Data Communications and Networking Fourth Edition

Forouzan

Chapter 24

Congestion Control and Quality of Service

24-1 DATA TRAFFIC

The main focus of congestion control and quality of service is data traffic. In congestion control we try to avoid traffic congestion. In quality of service, we try to create an appropriate environment for the traffic. So, before talking about congestion control and quality of service, we discuss the data traffic itself.

Topics discussed in this section:

Traffic Descriptor
Traffic Profiles

Figure 24.1 *Traffic descriptors*

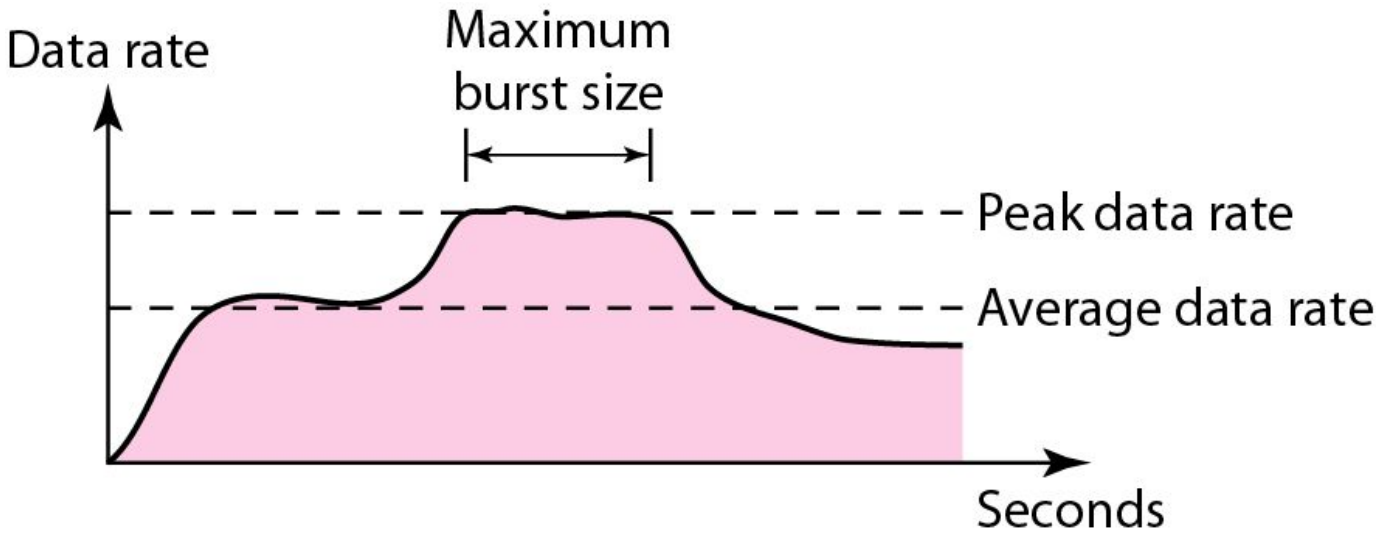
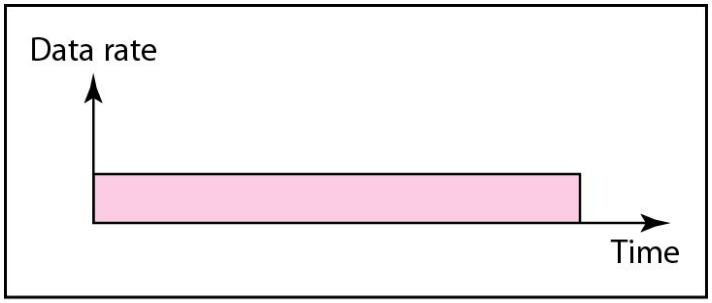
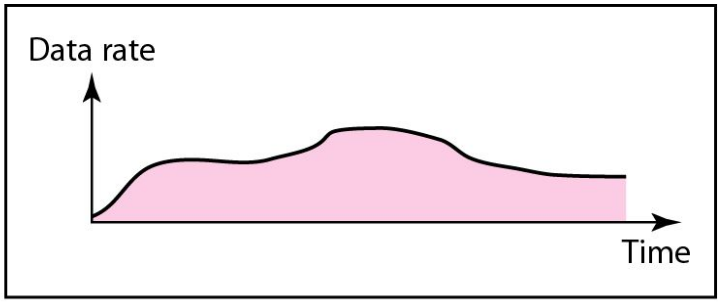


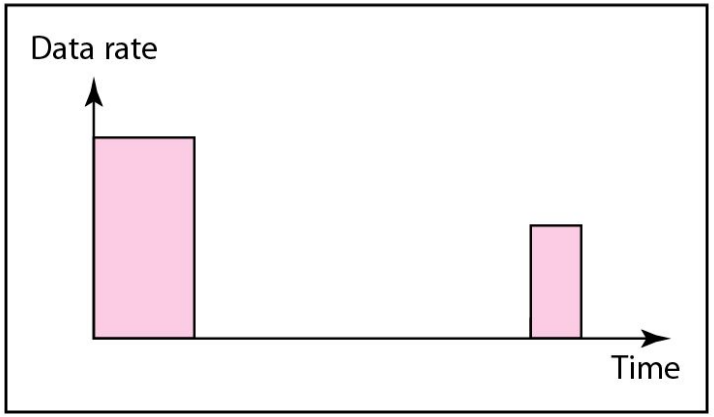
Figure 24.2 *Three traffic profiles*



a. Constant bit rate



b. Variable bit rate



c. Bursty

24-2 CONGESTION

Congestion in a network may occur if the load on the network—the number of packets sent to the network—is greater than the capacity of the network—the number of packets a network can handle. Congestion control refers to the mechanisms and techniques to control the congestion and keep the load below the capacity.

Topics discussed in this section:

Network Performance

Congestion

- Occurs when no. of packet sent to n/w is greater than capacity of n/w.
- Due to congestion – performance decrea

