COMPUTER NETWORKS

Module – 4: Transport Layer

Dr. Chandrappa S
NAGARJUNA

Transport Layer: Process to Process Delivery: UDP: TCP: TCP services, TCP features, Segment, A TCP connection. SCTP: SCTP services, SCTP features. [23.1,23.2,23.3,23.4]

Congestion Control and Quality of Service: Congestion control: Open loop congestion control and closed loop congestion control. [24.2,24.3]

Quality of Service: Flow Characteristics, Flow Classes, Techniques to improve QoS: Scheduling and Traffic Shaping. [24.5,24.6]

23-1 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.

Figure 23.1 *Types of data deliveries*

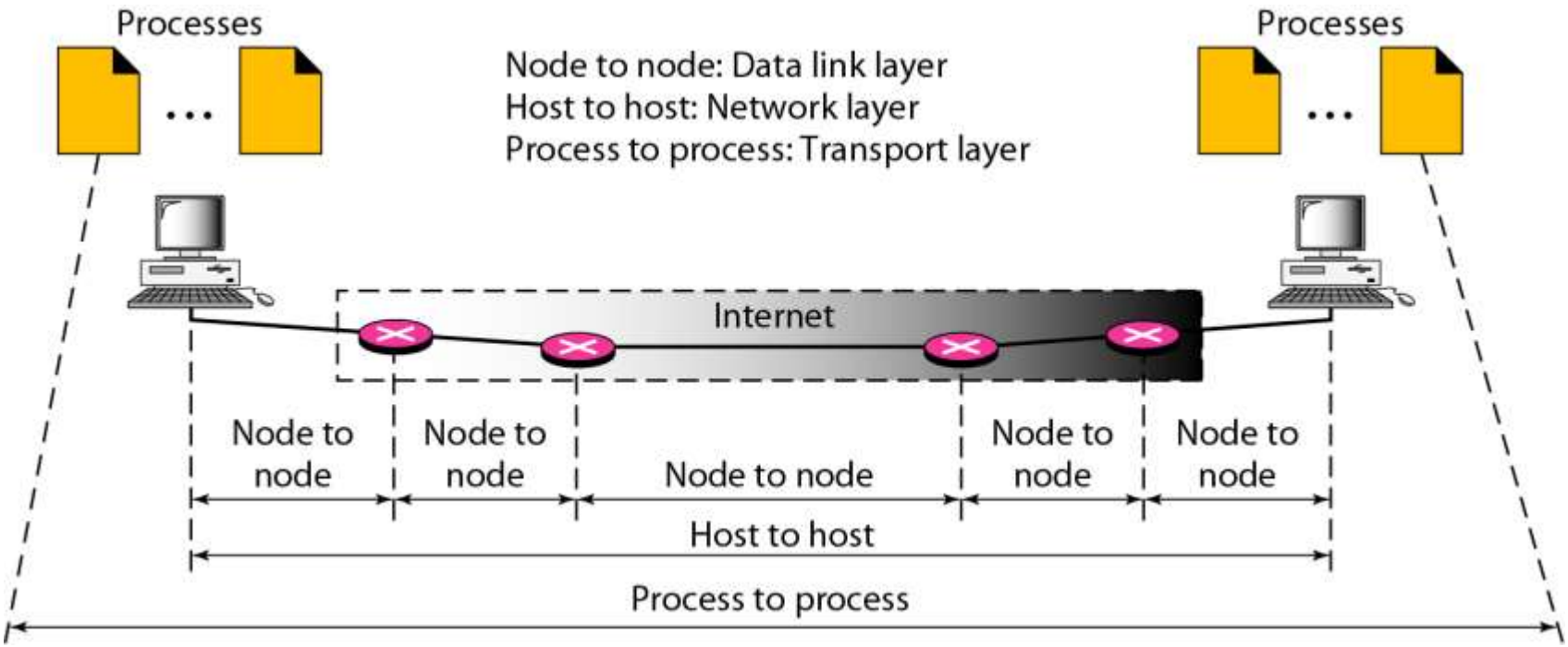


Figure 23.2 *Port numbers*

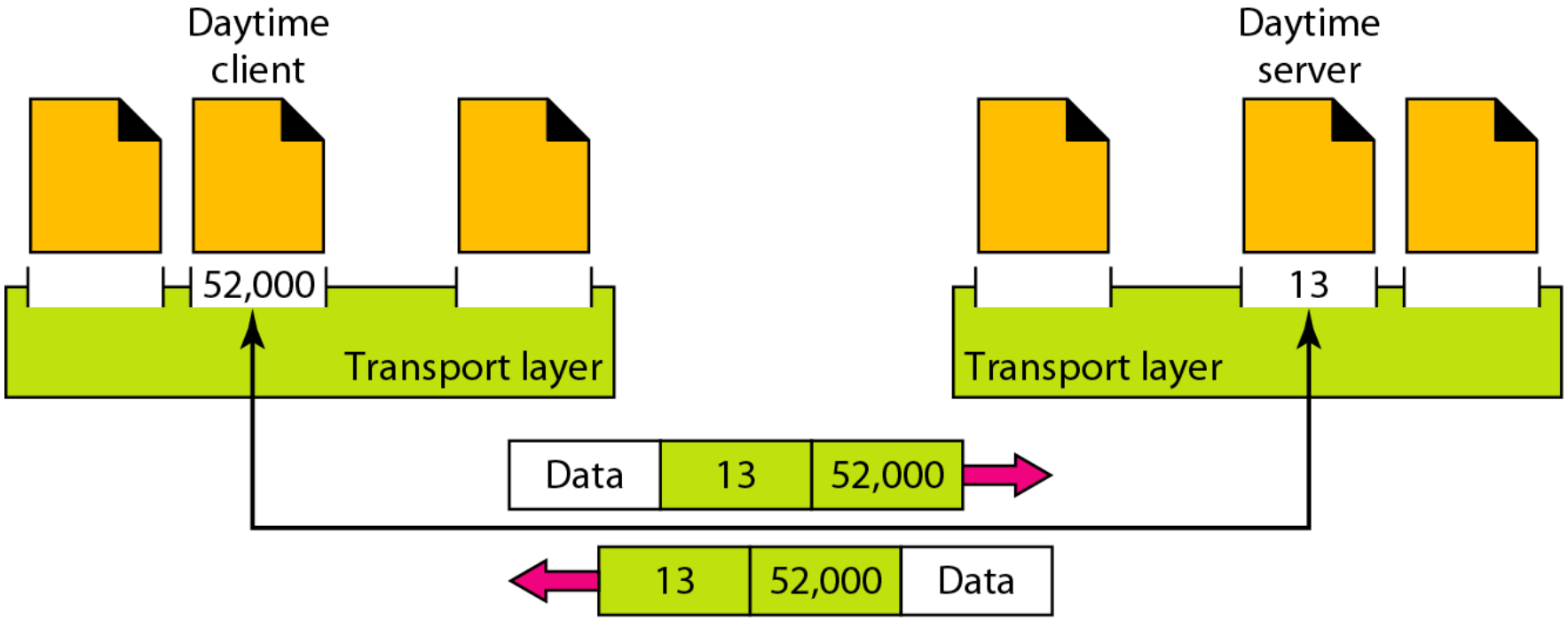


Figure 23.3 *IP addresses versus port numbers*

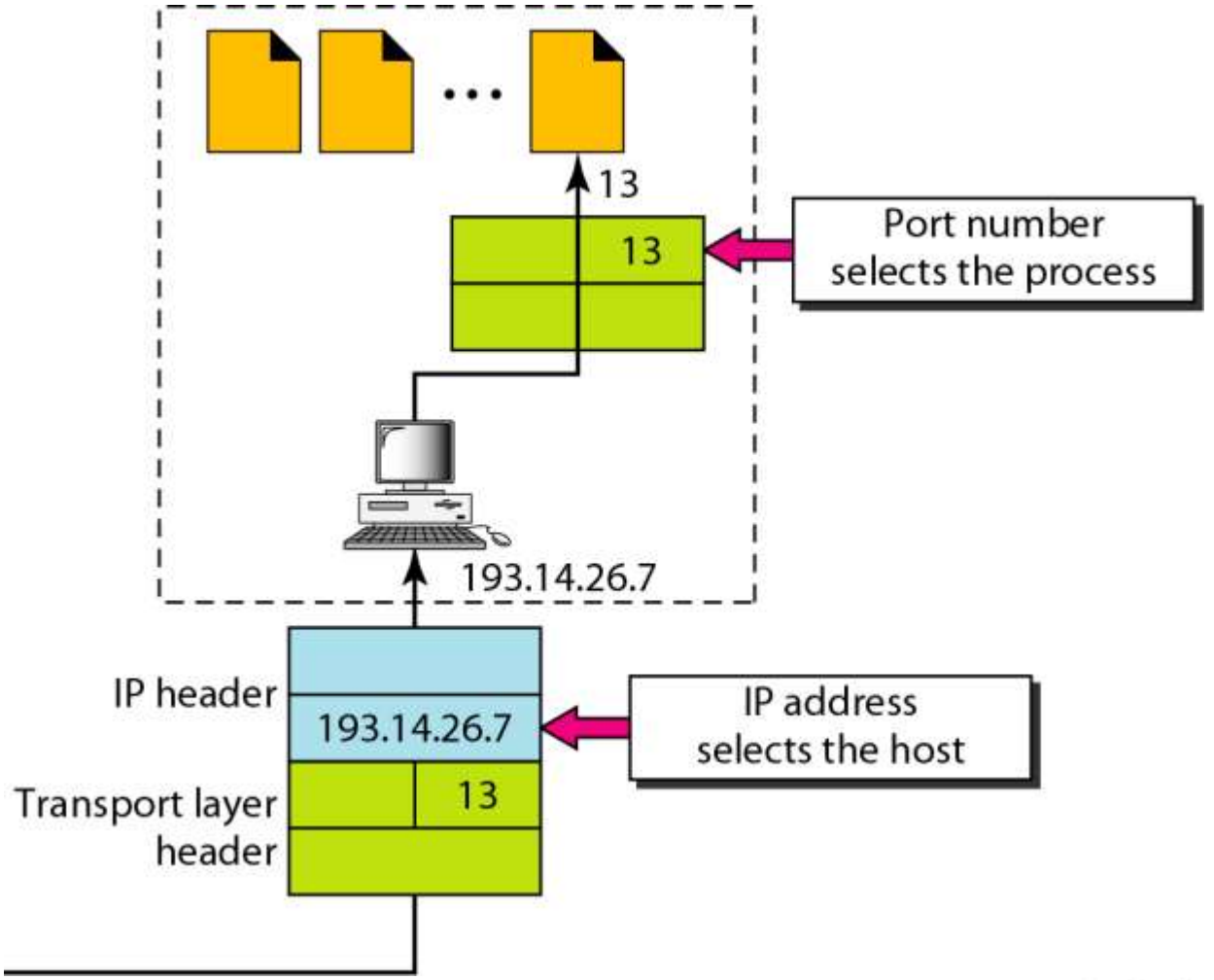


Figure 23.5 *Socket address*

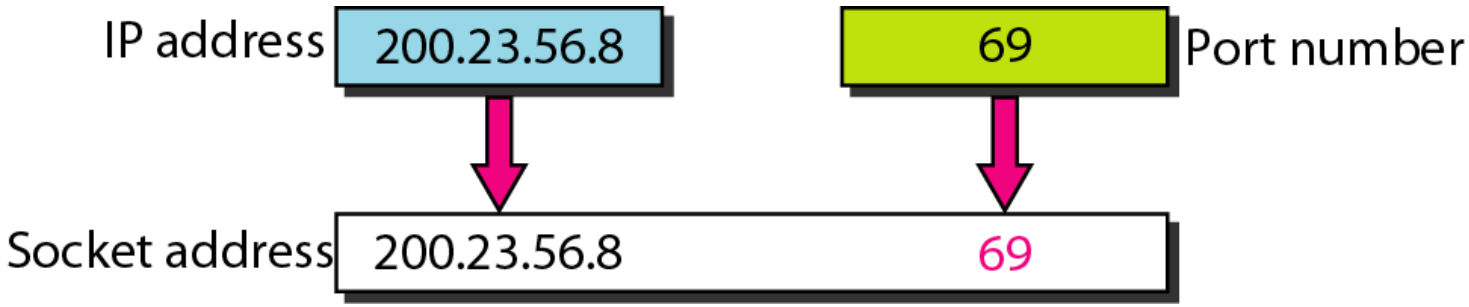


Figure 23.7 *Error control*

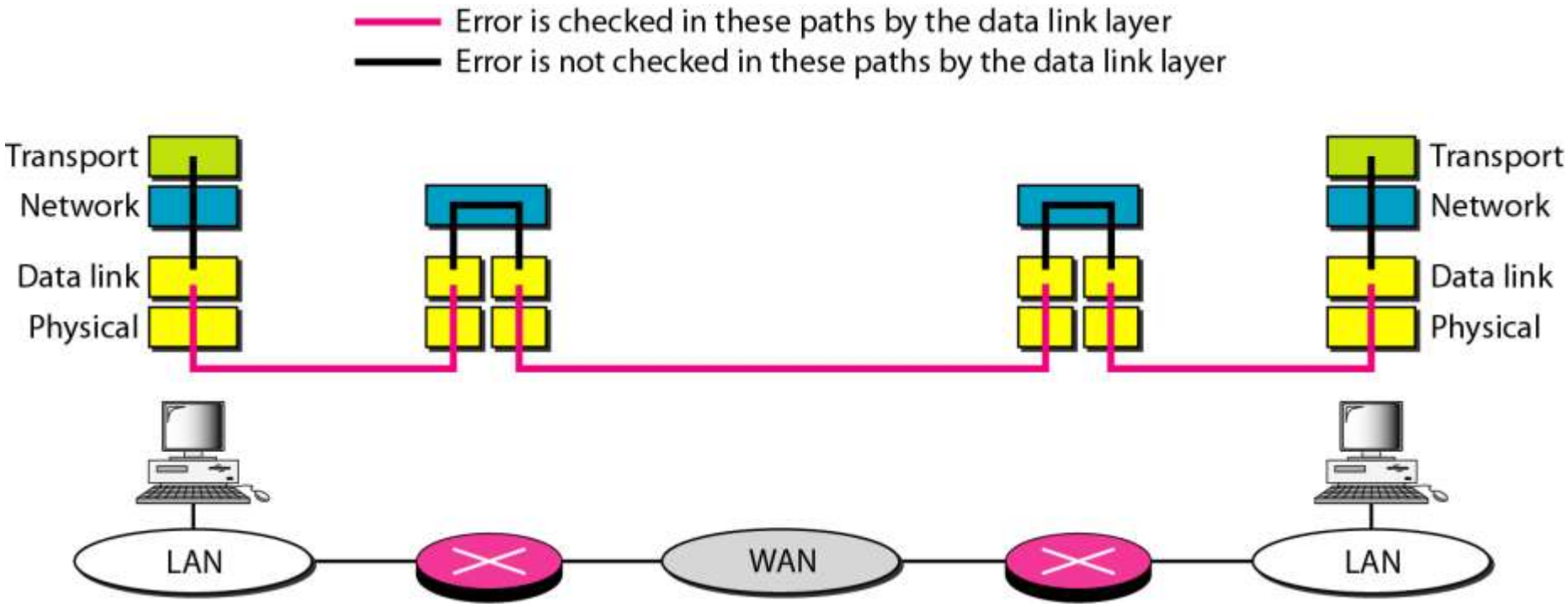
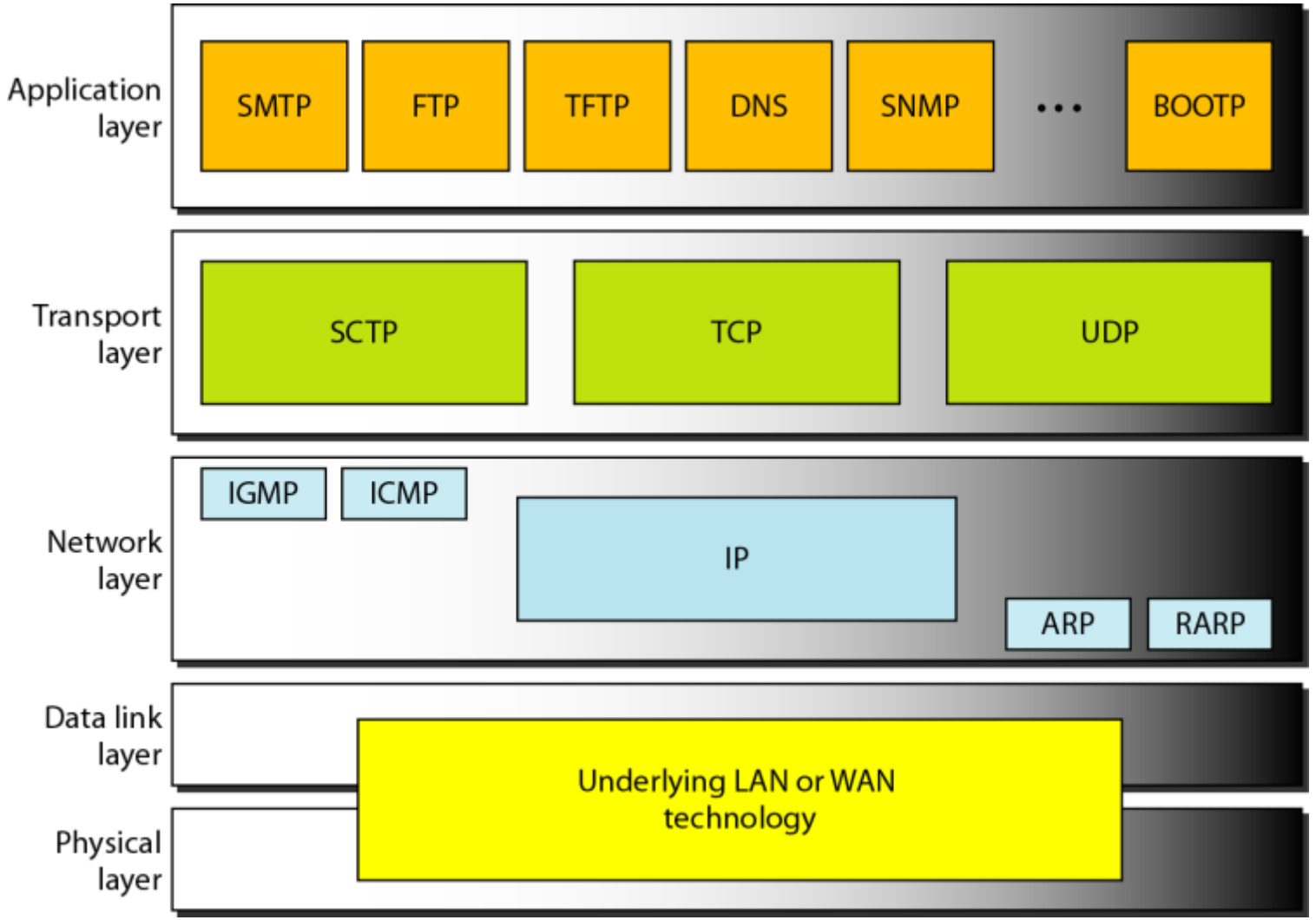


