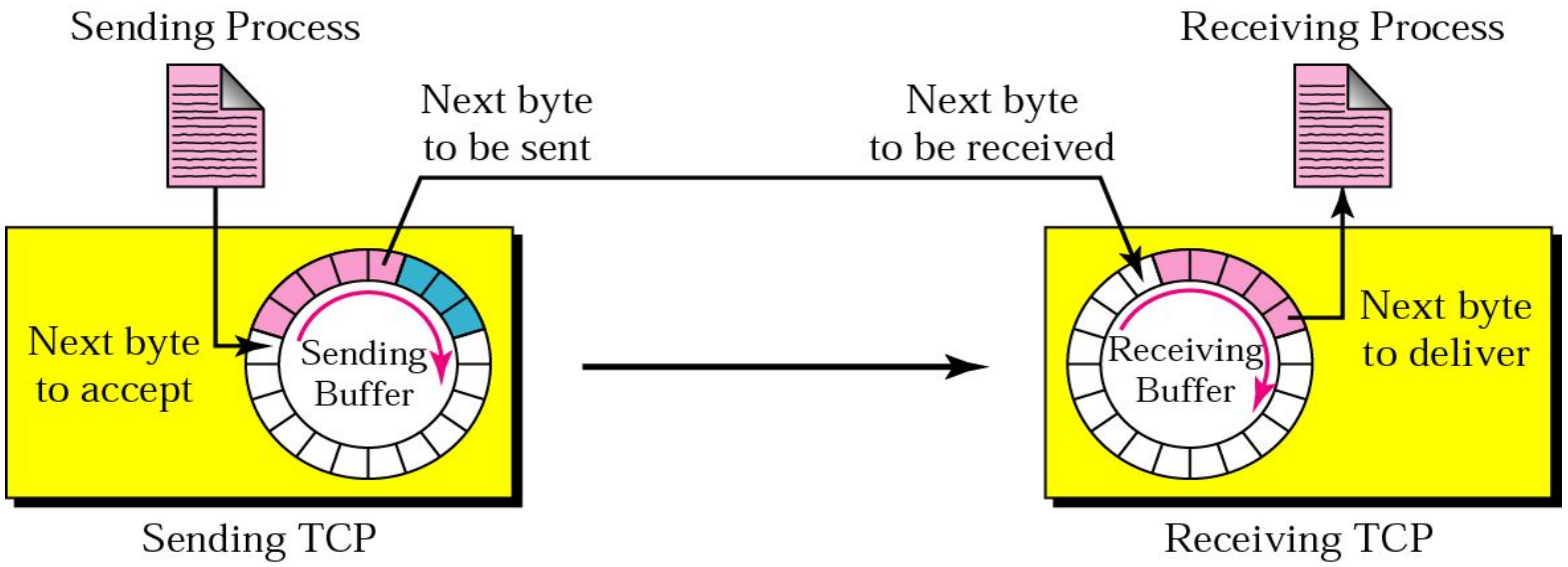
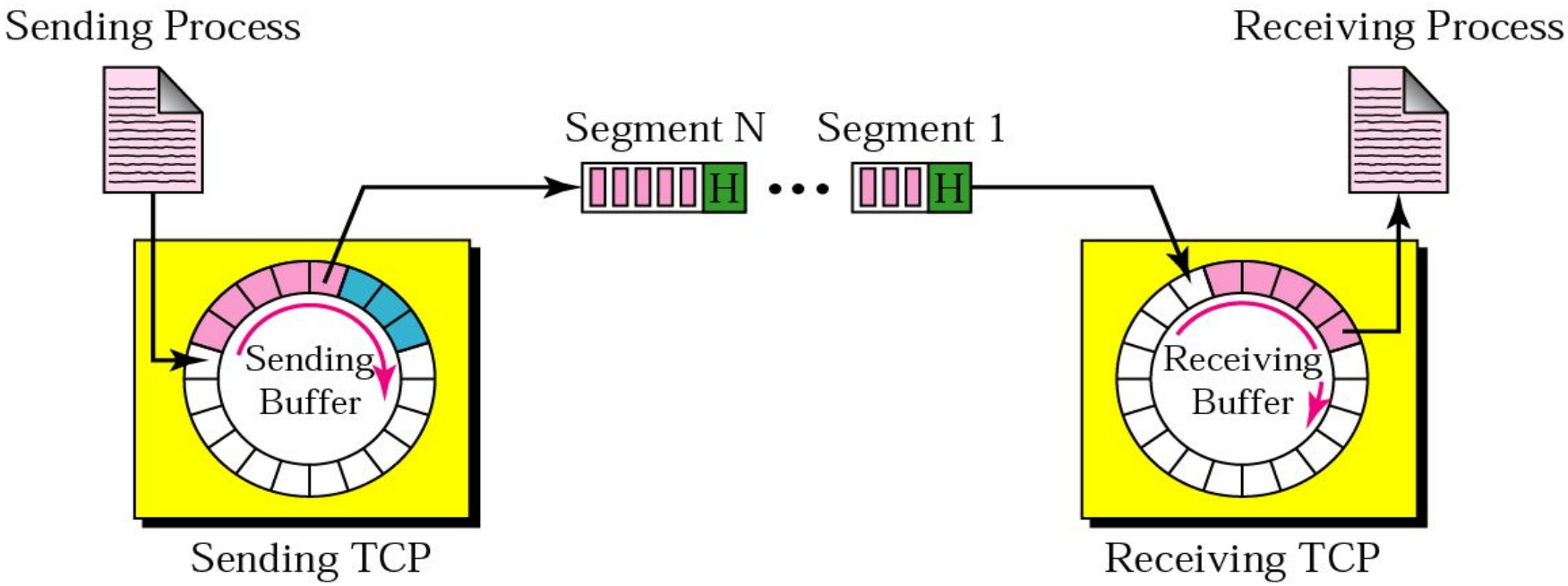


Figure 22.13 TCP segments



Note:

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

The value of the sequence number field in 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.