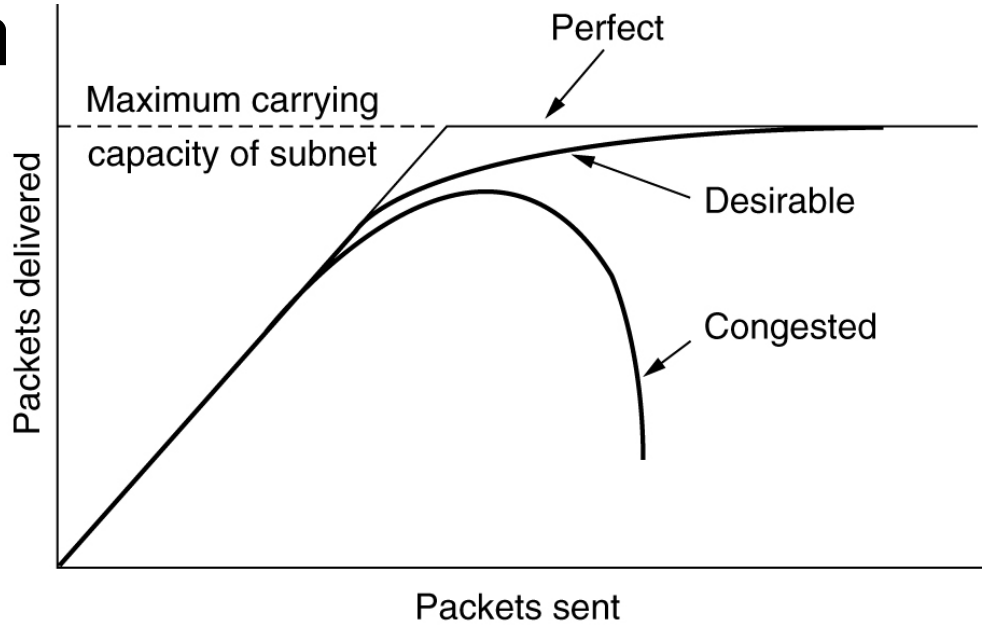


Figure 24.3 *Queues in a router*

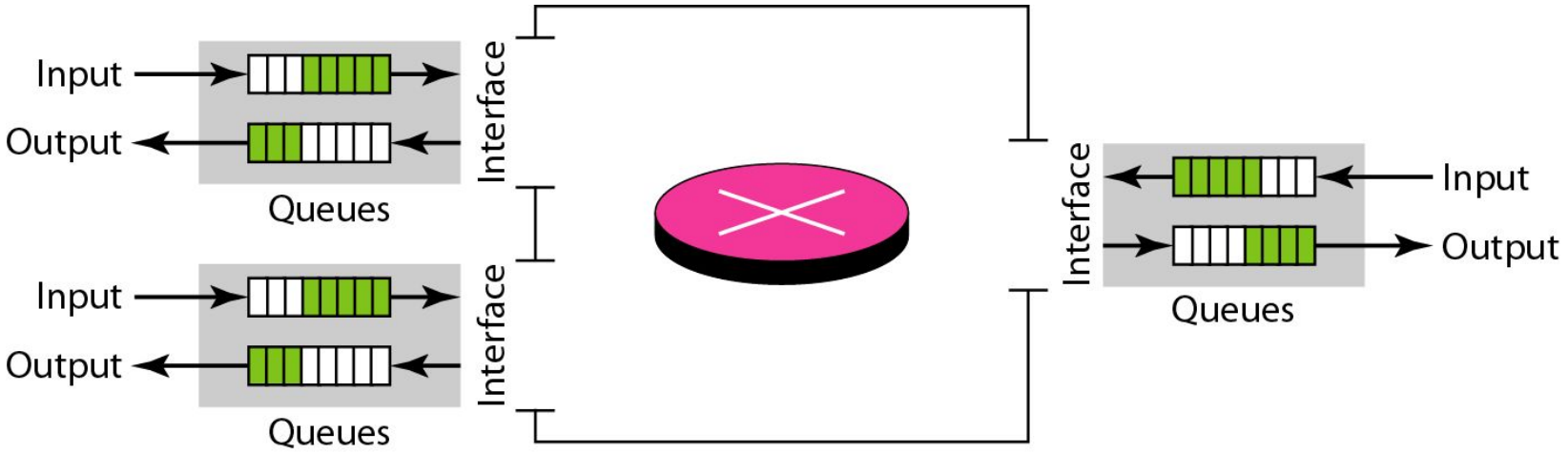
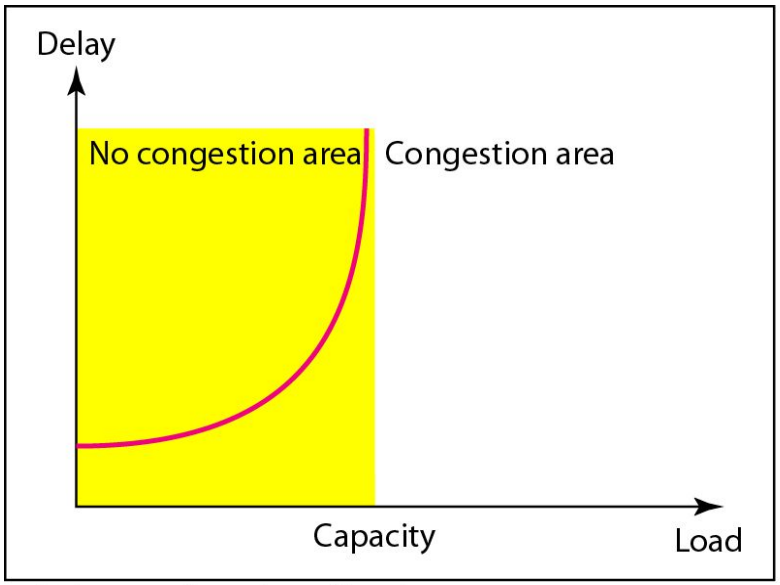
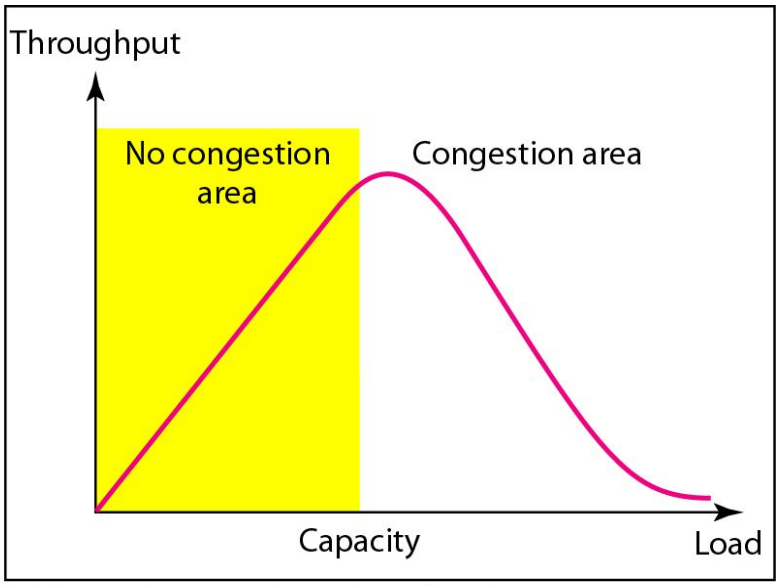


Figure *Packet delay and throughput as functions of load*



a. Delay as a function of load

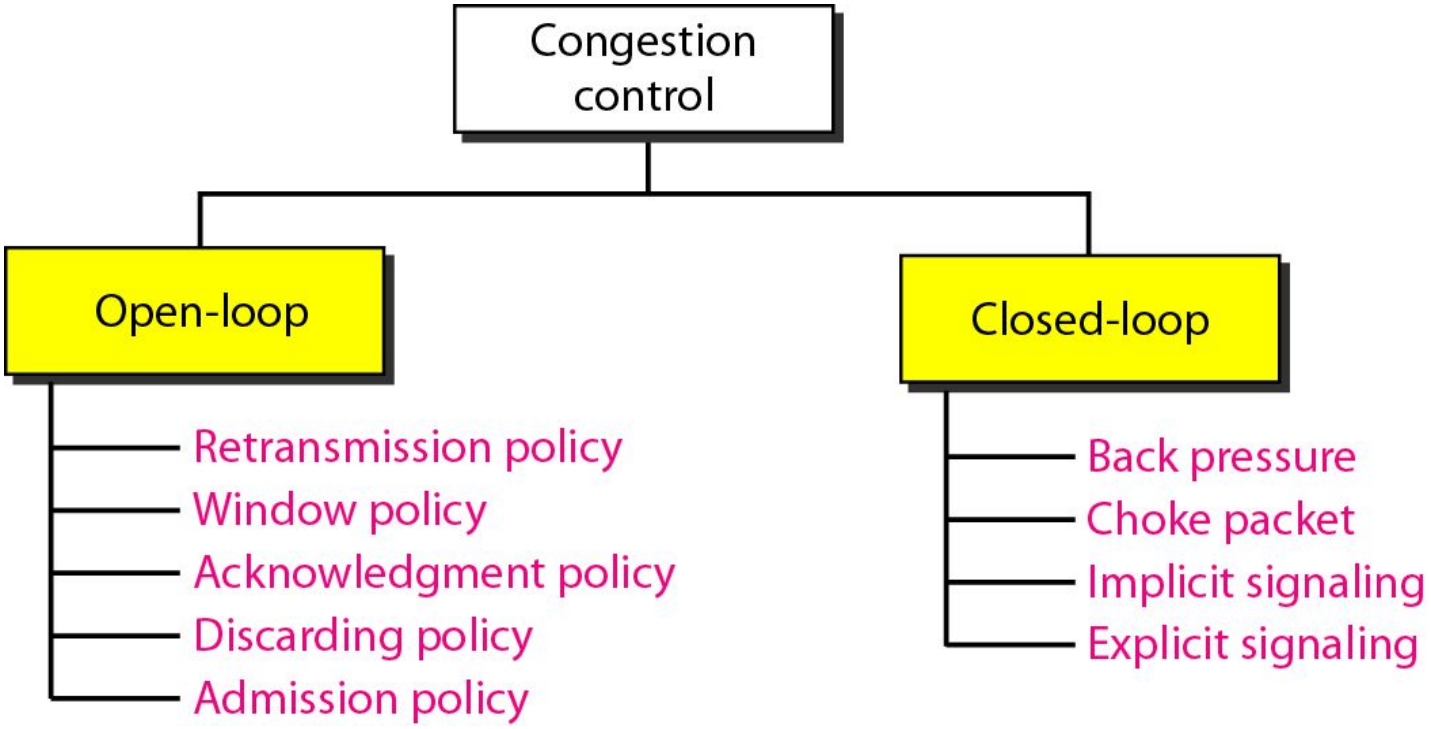


b. Throughput as a function of load

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).

Figure 24.5 *Congestion control categories*



Congestion Control

- Open Loop – policies to prevent congestion
  - **Retransmission Policy** (may also increase congestion)
  - **Window Policy** (used in Go Back-N & Selective repeat)
  - **Acknowledgement Policy** (applied by receiver-less ACK → less load, may acknowledge when N packets are received)
  - **Discarding Policy** (less sensitive packets may be dropped)
- Closed Loop – alleviate the congestion after it happens
  - **Back Pressure** (In VC only, info of congestion in reverse direction, all node from that node to source – slow down)
  - **Choke Packet** (info. in reverse direction but only source is affected)
  - **Implicit Signaling** (delay in receiving ACK is treated as congestion – source slow down)
  - **Explicit Signaling** (used in Frame Relay, no separate packet is used (as in back pressure & choke packet), signals are included in the packet itself, Two type of signals = BECN-slowdown packet, FECN-slowdown ACK)