Figure 23.8 *Position of UDP, TCP, and SCTP in TCP/IP suite*



23-2 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

Table 23.1 *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)

Figure 23.9 *User datagram format*

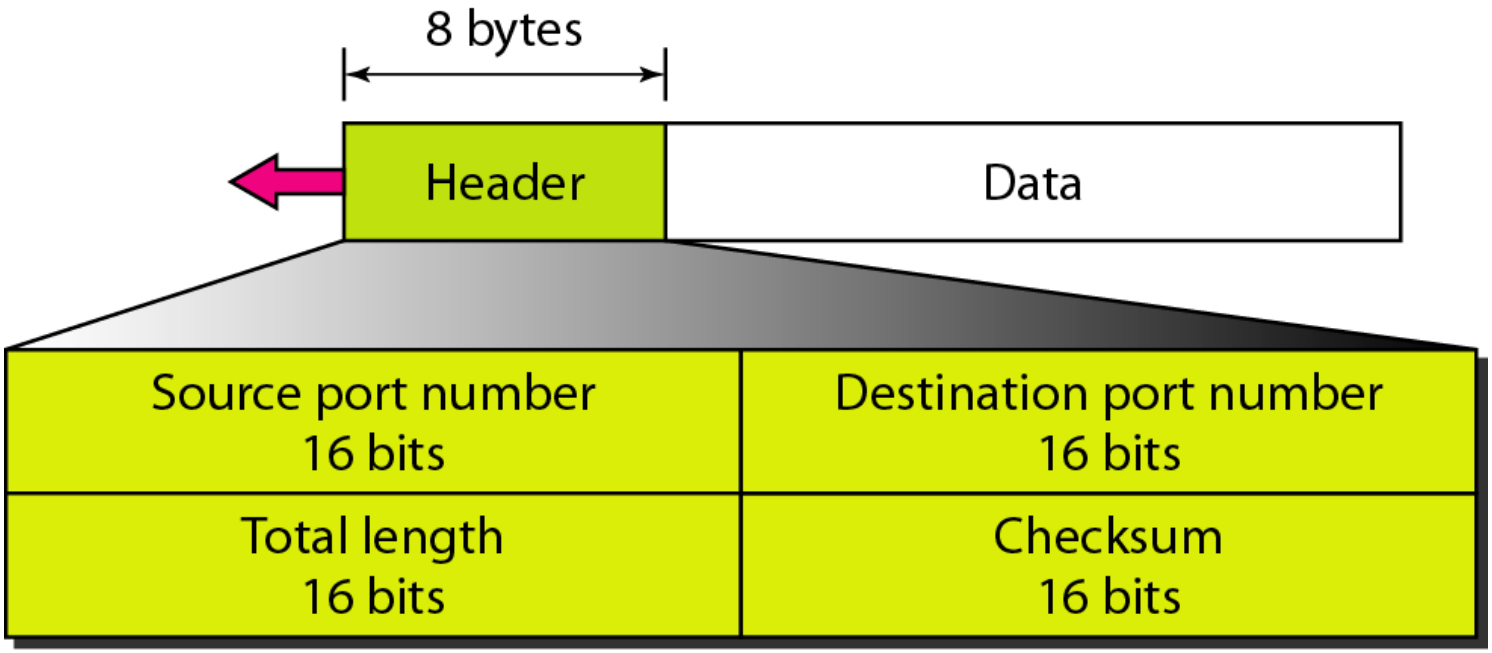
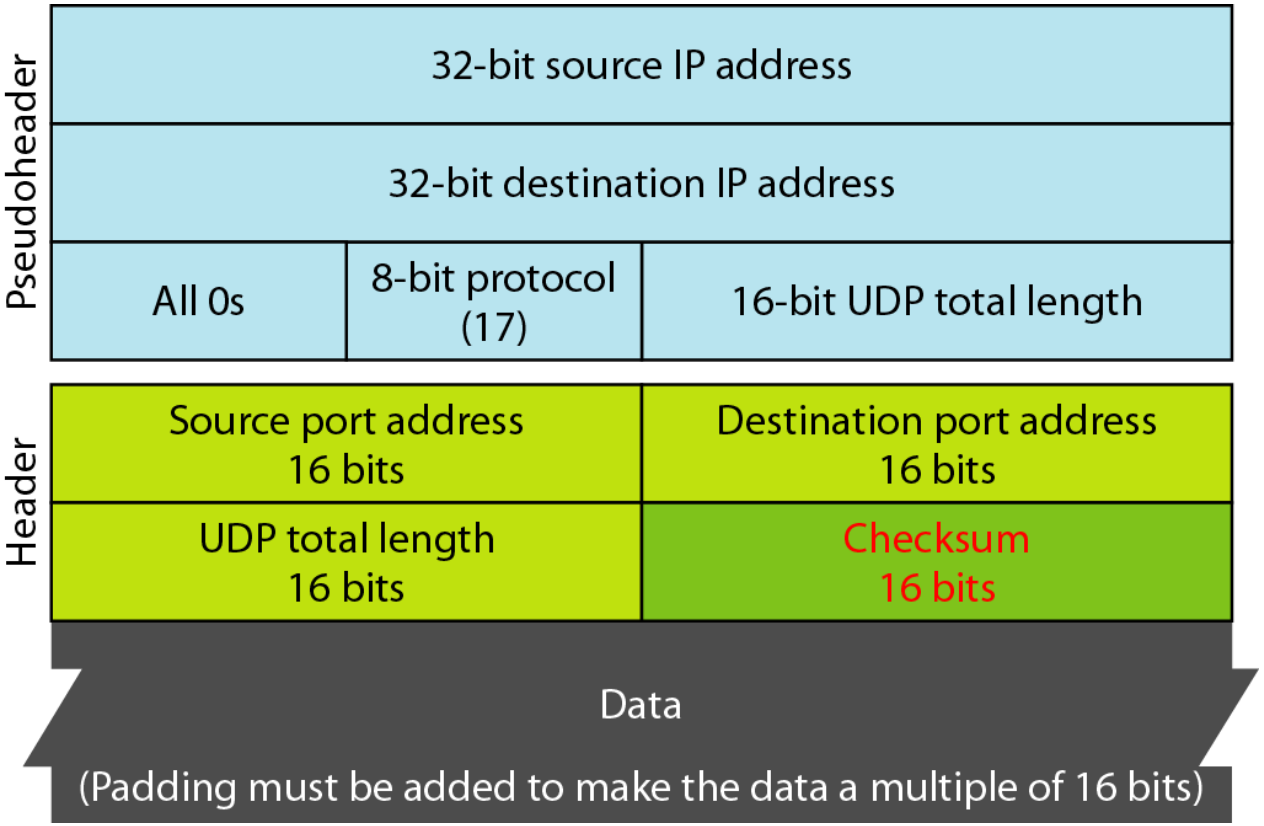


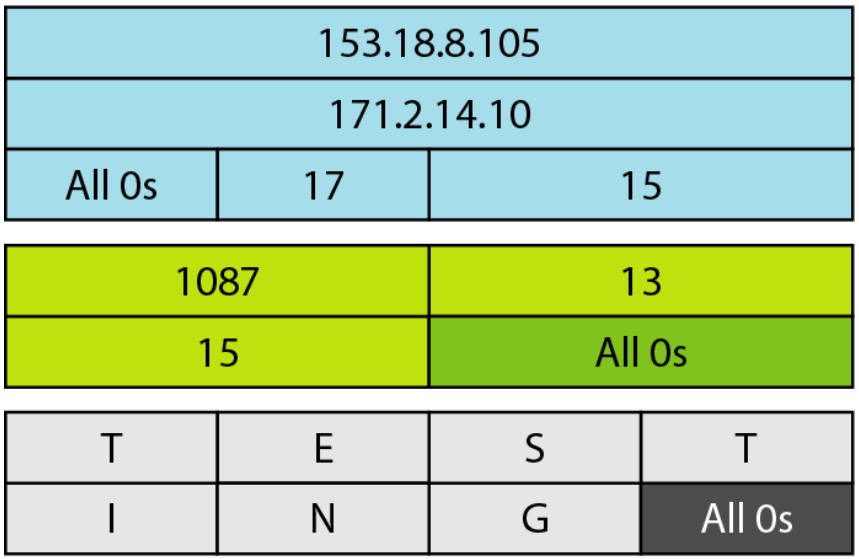
Figure 23.10 *Pseudoheader for checksum calculation*



Example 23.2

Figure 23.11 shows the checksum calculation for a very small user datagram with only 7 bytes of data. Because the number of bytes of data is odd, padding is added for checksum calculation. The pseudoheader as well as the padding will be dropped when the user datagram is delivered to IP.