Figure 22.14 TCP segment format

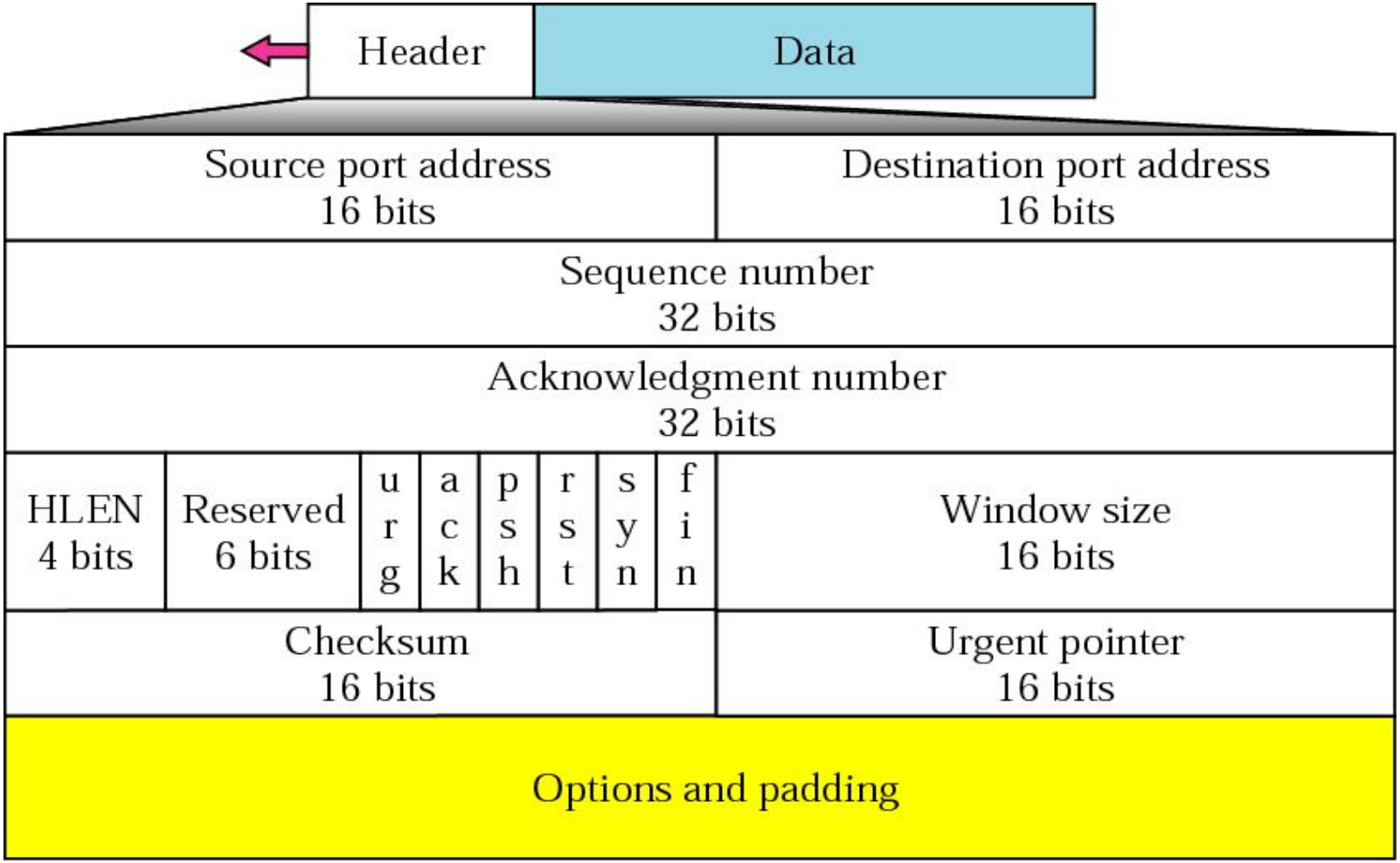


Figure 22.15 Control field

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

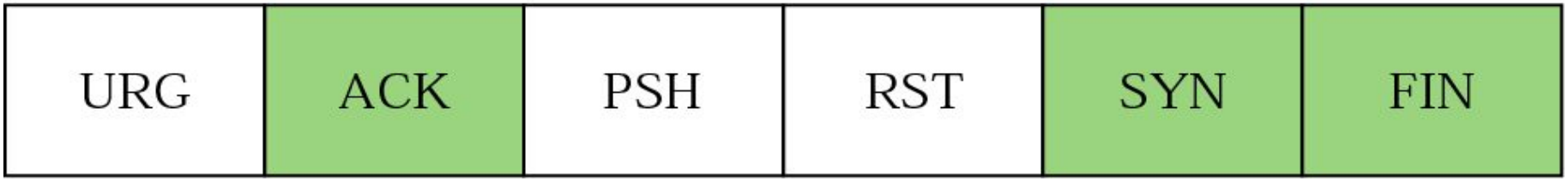


Table 22.3 Description of flags in the control field

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 22.16 Three-step connection establishment