Figure 24.6 *Backpressure method for alleviating congestion*

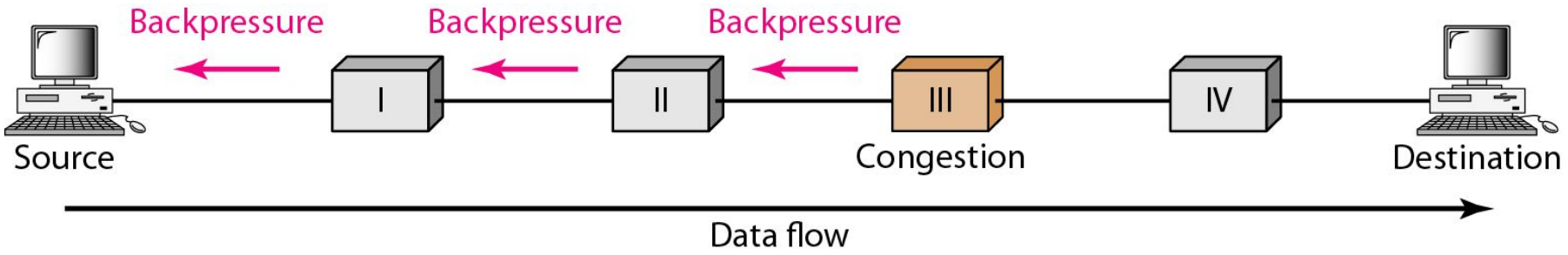
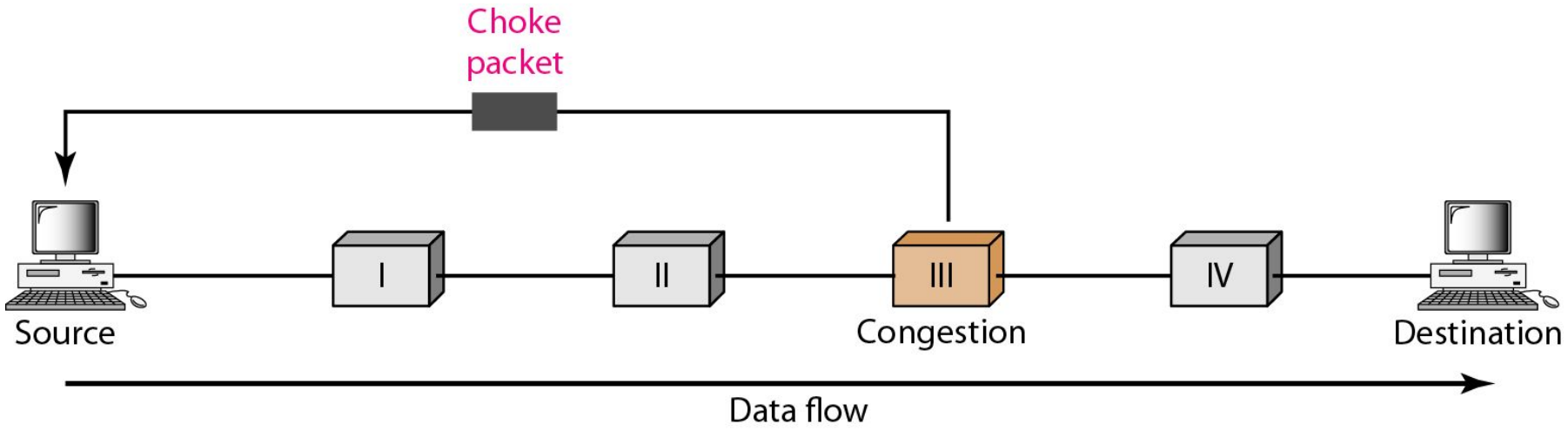
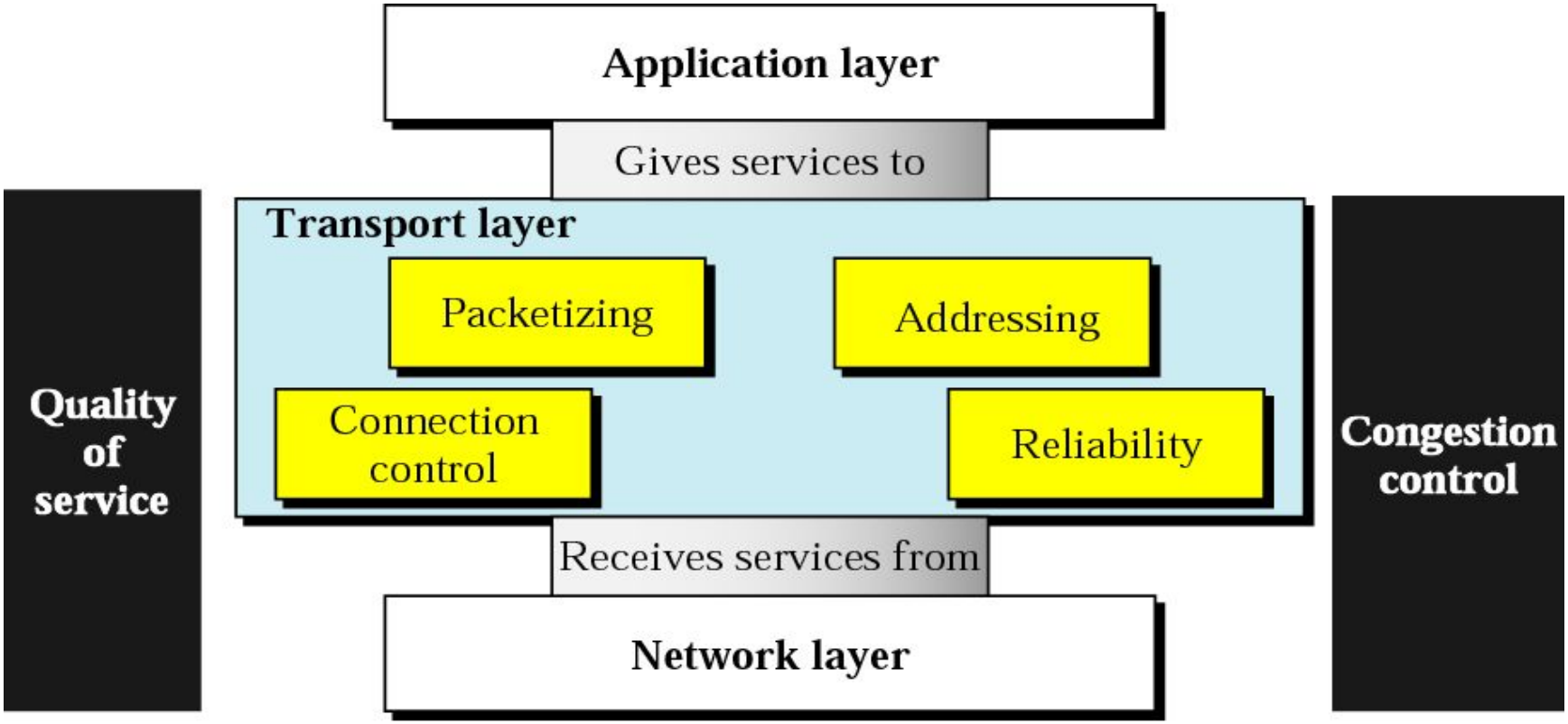


Figure 24.7 *Choke packet*



Position of transport layer



TRANSPORT LAYER

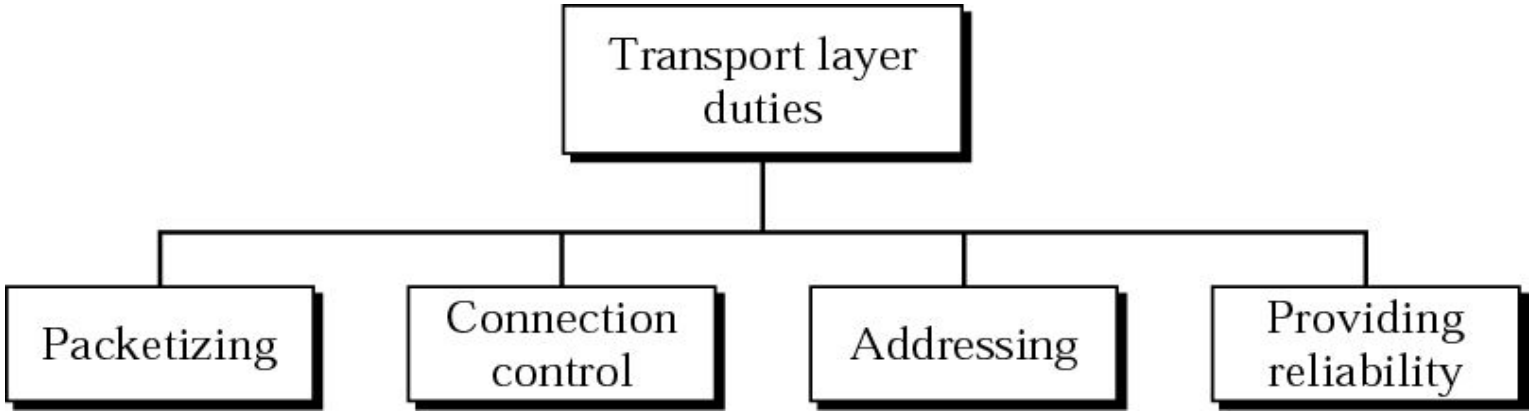
Data unit of this layer is known as **Segment**

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

The Transport layer makes sure data are sent and received in the correct sequence.

Transport layer provides connection services for the protocols and applications. Connection-oriented services or Connectionless services.

Transport layer duties



22.1 Process-to-Process Delivery

Client-Server Paradigm

Addressing

Multiplexing and Demultiplexing

Connectionless/Connection-Oriented

Reliable/Unreliable

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

Function of Transport layer are

Process-Level Addressing: (Port Addressing) Computer run several process (programs) at the Same time and transport layer is responsible for the delivery of a message from one process To another process. To identify process transport layer assign port address at this layer Example of port addresses are SMTP (25) , POP3 (110).