Figure 23.11 *Checksum calculation of a simple UDP user datagram*

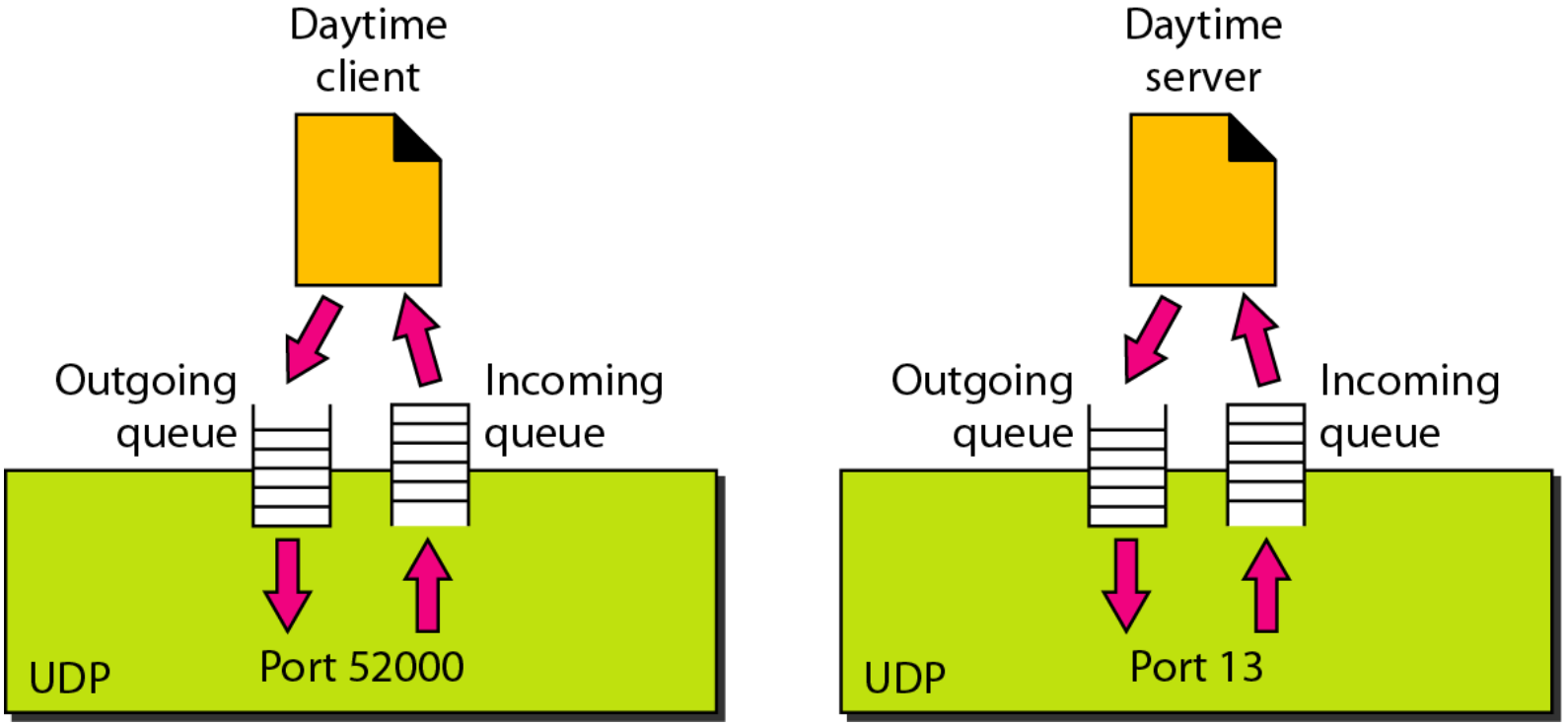


10011001 00010010	153.18
00001000 01101001	8.105
10101011 00000010	171.2
00001110 00001010	14.10
00000000 00010001	0 and 17
00000000 00001111	15
00000100 00111111	1087
00000000 00001101	13
00000000 00001111	15
00000000 00000000	0 (checksum)
01010100 01000101	T and E
01010011 01010100	S and T
01001001 01001110	I and N
01000111 00000000	G and 0 (padding)
10010110 11101011	Sum
01101001 00010100	Checksum

UDP Operation

- *Connectionless Services-This* means that each user datagram sent by UDP is an independent datagram.
- *Flow and Error Control-* There is no flow control and hence no window mechanism. There is no error control mechanism in UDP except for the checksum
- *Encapsulation and Decapsulation*-To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages in an IP datagram.
- *Queuing*

Figure 23.12 *Queues in UDP*



Use of UDP

- UDP is suitable for a process that requires simple request-response communication with little concern for flow and error control.
- UDP is suitable for a process with internal flow- and error-control mechanisms.
- UDP is a suitable transport protocol for multicasting.
- UDP is used for management processes such as SNMP
- UDP is used for some route updating protocols such as Routing Information Protocol (RIP)