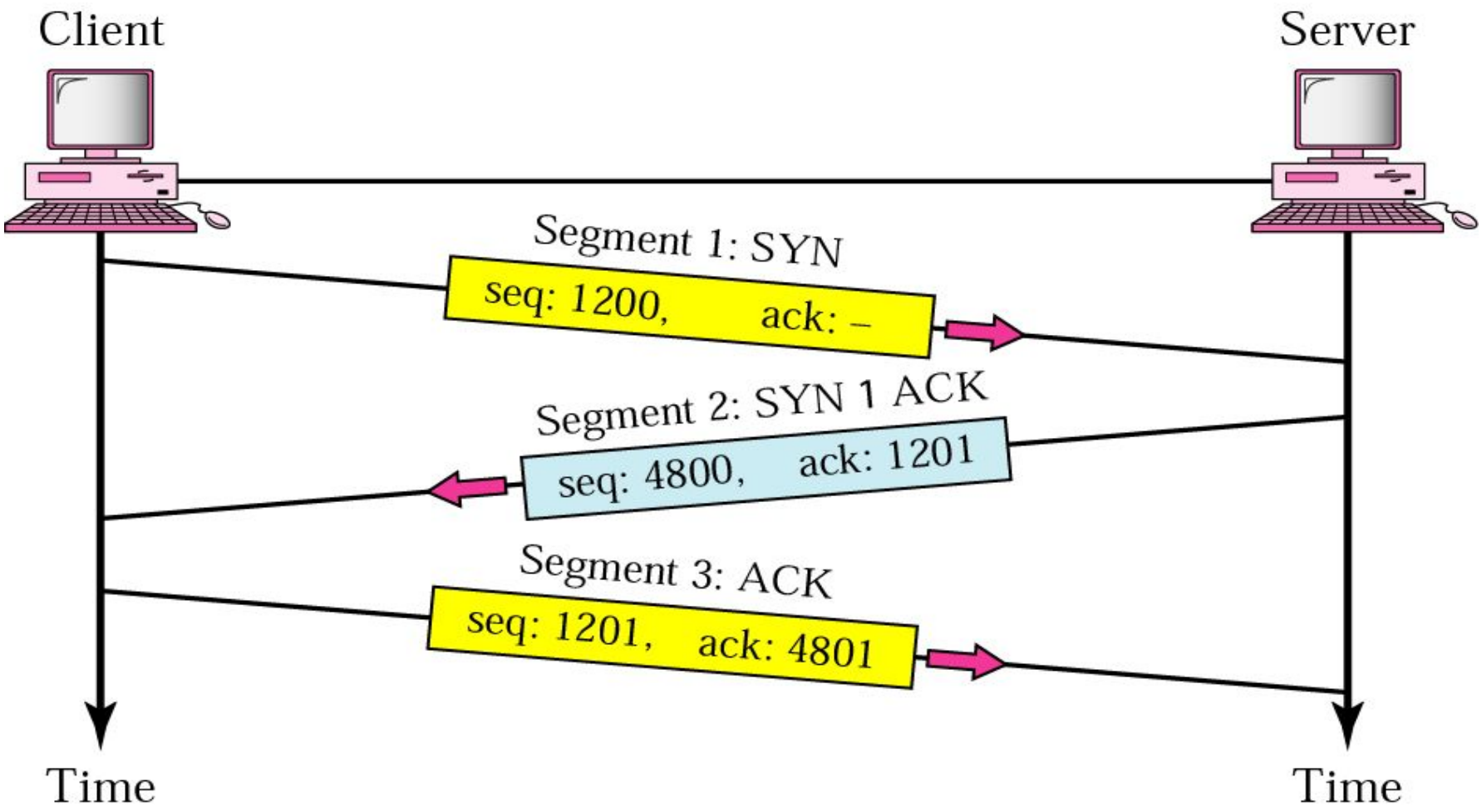


Figure 22.17 Four-step connection termination

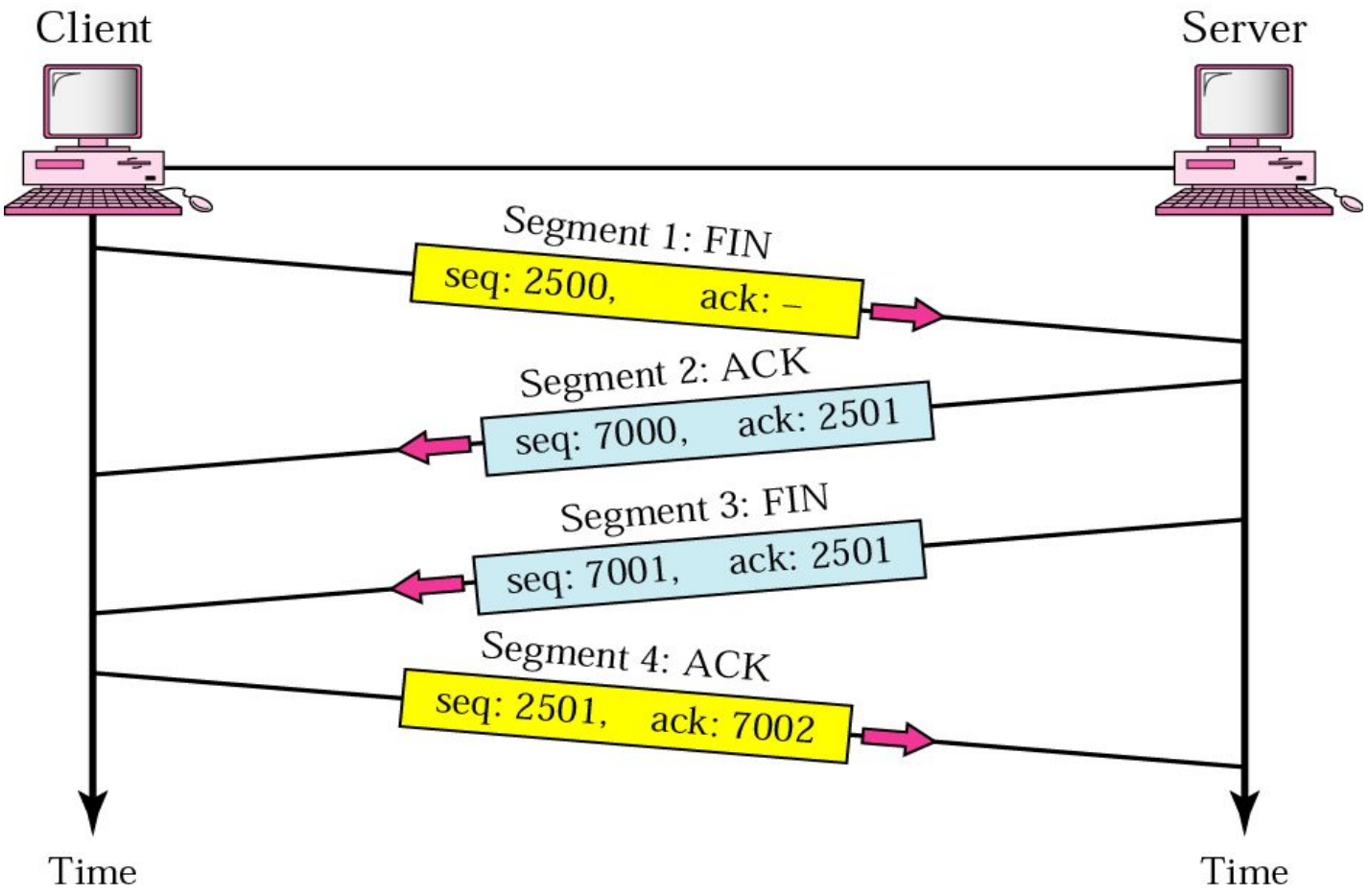


Table 22.4 States for TCP

State	Description
CLOSED	There is no connection.
LISTEN	The server is waiting for calls from the client.
SYN-SENT	A connection request is sent; waiting for acknowledgment.
SYN-RCVD	A connection request is received.
ESTABLISHED	Connection is established.
FIN-WAIT-1	The application has requested the closing of the connection.
FIN-WAIT-2	The other side has accepted the closing of the connection.
TIME-WAIT	Waiting for retransmitted segments to die.
CLOSE-WAIT	The server is waiting for the application to close.
LAST-ACK	The server is waiting for the last acknowledgment.