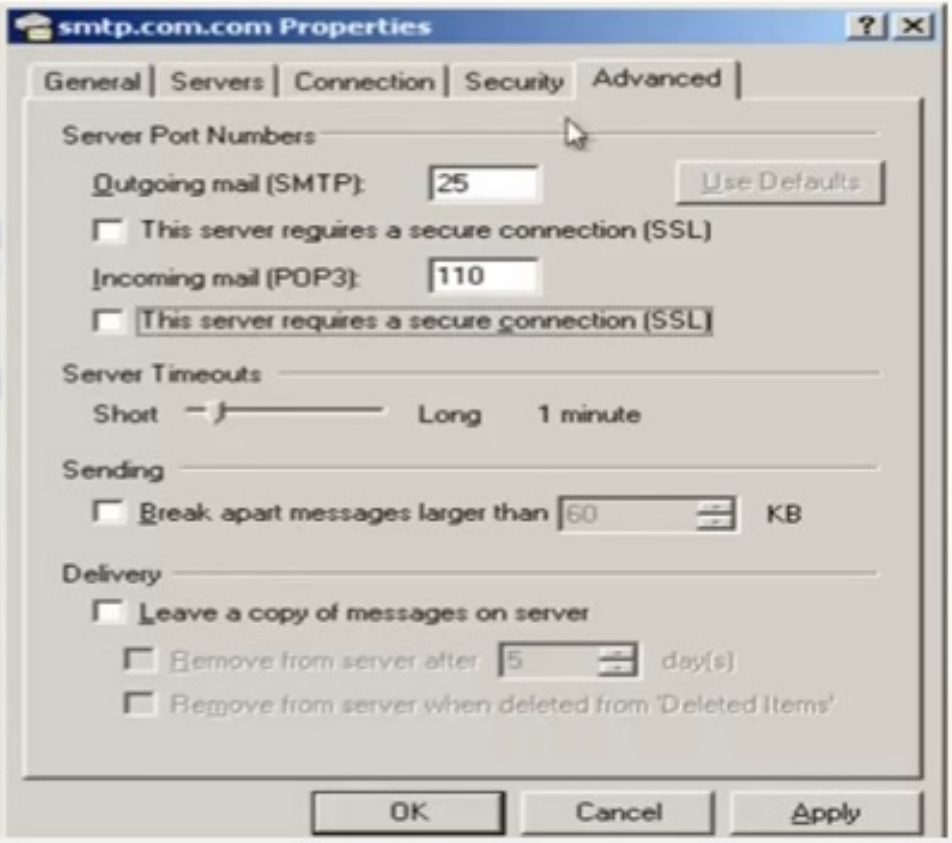


Figure 22.1 Types of data deliveries

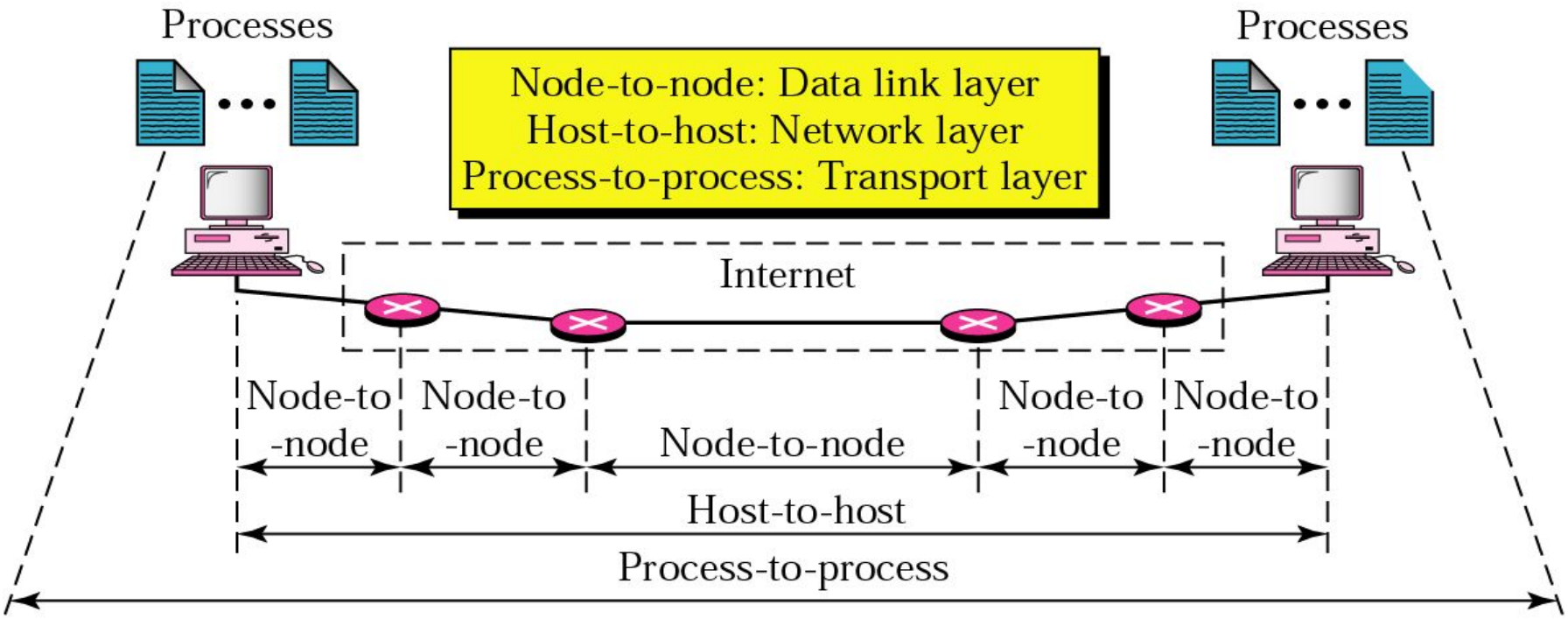


Figure 22.2 Port numbers

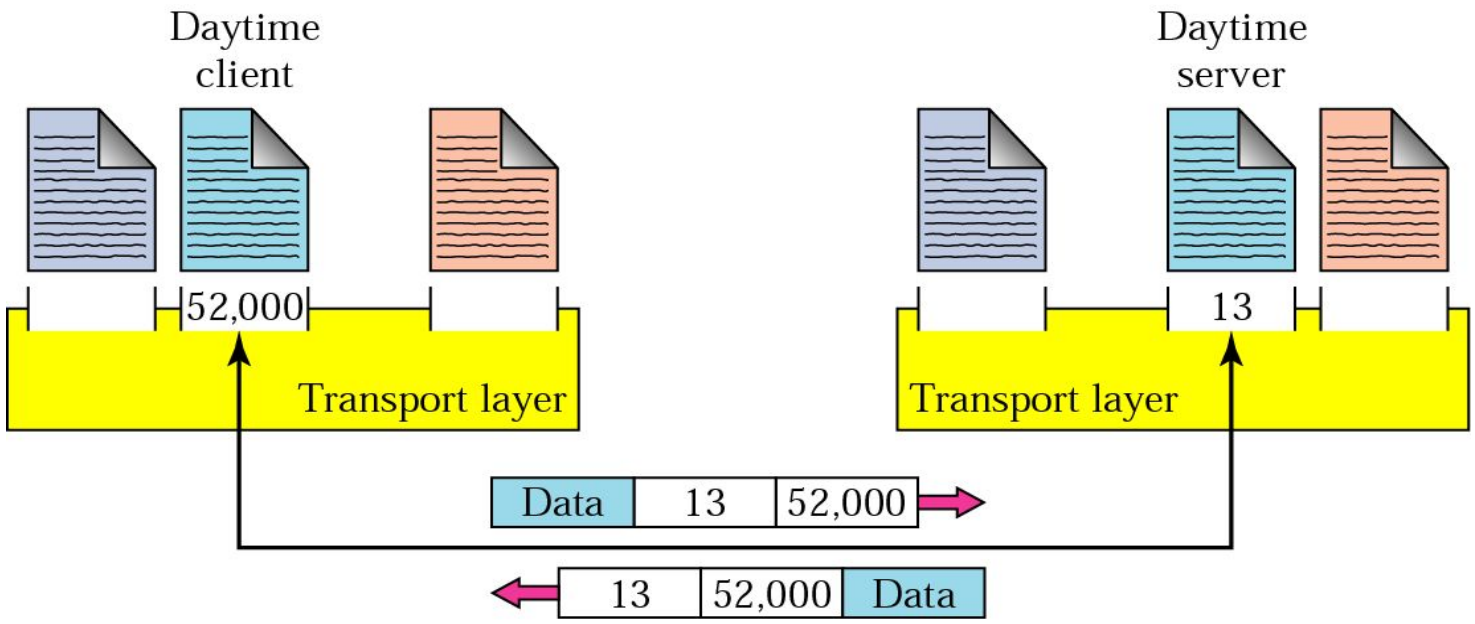


Figure 22.3 IP addresses versus port numbers