23-3 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

TCP Services

Process-to-Process Communication-TCP

provides process-to-process communication using port numbers

Table 23.2 ***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

Stream Delivery-sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes

Figure 23.13 ***Stream delivery***

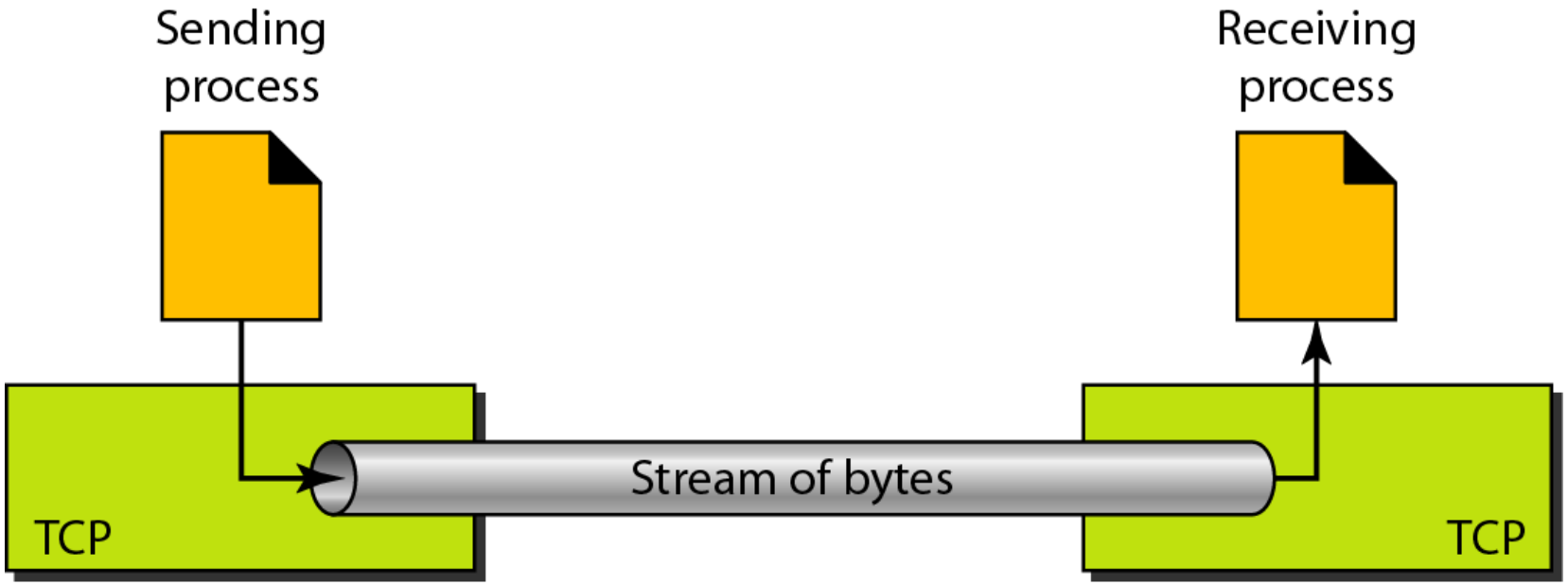


Figure 23.14 *Sending and receiving buffers*

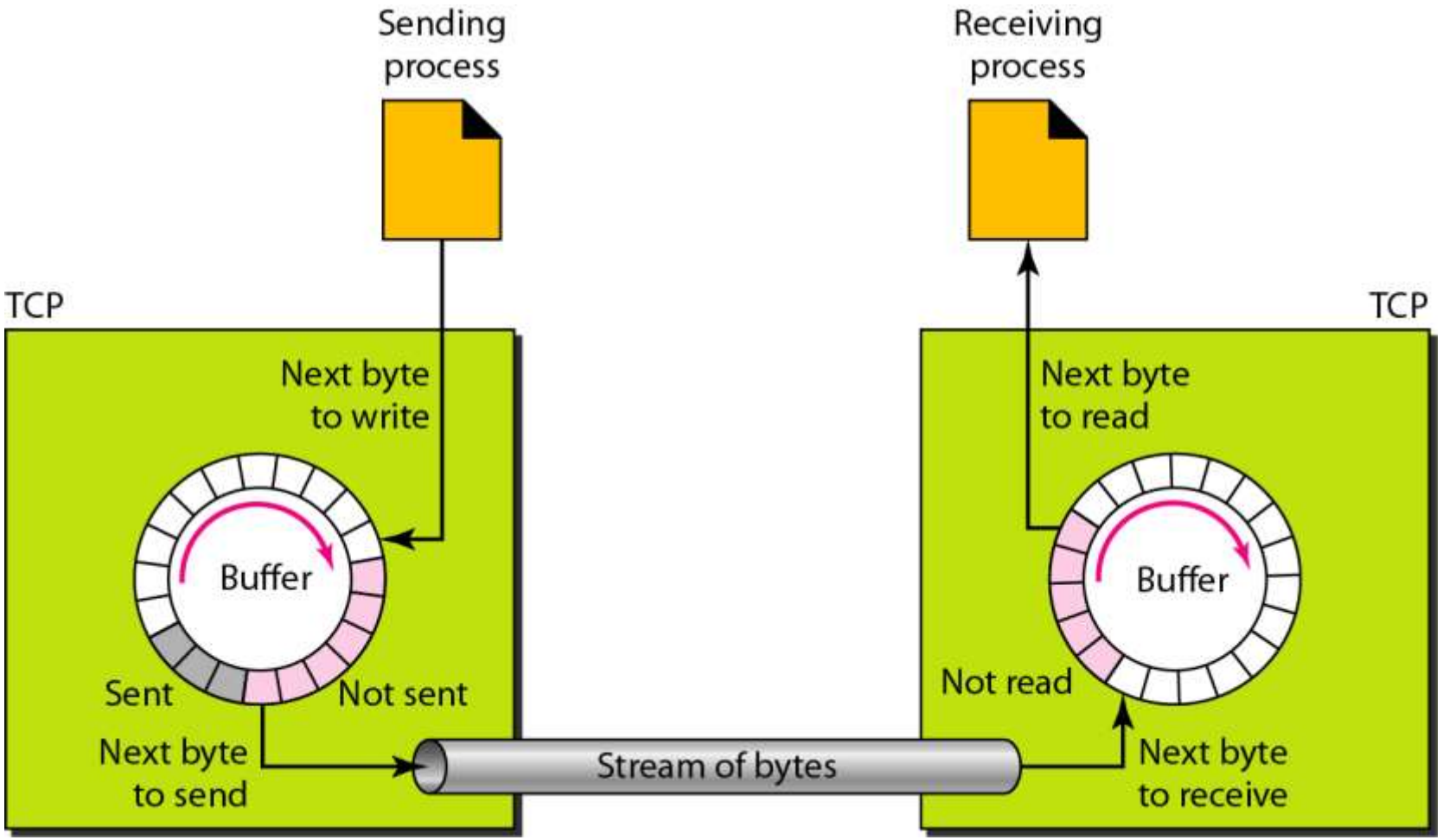
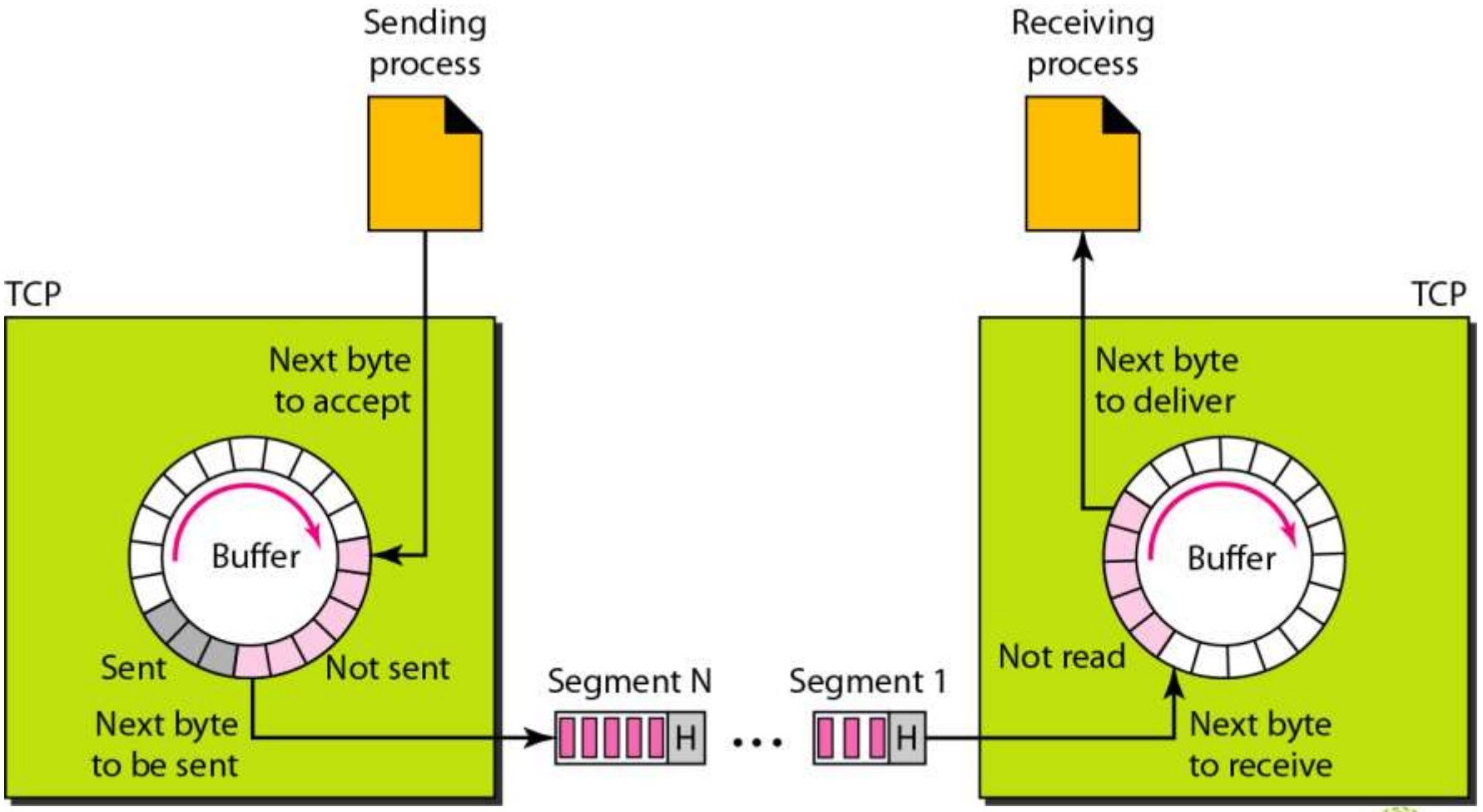


Figure 23.15 *TCP segments*



Full-Duplex Communication

Connection-Oriented Service

Reliable Service-It uses an acknowledgment mechanism to check the safe and sound arrival of data

TCP Features

Numbering System-

Byte Number: TCP numbers all data bytes that are transmitted in a connection. Numbering is independent in each direction. When TCP receives bytes of data from a process, it stores them in the sending buffer and numbers them. The numbering does not necessarily start from O. Instead, TCP generates a random number between 0 and $2^{32} - 1$ for the number of the first byte

Sequence Number : After the bytes have been numbered, TCP assigns a sequence number to each segment that is being sent. **The sequence number for each segment is the number of the first byte carried in that segment**.

Acknowledgment Number: confirms the bytes it has received. defines the number of the next byte that the party expects to receive. If party uses 5643 as an acknowledgment number, it has received all bytes from the beginning up to 5642