Congestion Control and Quality of Service

23.1 Data Traffic

Traffic Descriptor

Traffic Profiles

Figure 23.1 Traffic descriptors

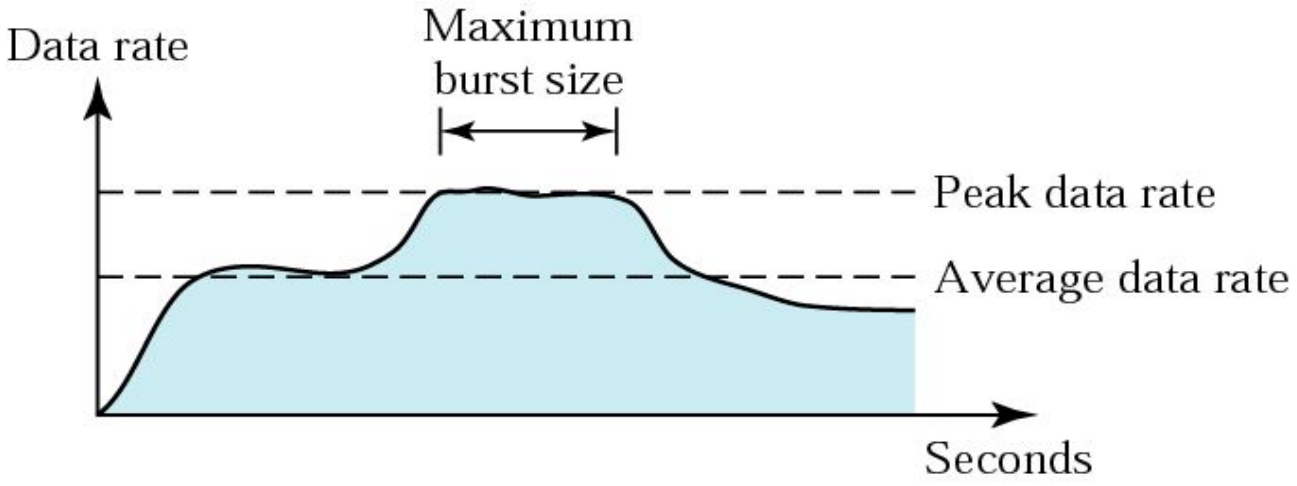


Figure 23.2 Constant-bit-rate traffic

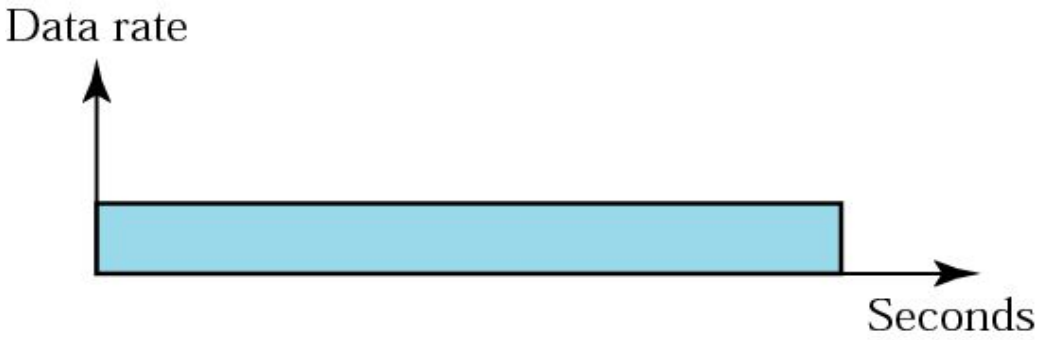


Figure 23.3 Variable-bit-rate traffic

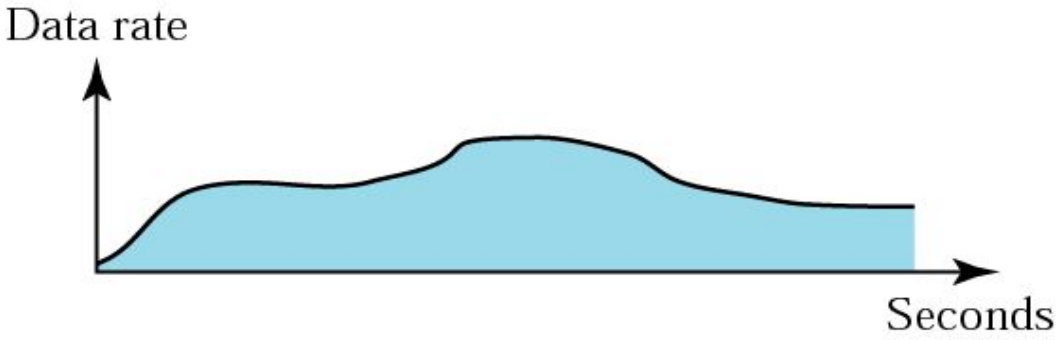
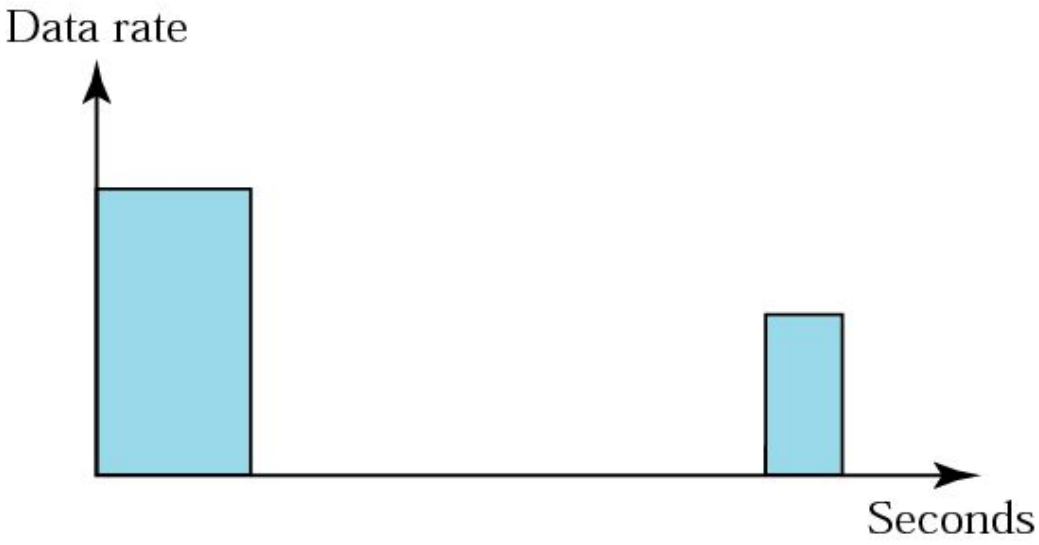


Figure 23.4 Bursty traffic



23.2 Congestion

Network Performance

Figure 23.5 Incoming packet

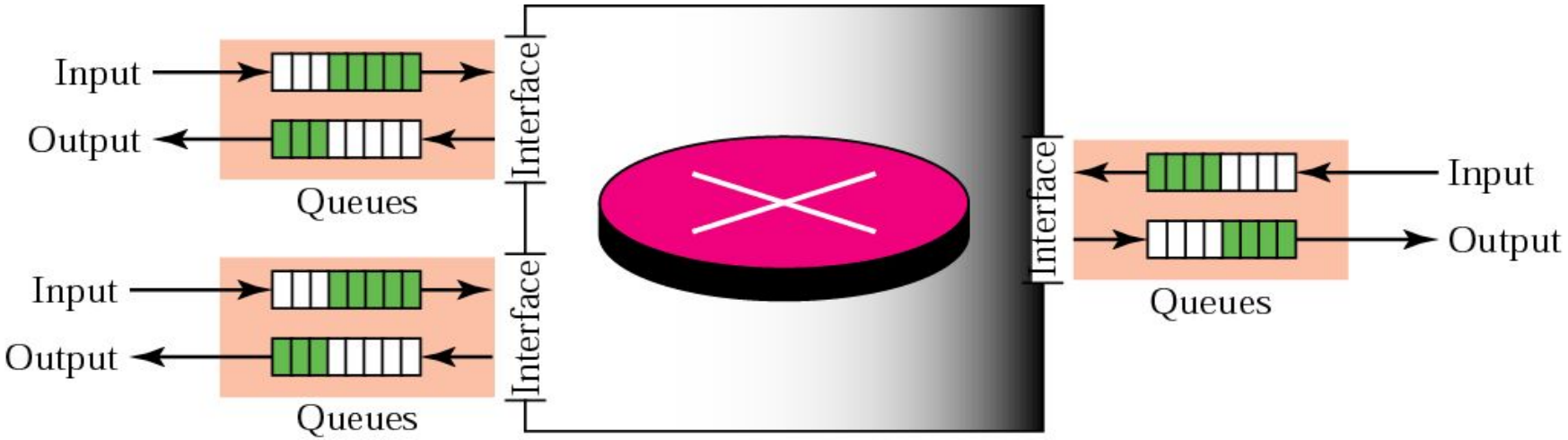
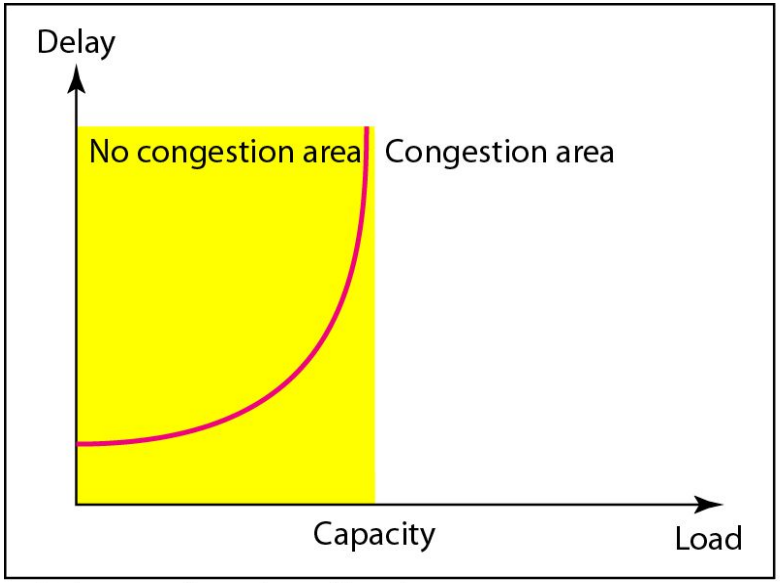
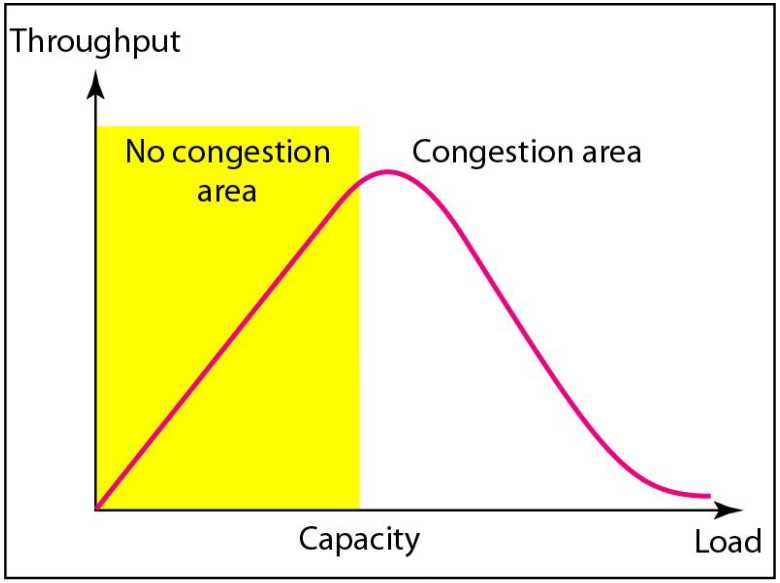