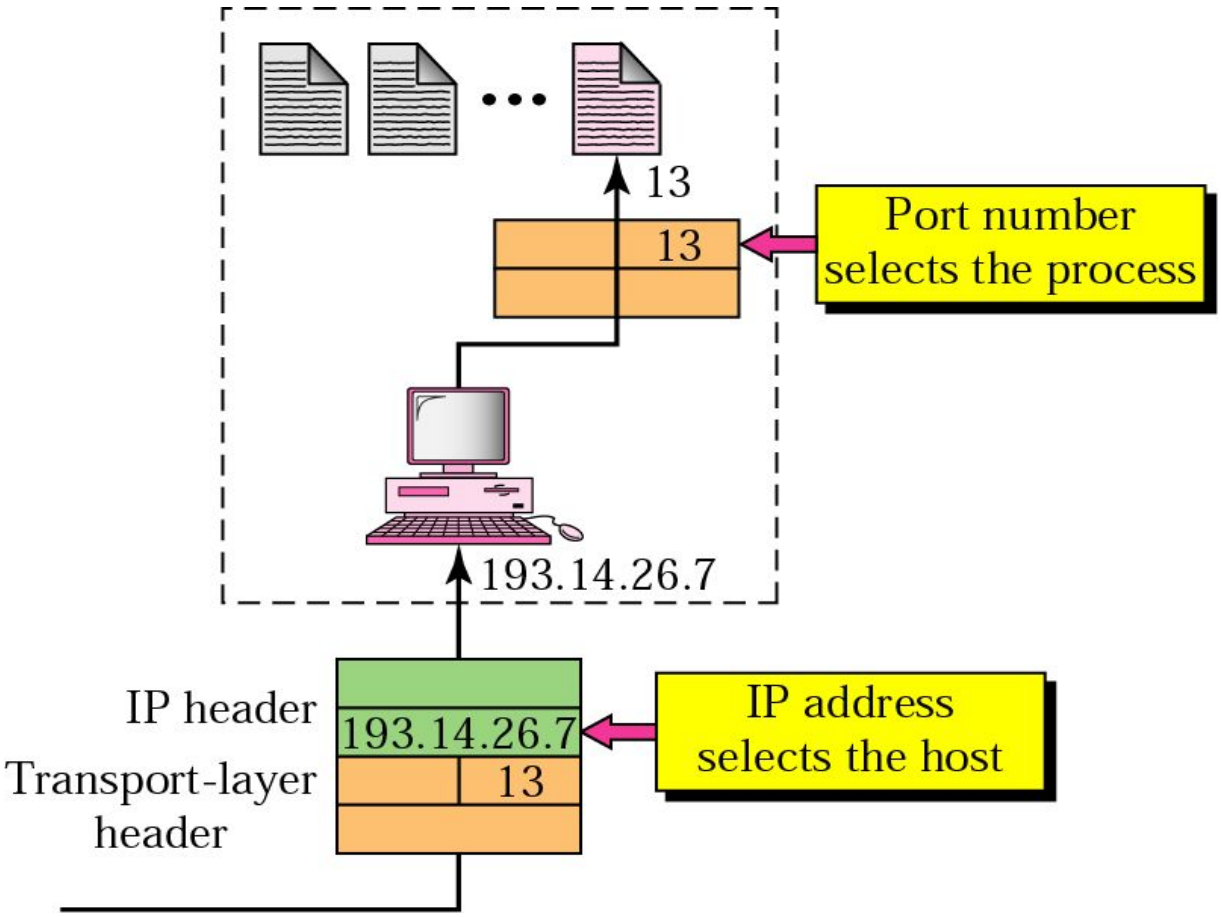


Figure 22.4 IANA ranges

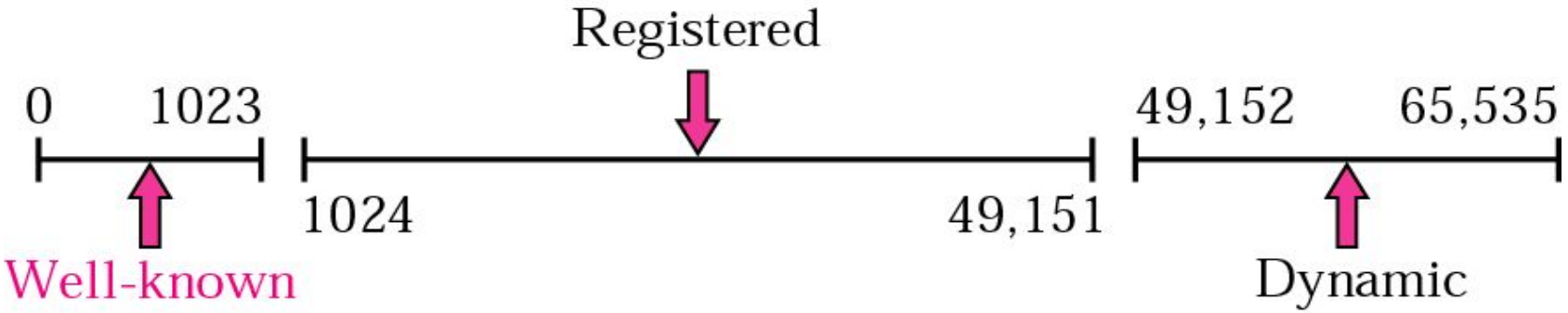


Figure 22.5 Socket address

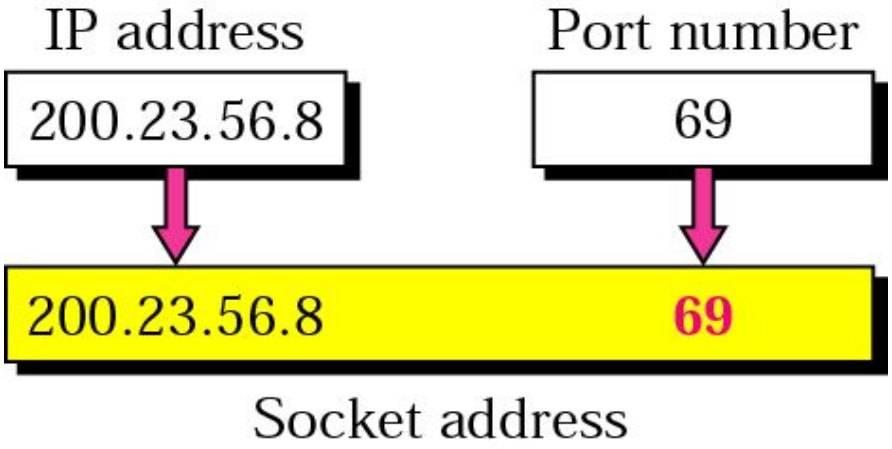


Figure 22.6 Multiplexing and demultiplexing