Example 23.3

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)**

Flow Control
Error Control
Congestion Control

Note

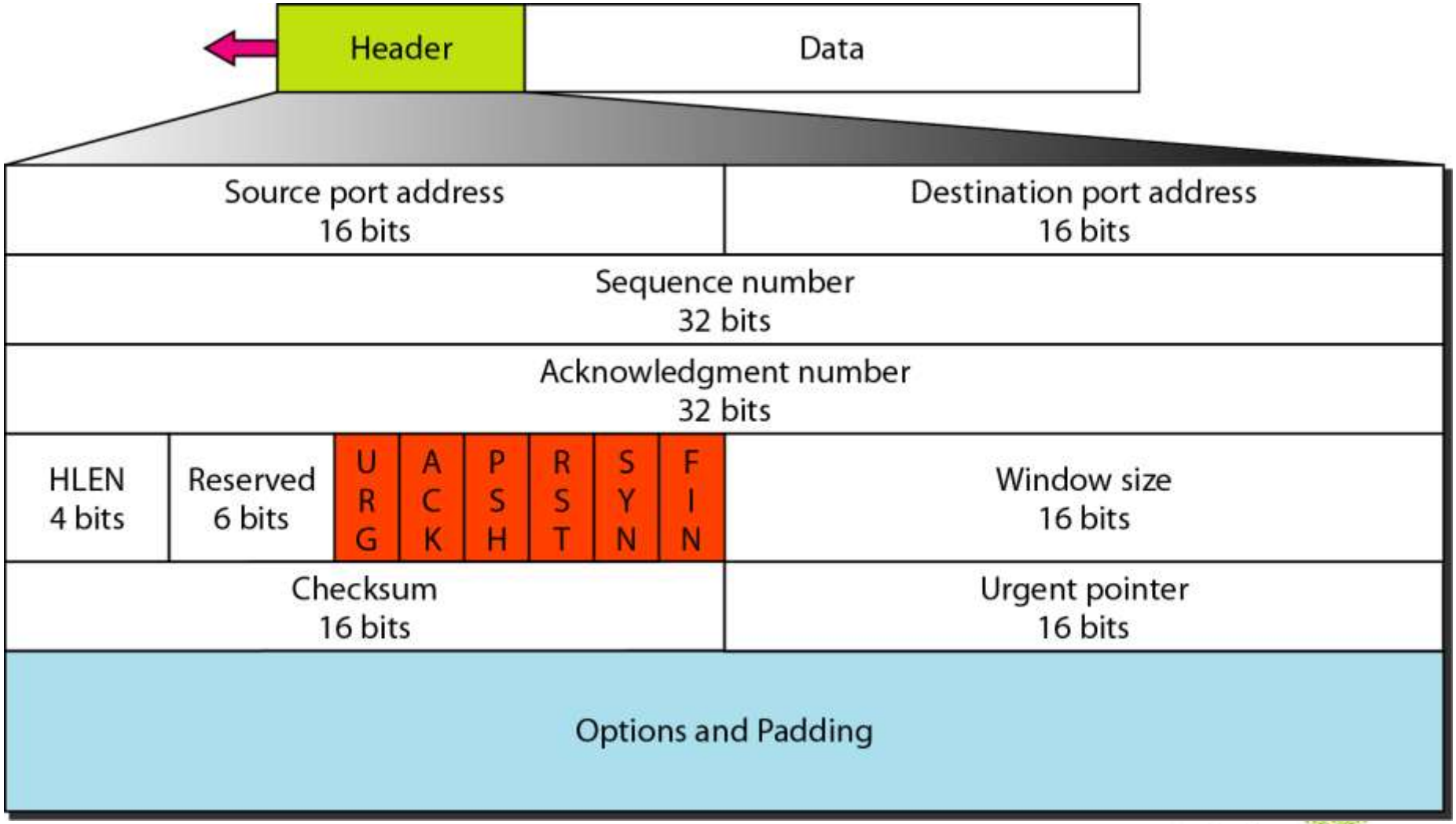
The value in the sequence number field of a segment defines the number of the first data byte contained in that segment.

Note

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

Segment

Figure 23.16 *TCP segment format*

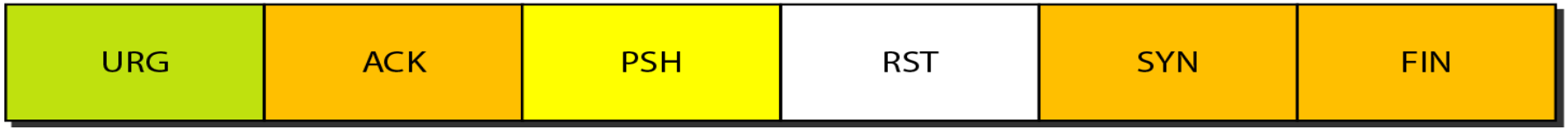


- A TCP segment consists of two main parts:
  - Header – contains control information
  - Data – carries the actual payload from the application

1. Source Port Address (16 bits)
   - Identifies the sending application on the source device.
2. Destination Port Address (16 bits)
   - Identifies the receiving application on the destination device.
3. Sequence Number (32 bits)
   - Specifies the byte number of the first data byte in this segment.
   - Used for ordering and reliable delivery.
4. Acknowledgment Number (32 bits)
   - Indicates the next expected byte from the sender.
   - Used to confirm successful data receipt.
5. Header Length (HLEN – 4 bits)
   - Specifies the length of the TCP header in 32-bit words (minimum = 5).
6. Reserved (6 bits)
   - Reserved for future use; must be set to 0.

8. *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



Flag	Full Form	Purpose
URG	Urgent	Indicates urgent data (urgent pointer valid)
ACK	Acknowledgment	Acknowledgment field is valid
PSH	Push	Request to push buffered data to the application
RST	Reset	Reset the connection
SYN	Synchronize	Used to initiate a connection
FIN	Finish	Used to terminate a connection

8. Window Size (16 bits)-Specifies how many bytes the sender is willing to receive (flow control).

9. Checksum (16 bits)-Used for error detection across the entire segment (header + data).

10. Urgent Pointer (16 bits)Points to the end of urgent data in the segment (used if URG flag is set)

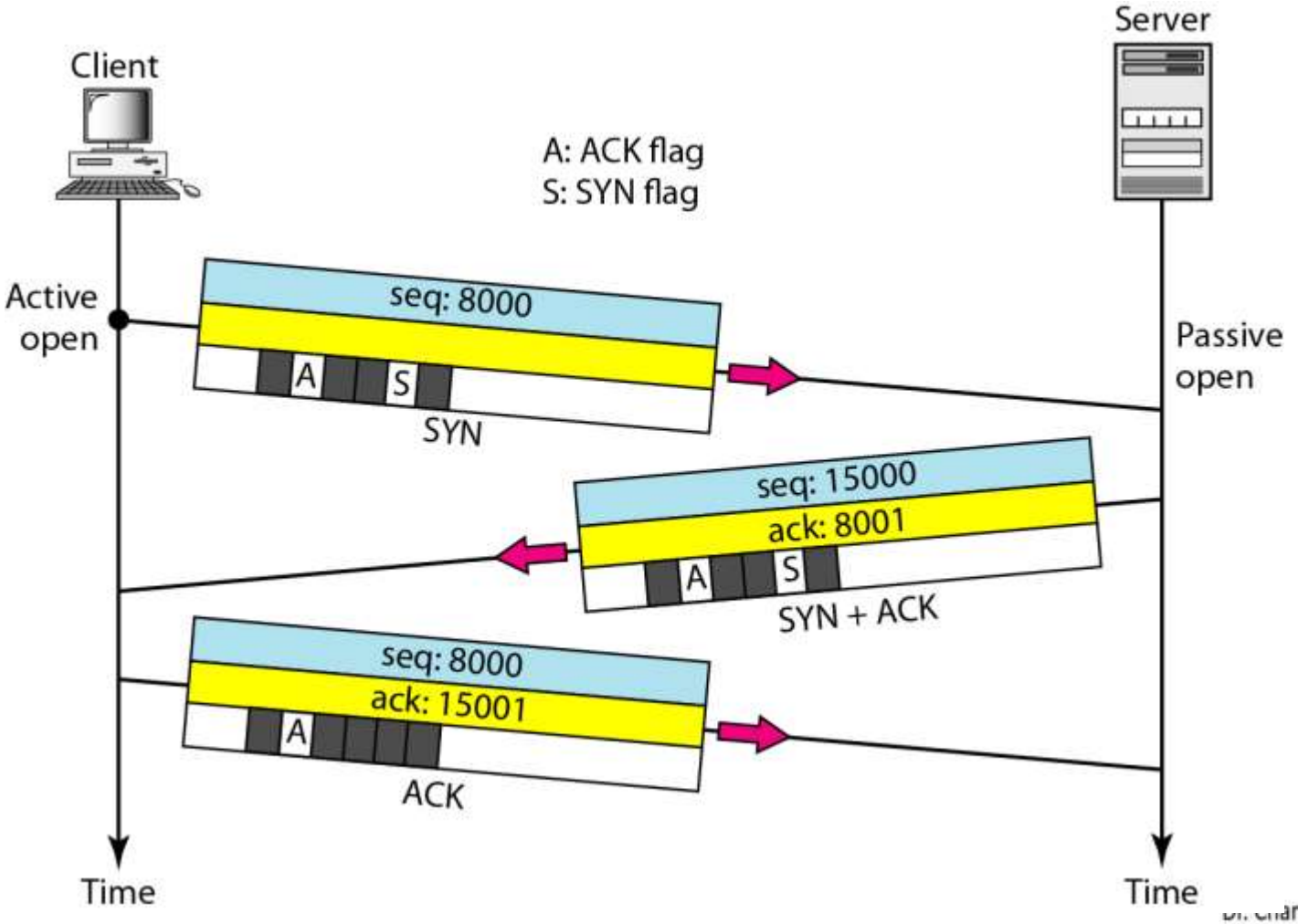
11. Options and Padding (variable length)-Optional parameters used for performance (e.g., Maximum Segment Size, window scaling).Padding ensures the header length is a multiple of 32 bits.