Figure *Packet delay and throughput as functions of load*



a. Delay as a function of load



b. Throughput as a function of load

24-3 CONGESTION CONTROL

Congestion control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened. In general, we can divide congestion control mechanisms into two broad categories: open-loop congestion control (prevention) and closed-loop congestion control (removal).

Topics discussed in this section:

Open-Loop Congestion Control
Closed-Loop Congestion Control

Figure 24.5 *Congestion control categories*

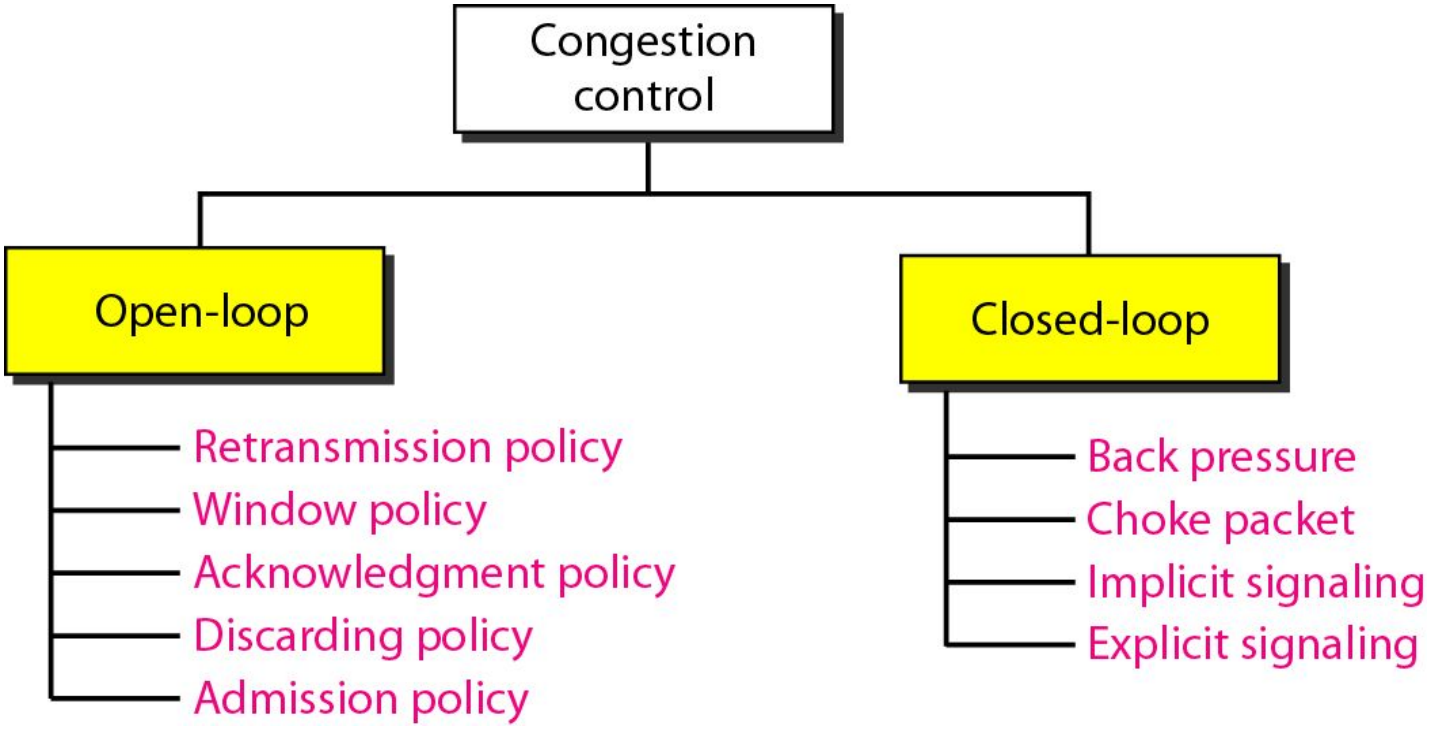


Figure 24.6 *Backpressure method for alleviating congestion*

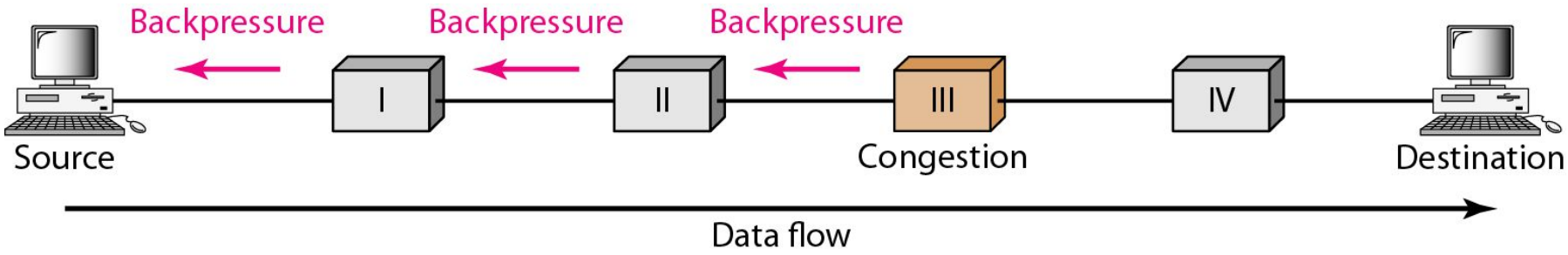
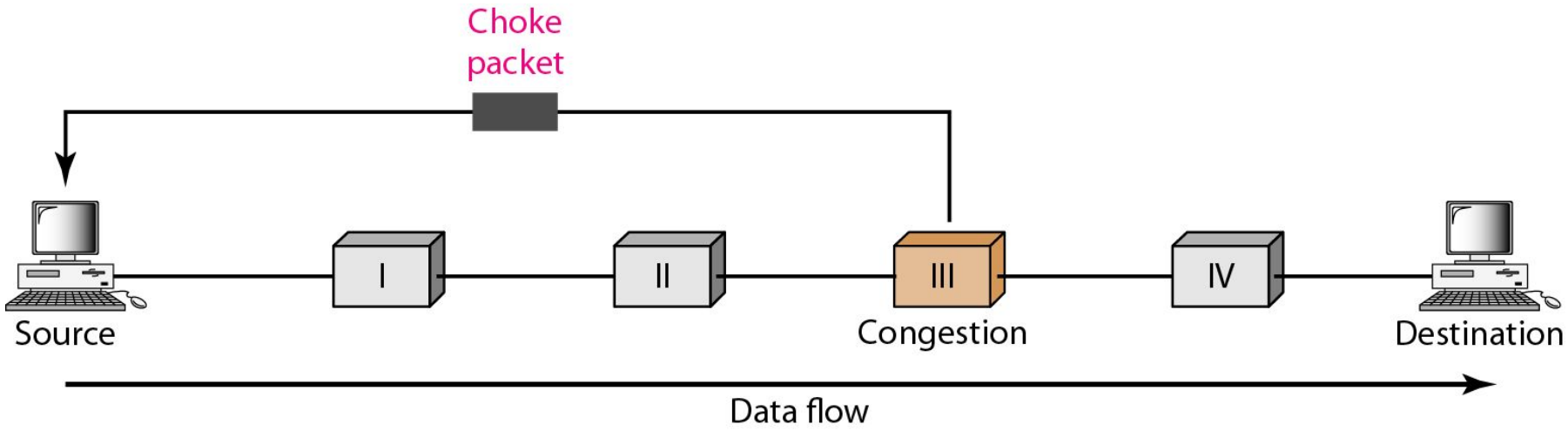


Figure 24.7 *Choke packet*



TCP assumes that the cause of a lost segment is due to congestion in the network.

If the cause of the lost segment is congestion, retransmission of the segment does not remove the cause—it aggravates it.