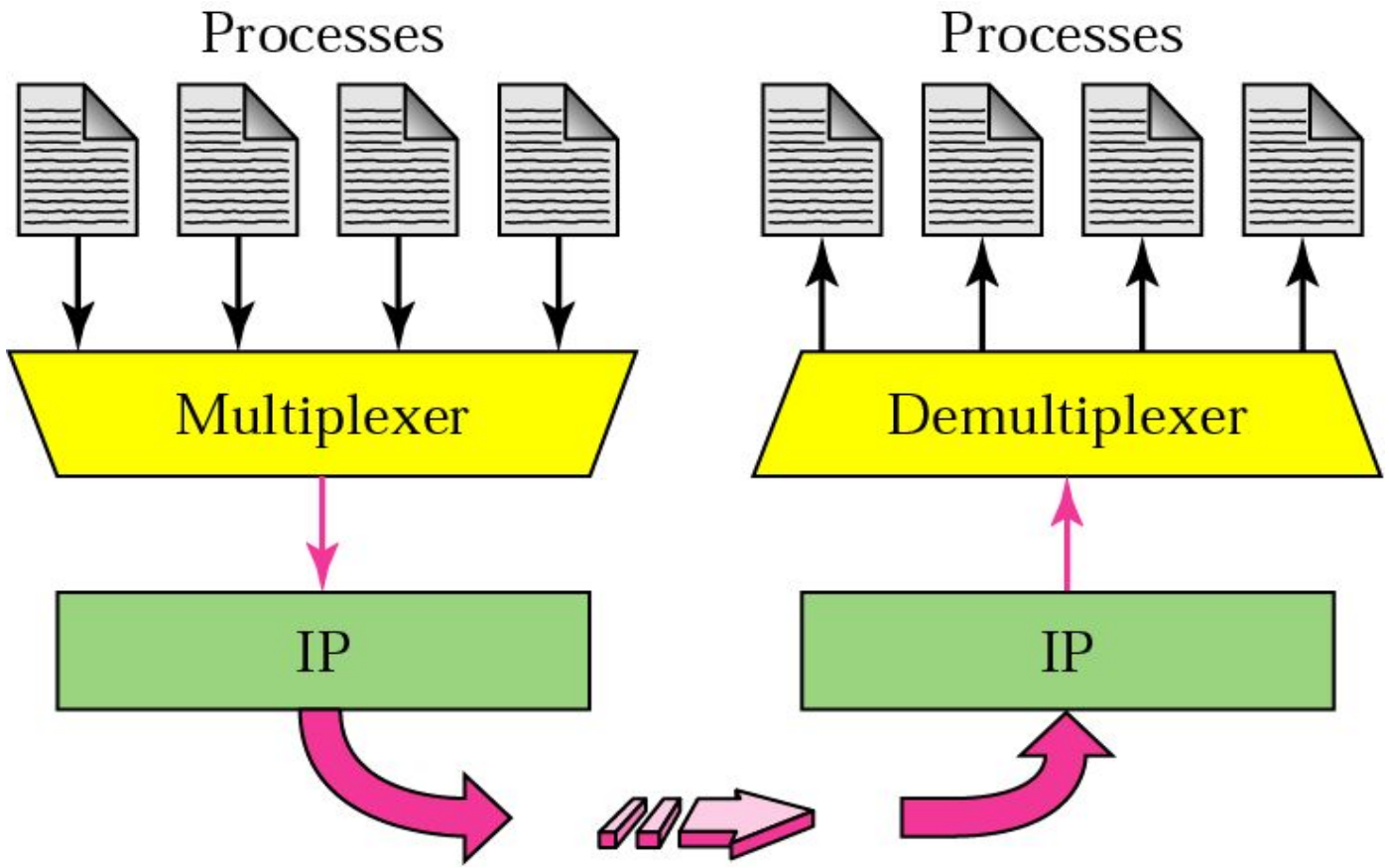
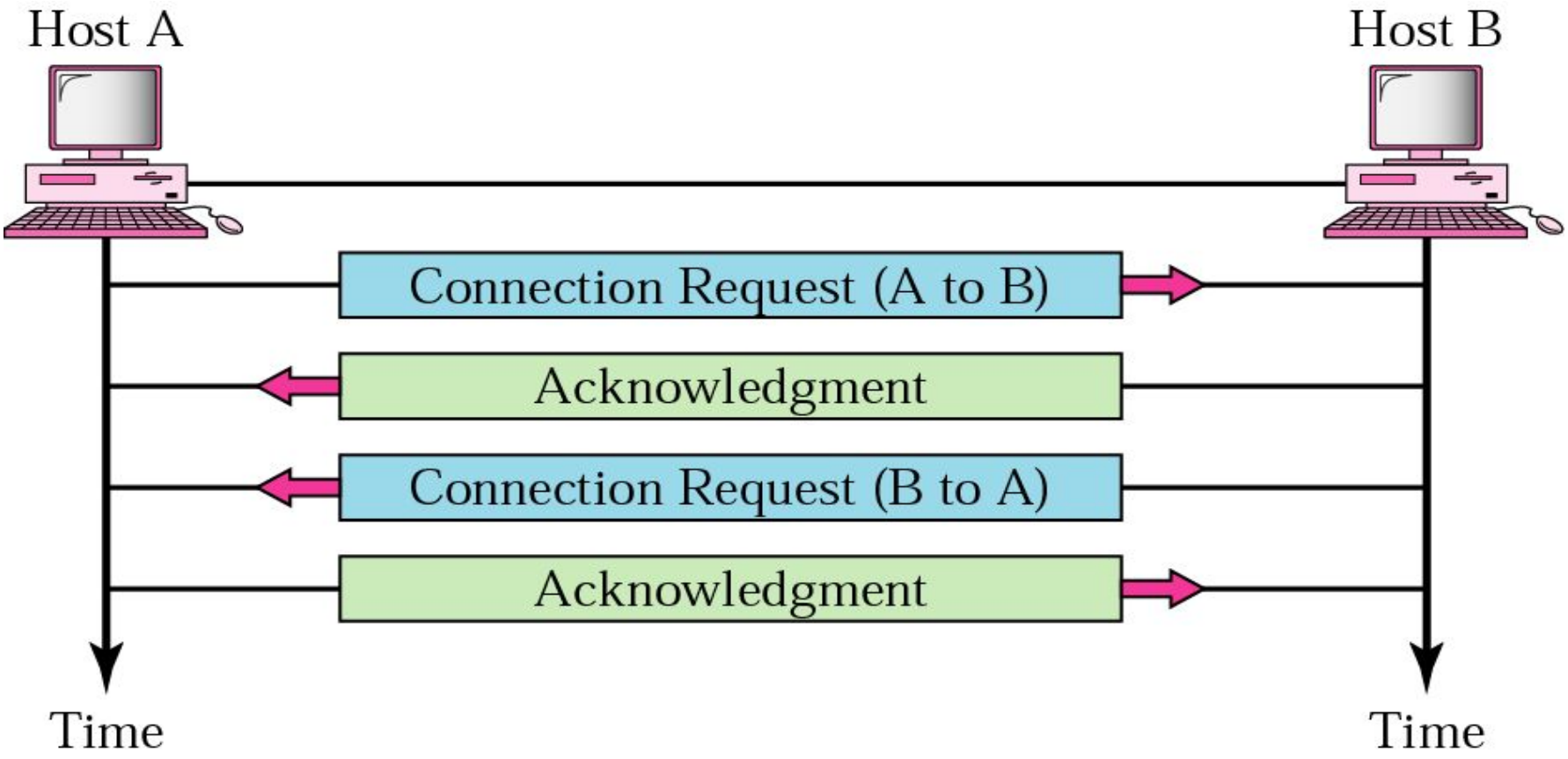


Figure 22.7 Connection establishment



TCP Three-way Handshake

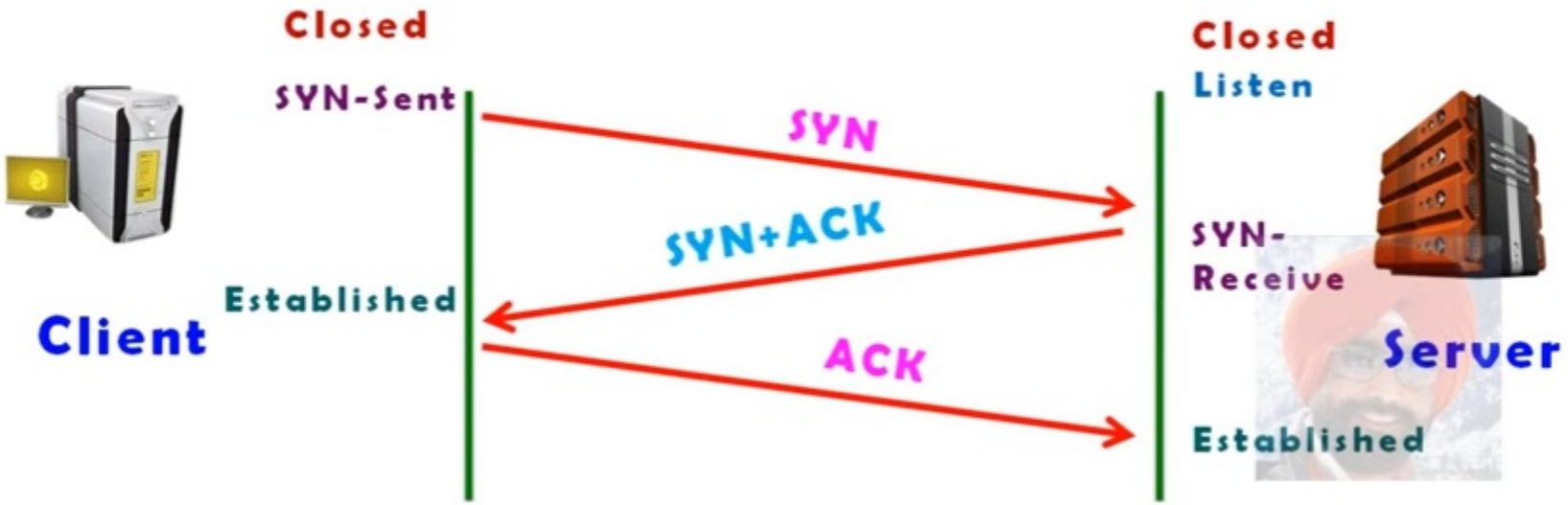
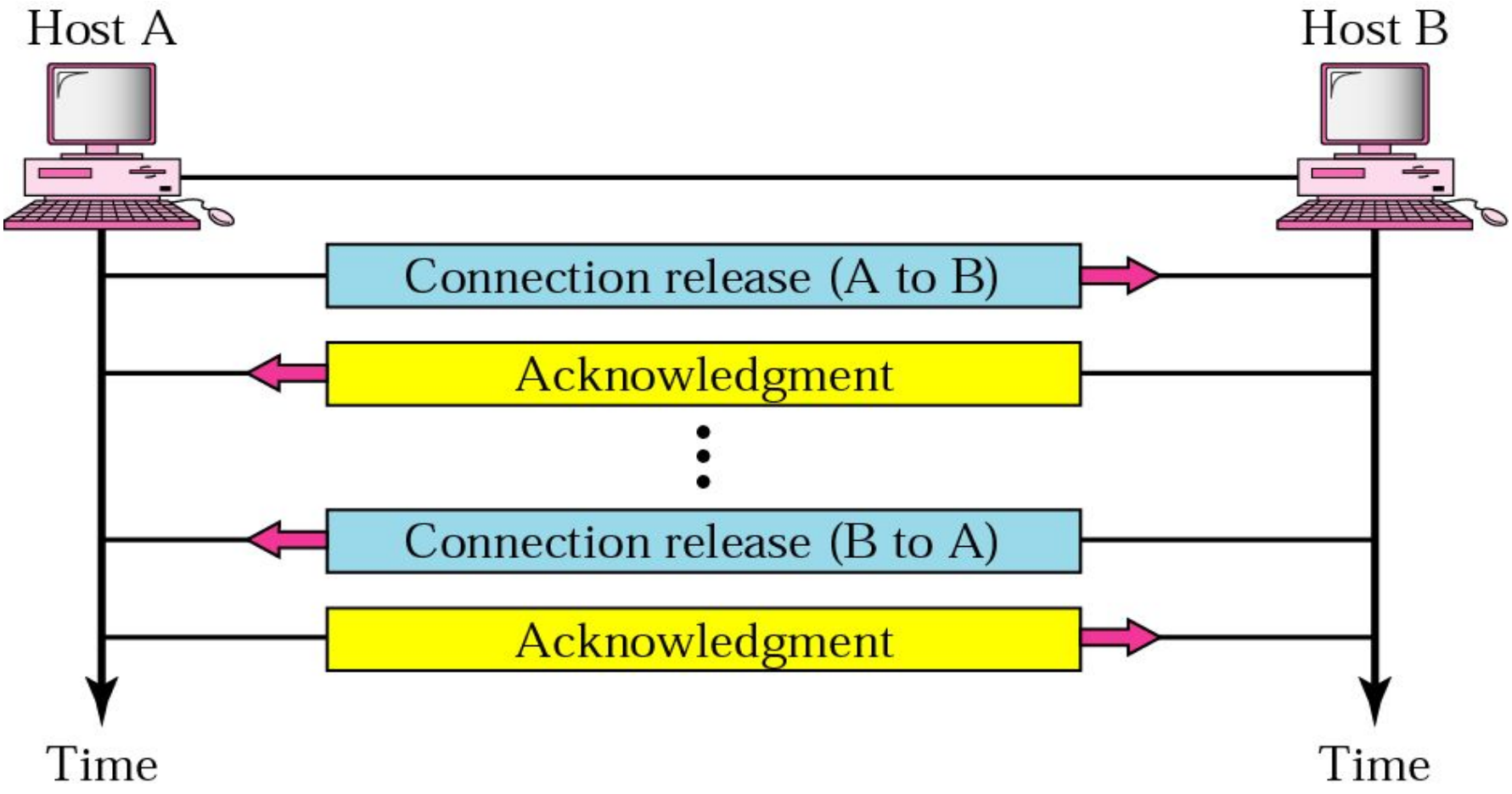