12. Data-Contains the actual message (payload) being transmitted.

1. Connection Establishment

1. The client sends the first segment, a SYN segment, in which only the SYN flag is set. When the data transfer starts, the sequence number is incremented by 1. We can say that the SYN segment carries no real data
2. The server sends the second segment, a SYN +ACK segment, with 2 flag bits set: SYN and ACK
3. The client sends the third segment. This is just an ACK segment. It acknowledges the receipt of the second segment with the ACK flag

Figure 23.18 *Connection establishment using three-way handshaking*

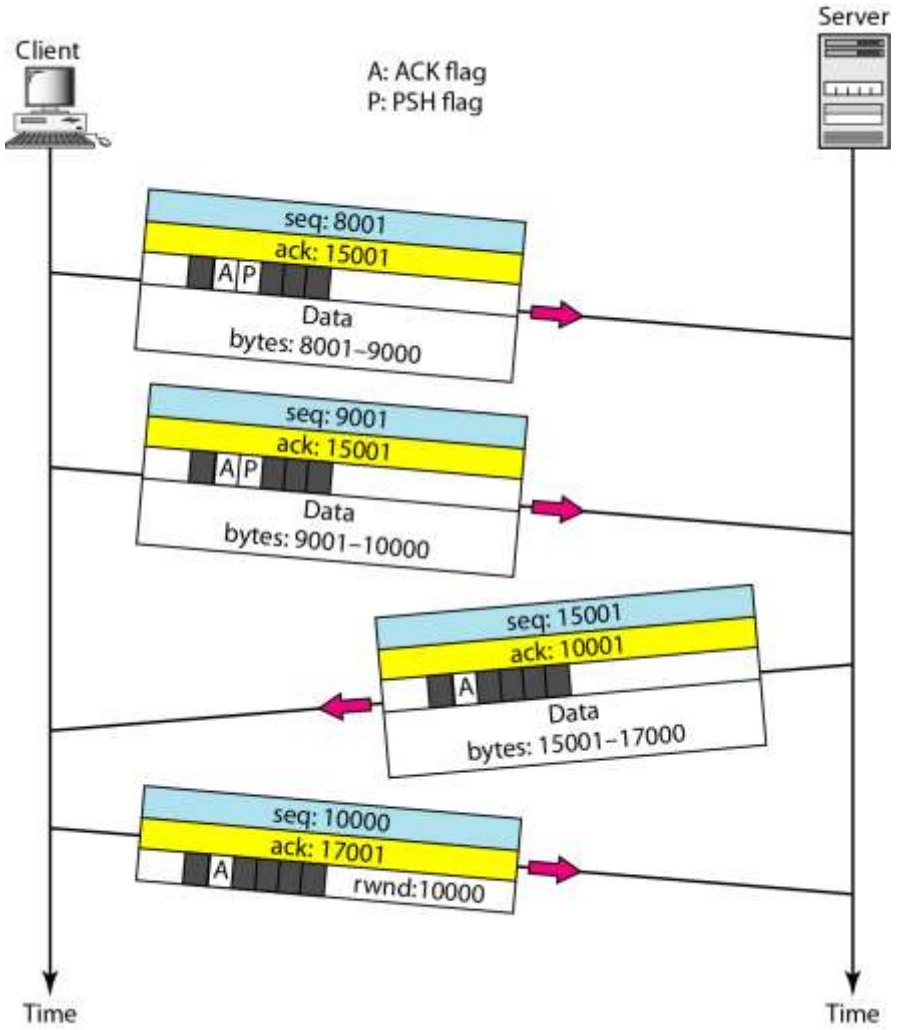


2. Data transfer

- After connection is established (not shown in the figure), the client
- sends 2000 bytes of data in two segments.
- The server then sends 2000 bytes in one segment.
- The client sends one more segment.
- The first three segments carry both data and acknowledgment, but the last segment carries only an acknowledgment because there are no more data to be sent
- PSH (push) flag set by the client indicates that server TCP knows to deliver data to the server process as soon as they are received

- send a segment with the URG bit set. The sending application program tells the sending TCP that the piece of data is urgent.
- The sending TCP creates a segment and inserts the urgent data at the beginning of the segment.
- The urgent pointer field in the header defines the end of the urgent data and the start of normal data.
- When the receiving TCP receives a segment with the URG bit set, it extracts the urgent data from the segment, using the value of the urgent pointer, and delivers them, out of order, to the receiving application program

Figure 23.19 *Data transfer*



3. Connection Termination

1. In a normal situation, the client TCP, after receiving a close command from the client process, sends the first segment, a FIN segment in which the FIN flag is set.
2. The server TCP, after receiving the FIN segment, informs its process of the situation and sends the second segment, a FIN +ACK segment, to confirm the receipt
3. The client TCP sends the last segment, an ACK segment, to confirm the receipt of the FIN segment

Figure 23.20 *Connection termination using three-way handshaking*

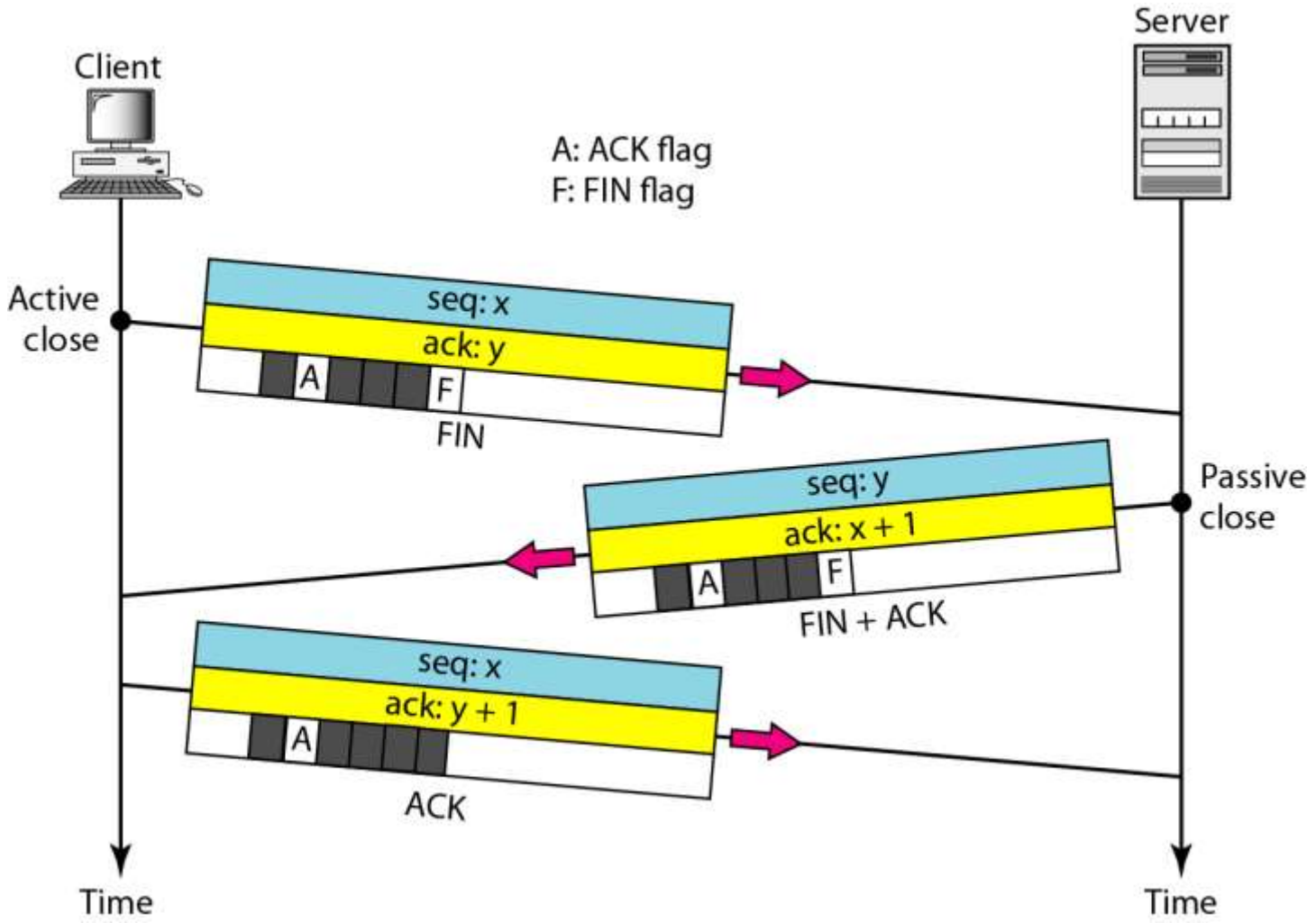
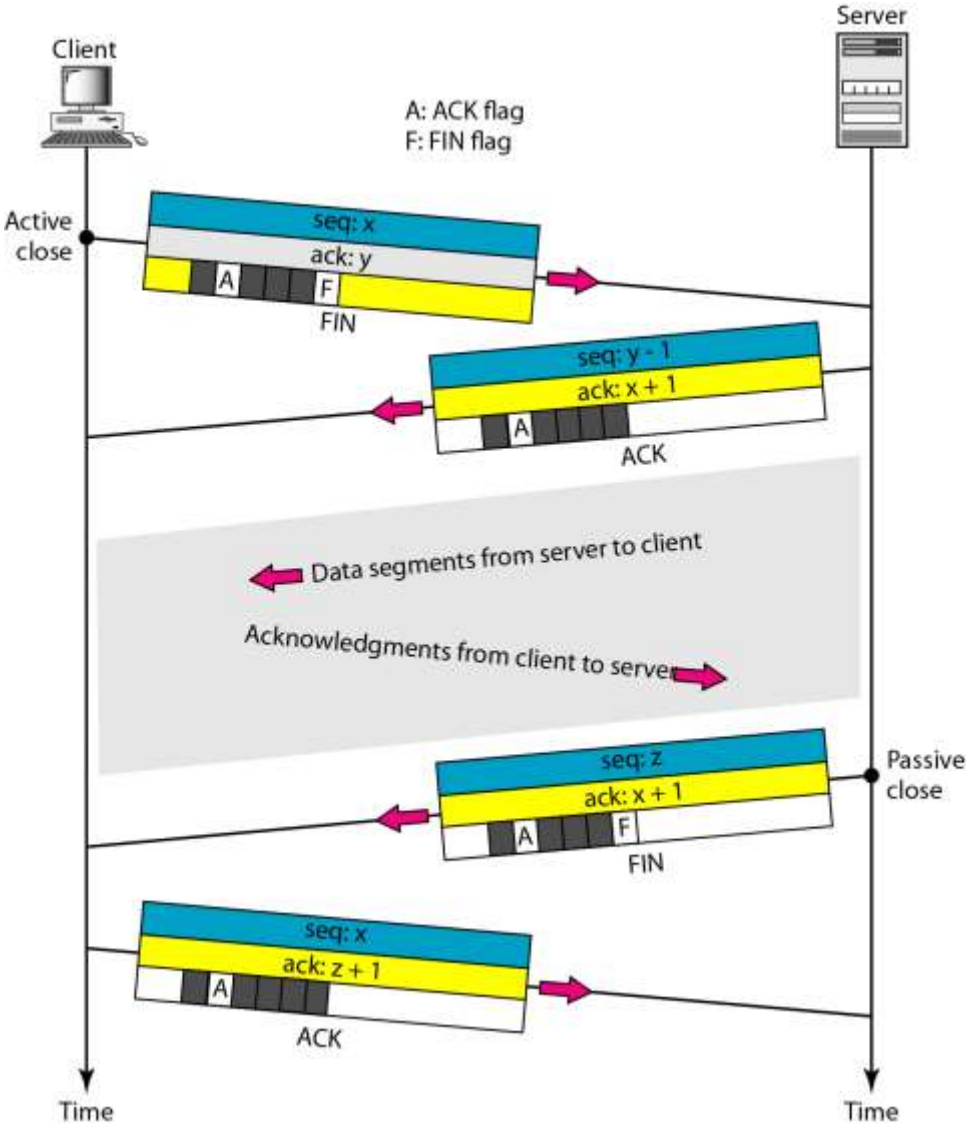
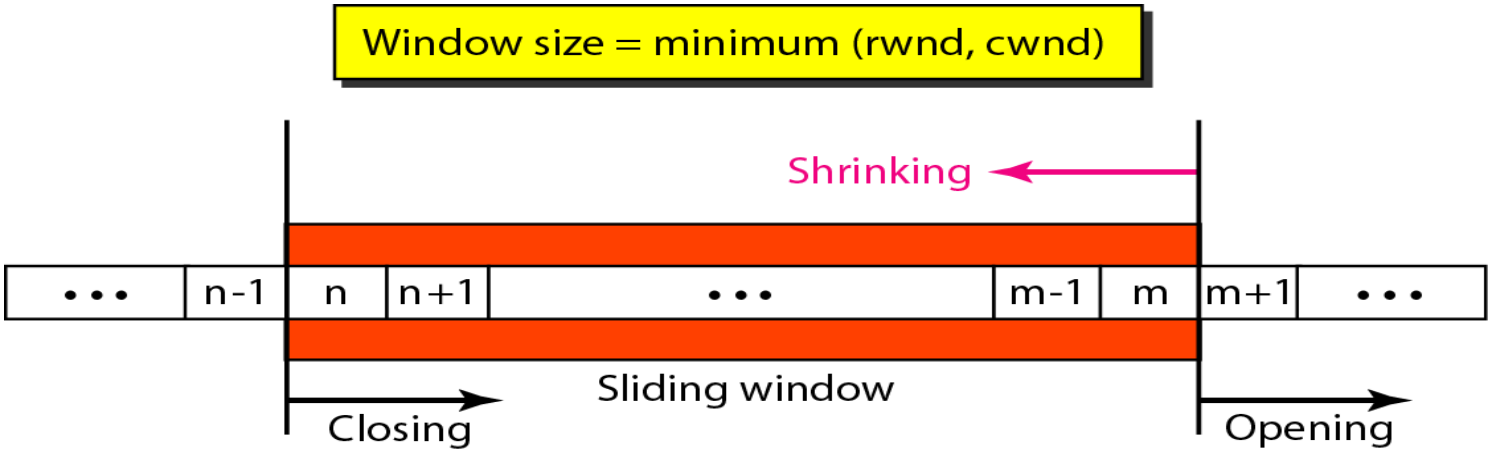