23.5 Quality of Service

Flow Characteristics

Flow Classes

23.6 Techniques to Improve QoS

Scheduling

Traffic Shaping

Resource Reservation

Admission Control

Figure 23.12 Flow characteristics

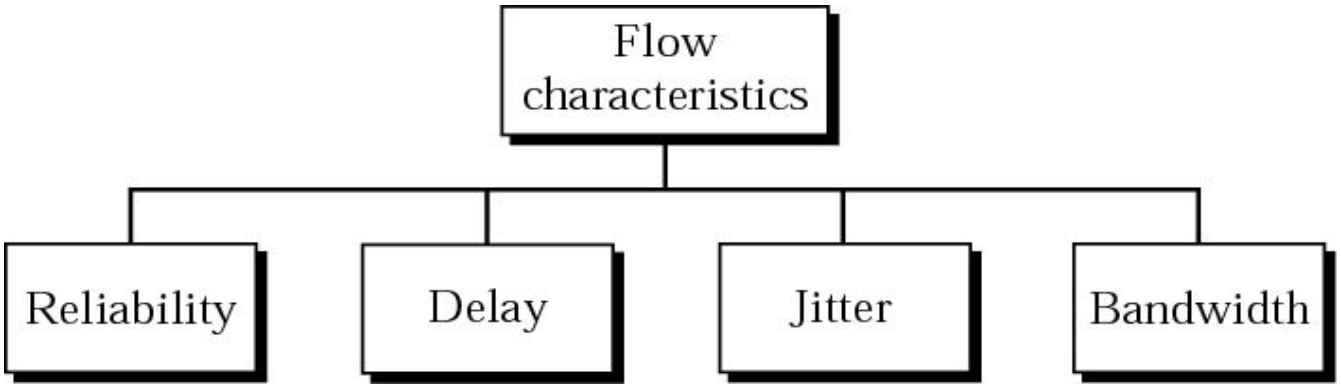


Figure 23.13 FIFO queue

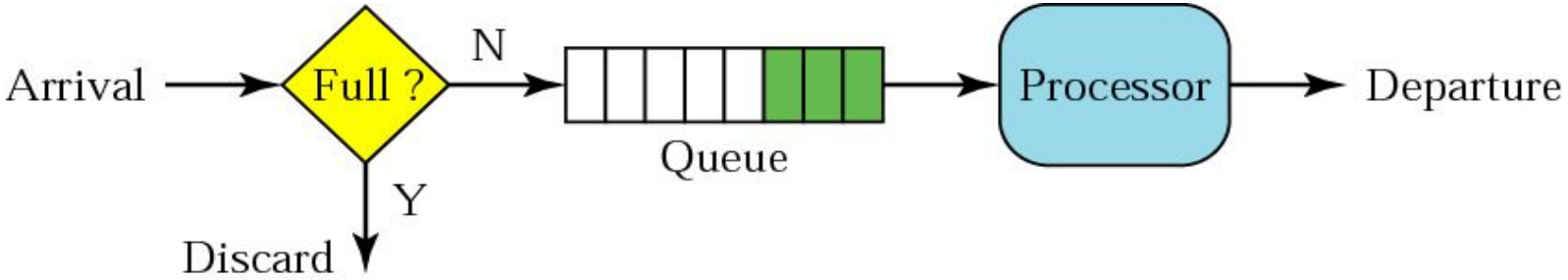