Figure 22.8 Connection termination



22.2 UDP

Port Numbers

User Datagram

Applications

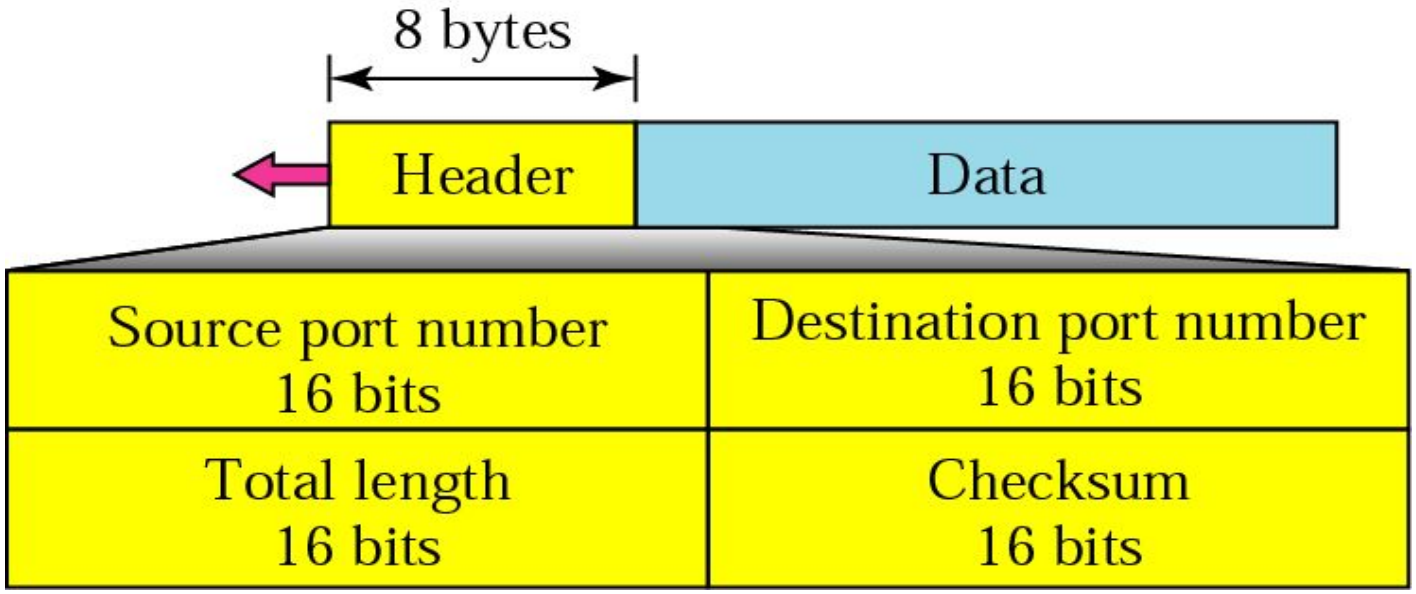
Note:

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.

Note:

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**

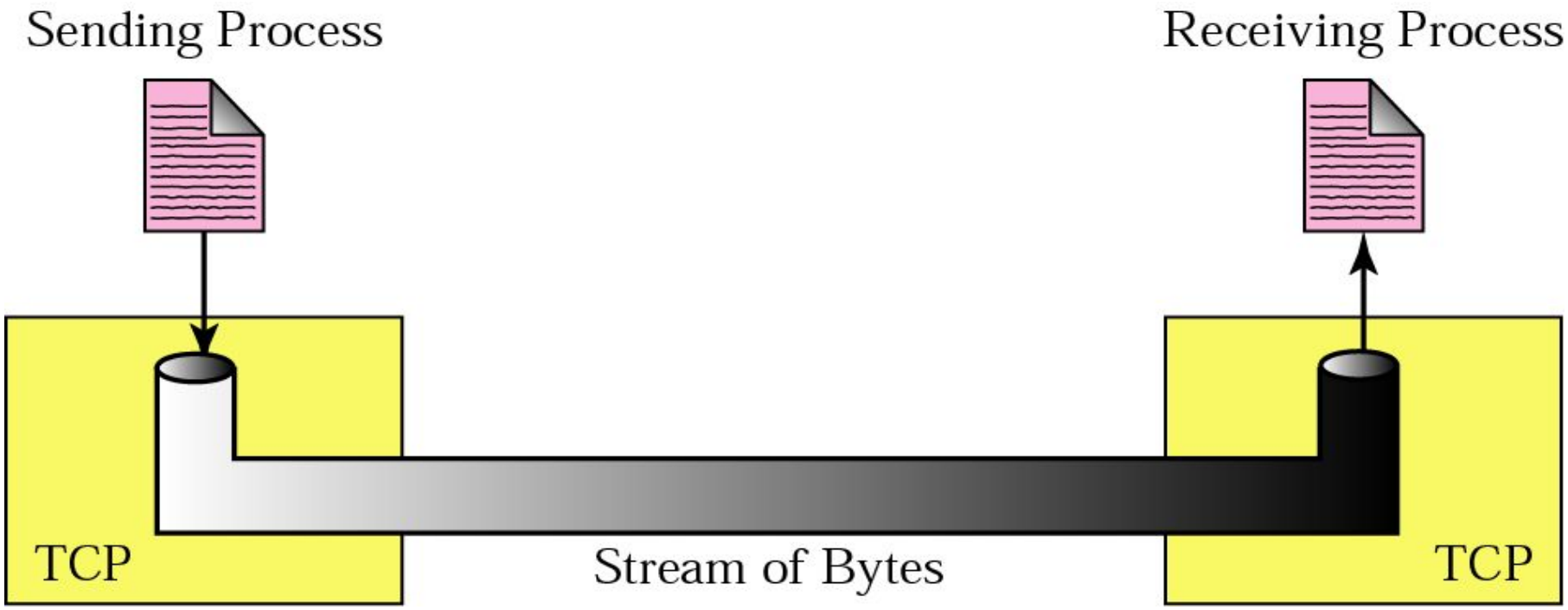
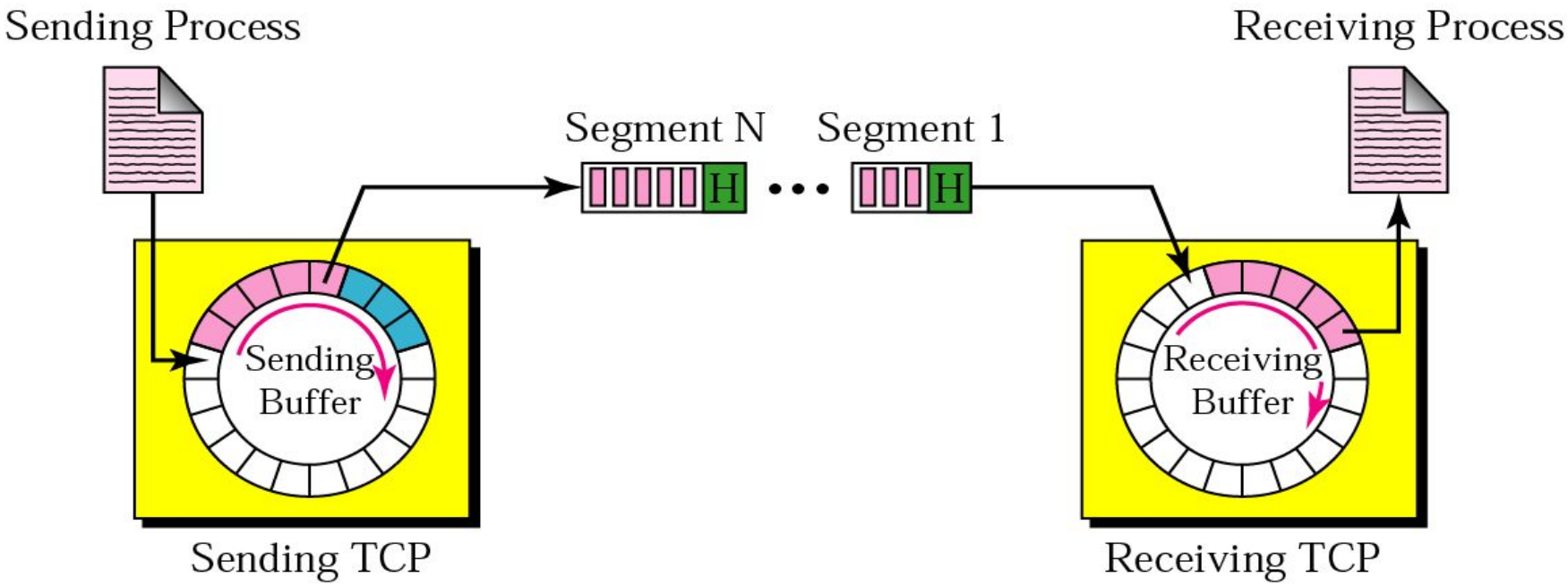


Figure 22.12 Sending and receiving buffers

Figure 22.13 TCP segments



Example 1

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?