Figure 23.14 Priority queuing

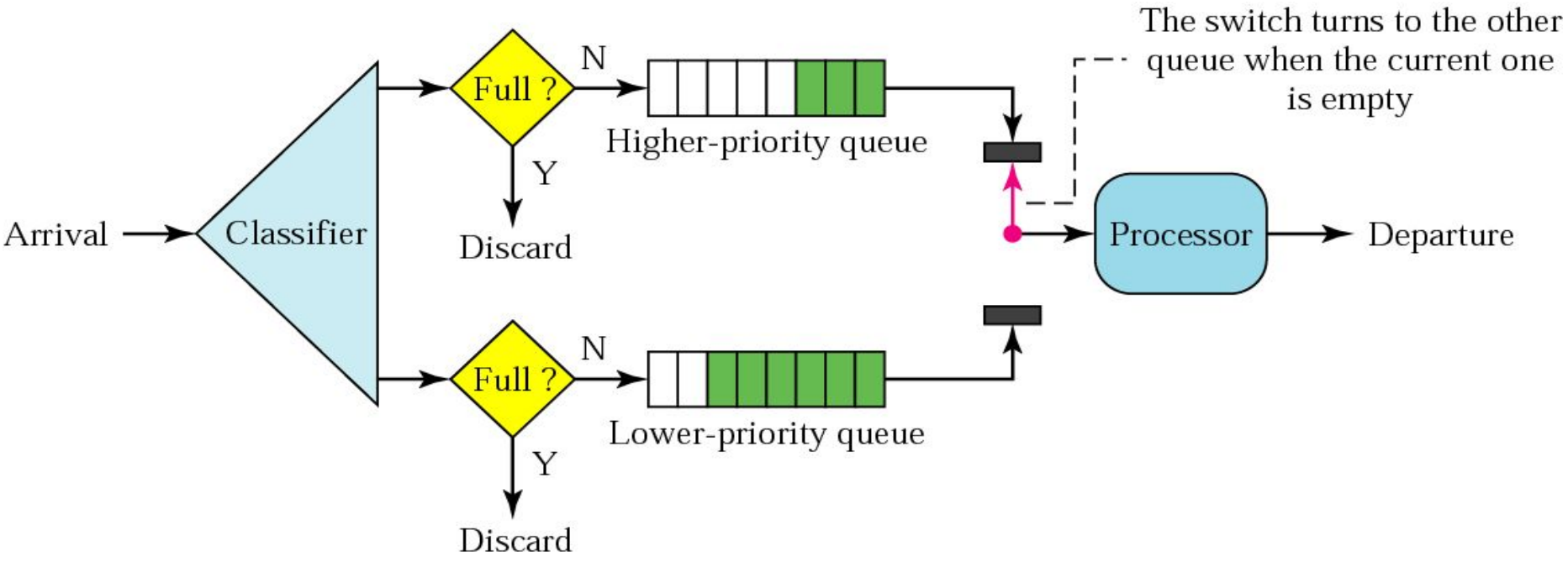


Figure 23.15 Weighted fair queuing

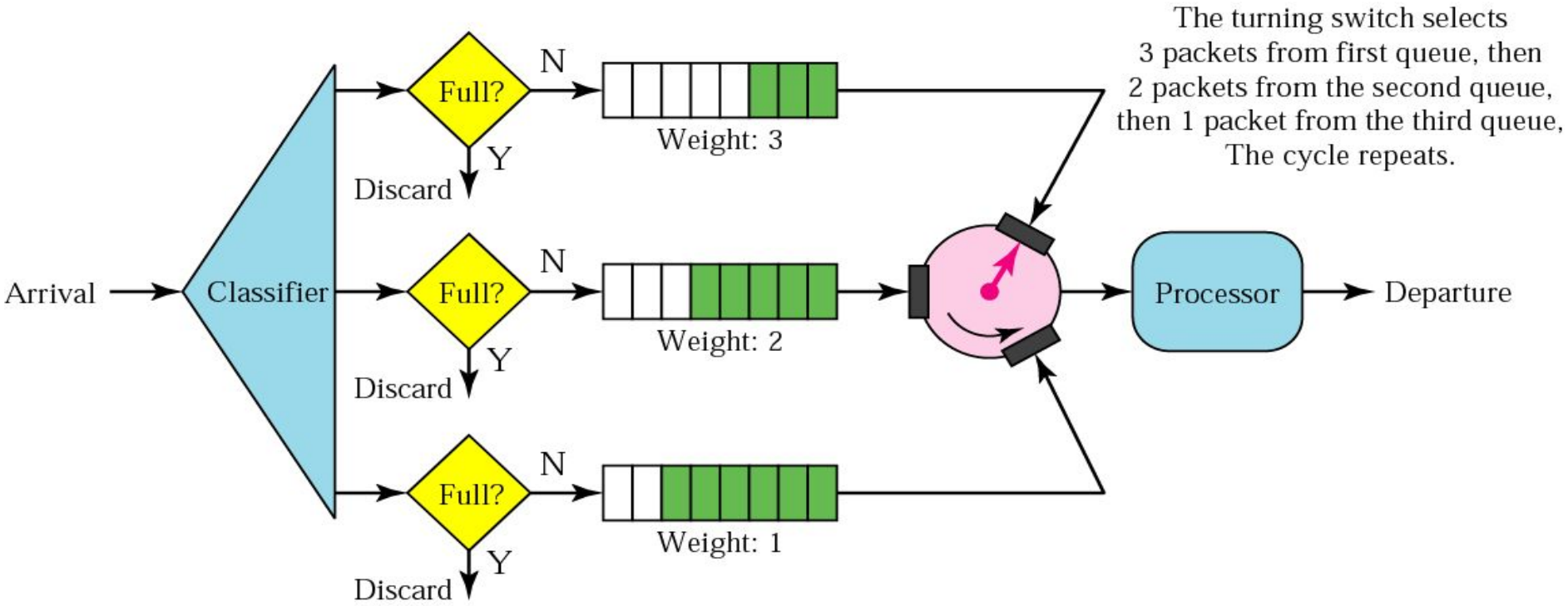
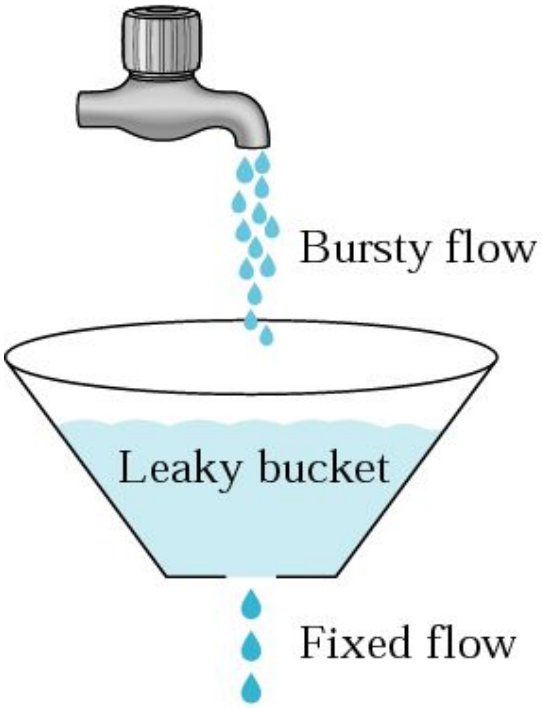
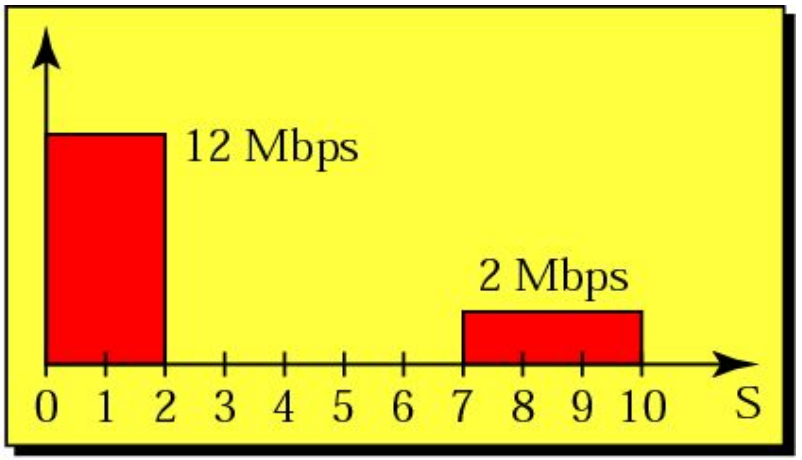
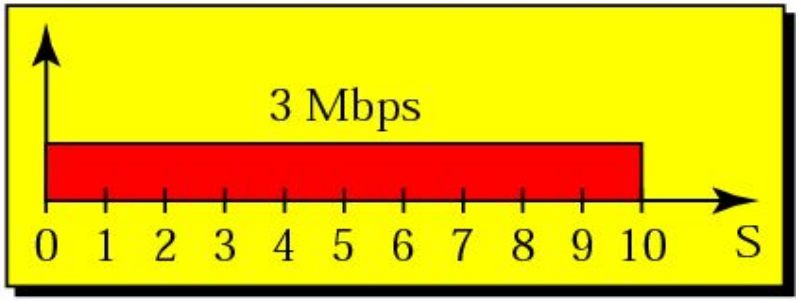


Figure 23.16 Leaky bucket



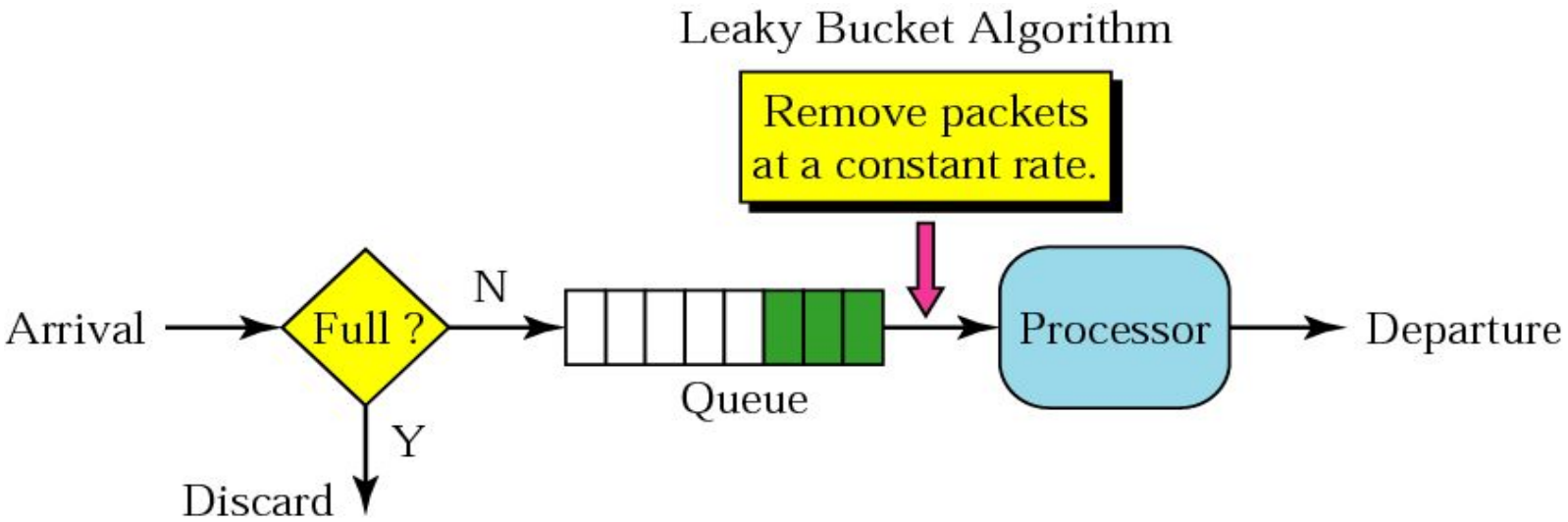


Bursty data



Fixed-rate data

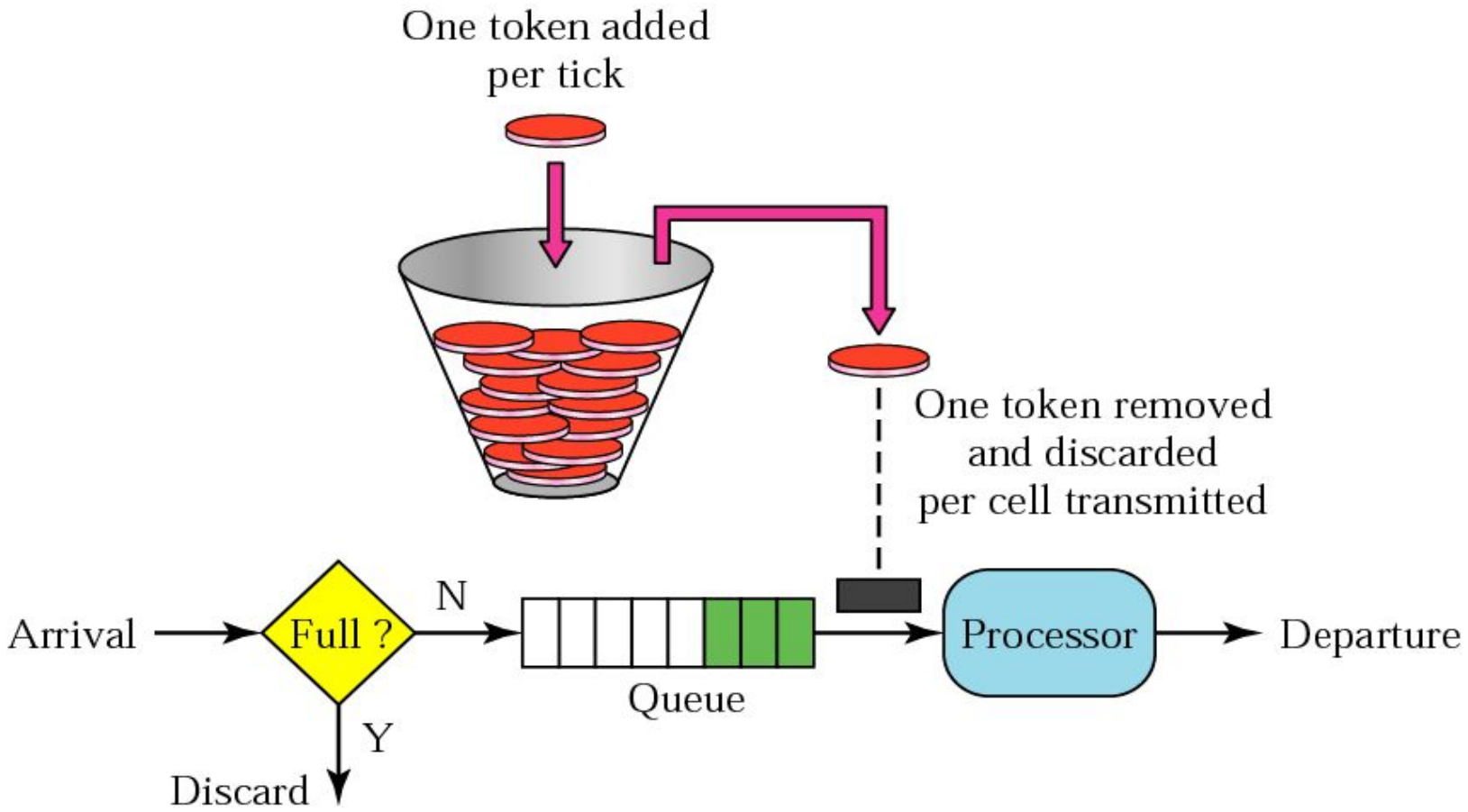
Figure 23.17 Leaky bucket implementation



Note:

A leaky bucket algorithm shapes bursty traffic into fixed-rate traffic by averaging the data rate. It may drop the packets if the bucket is full.

Figure 23.18 Token bucket



Note:

The token bucket allows bursty traffic at a regulated maximum rate.