Solution

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)

Note:

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

Note:

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

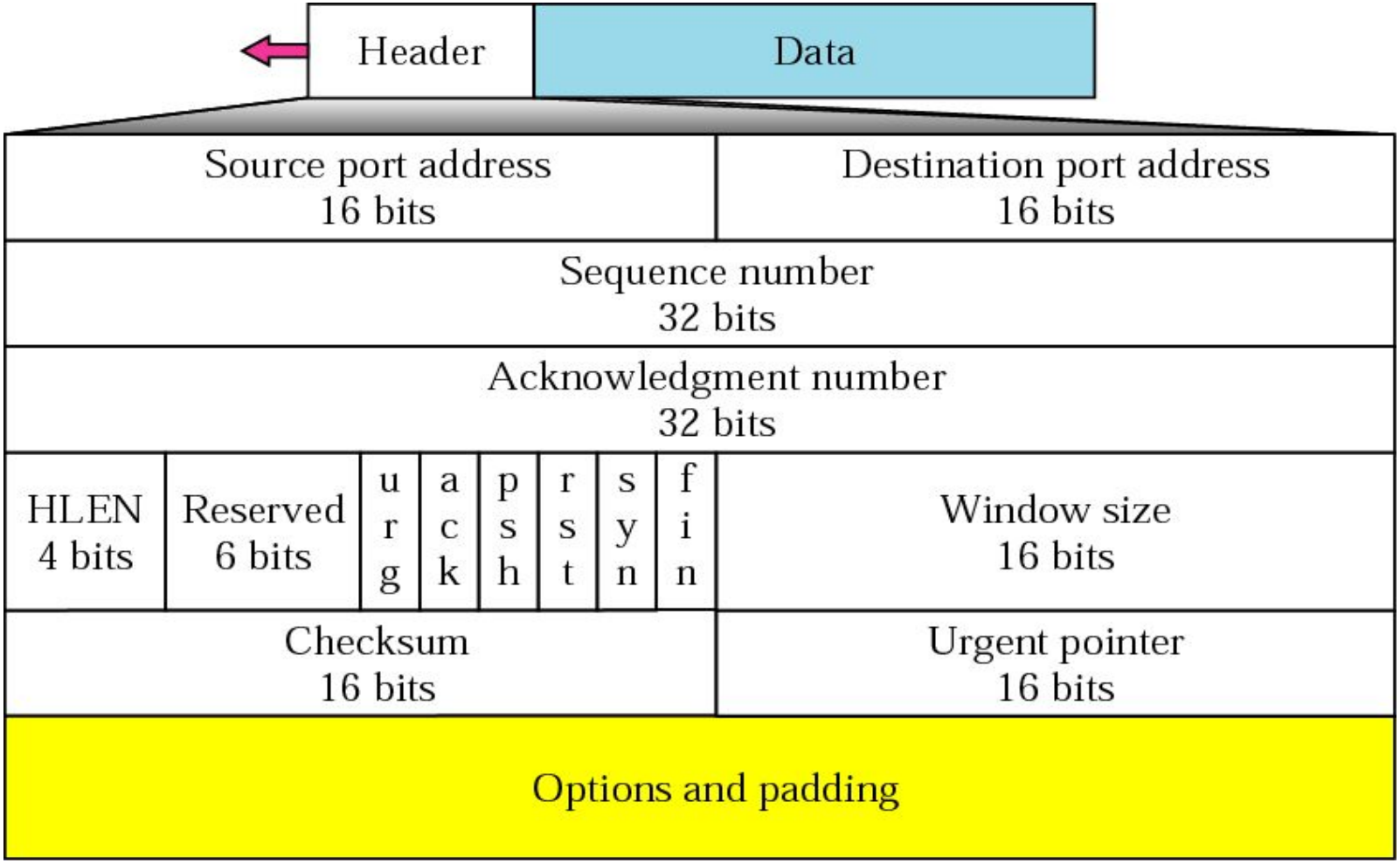


Figure 22.15 Control field

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

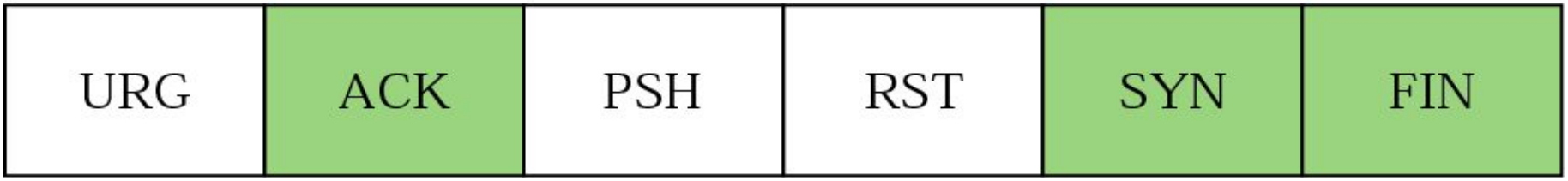


Figure 22.16 Three-step connection establishment

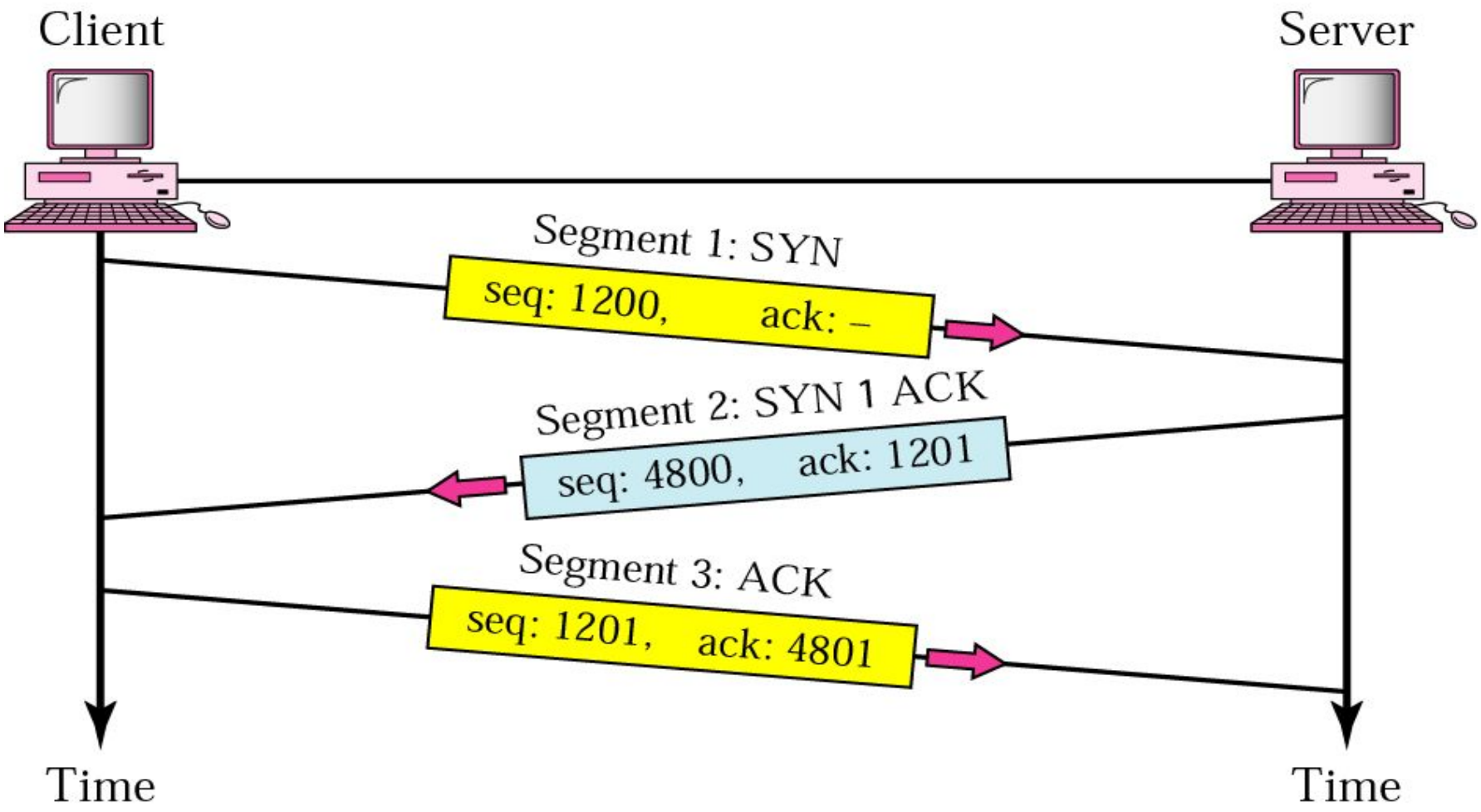


Figure 22.17 Four-step connection termination