Figure 23.21 *Half-close*



Flow Control

Figure 23.22 *Sliding window*



- Opening a window means moving the right wall to the right. This allows more new bytes in the buffer that are eligible for sending.
- Closing the window means moving the left wall to the right. This means that some bytes have been acknowledged and the sender
- Srinking the window means moving the right wall to the left. it means revoking the eligibility of some bytes for sending

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.

Example 23.4

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.

Example 23.5

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?

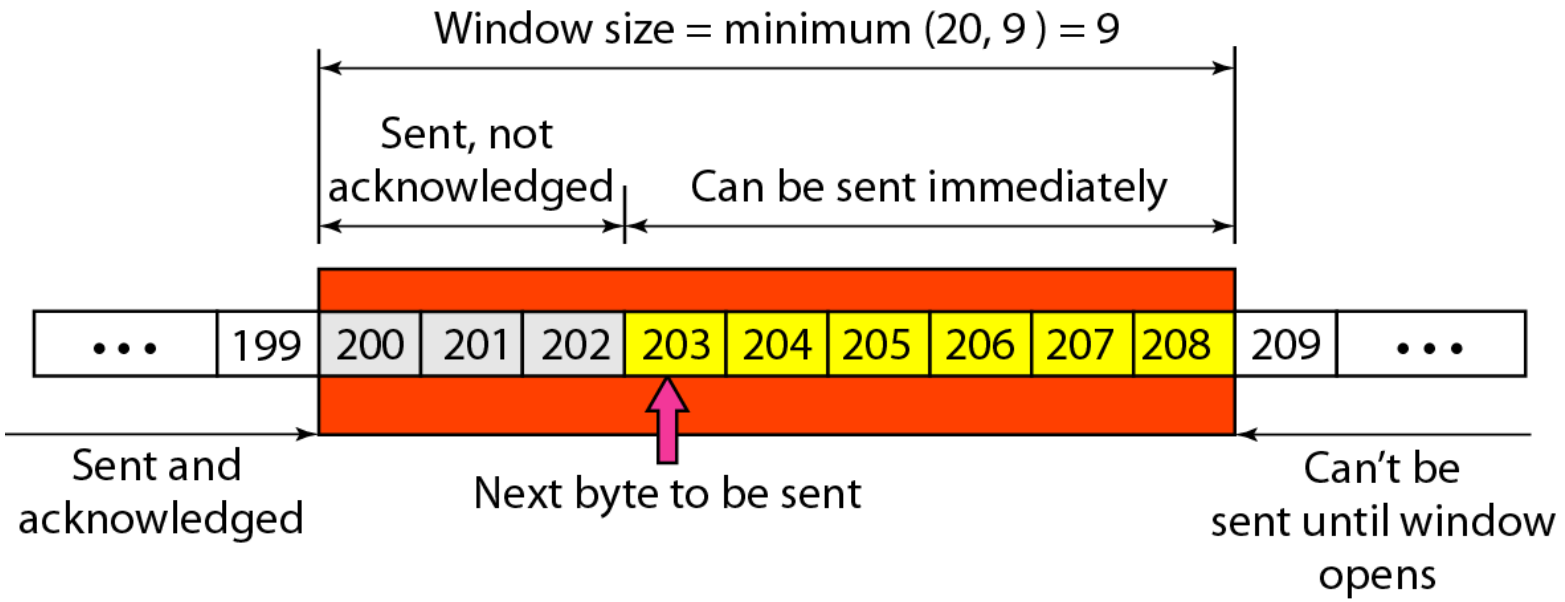
Solution

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

Example 23.6

Figure 23.23 shows an unrealistic example of a sliding window. The sender has sent bytes up to 202. We assume that cwnd is 20 (in reality this value is thousands of bytes). The receiver has sent an acknowledgment number of 200 with an rwnd of 9 bytes (in reality this value is thousands of bytes). The size of the sender window is the minimum of rwnd and cwnd, or 9 bytes. Bytes 200 to 202 are sent, but not acknowledged. Bytes 203 to 208 can be sent without worrying about acknowledgment. Bytes 209 and above cannot be sent.

Figure 23.23 *Example 23.6*



Note

Some points about TCP sliding windows:

- The size of the window is the lesser of rwnd and cwnd.
- The source does not have to send a full window's worth of data.
- The window can be opened or closed by the receiver, but should not be shrunk.
- The destination can send an acknowledgment at any time as long as it does not result in a shrinking window.
- The receiver can temporarily shut down the window; the sender, however, can always send a segment of 1 byte after the window is shut down.

Error Control Methods- *Checksum, Acknowledgment,*

Retransmission After RTO, Retransmission After Three Duplicate ACK

Figure 23.24 ***Normal operation***

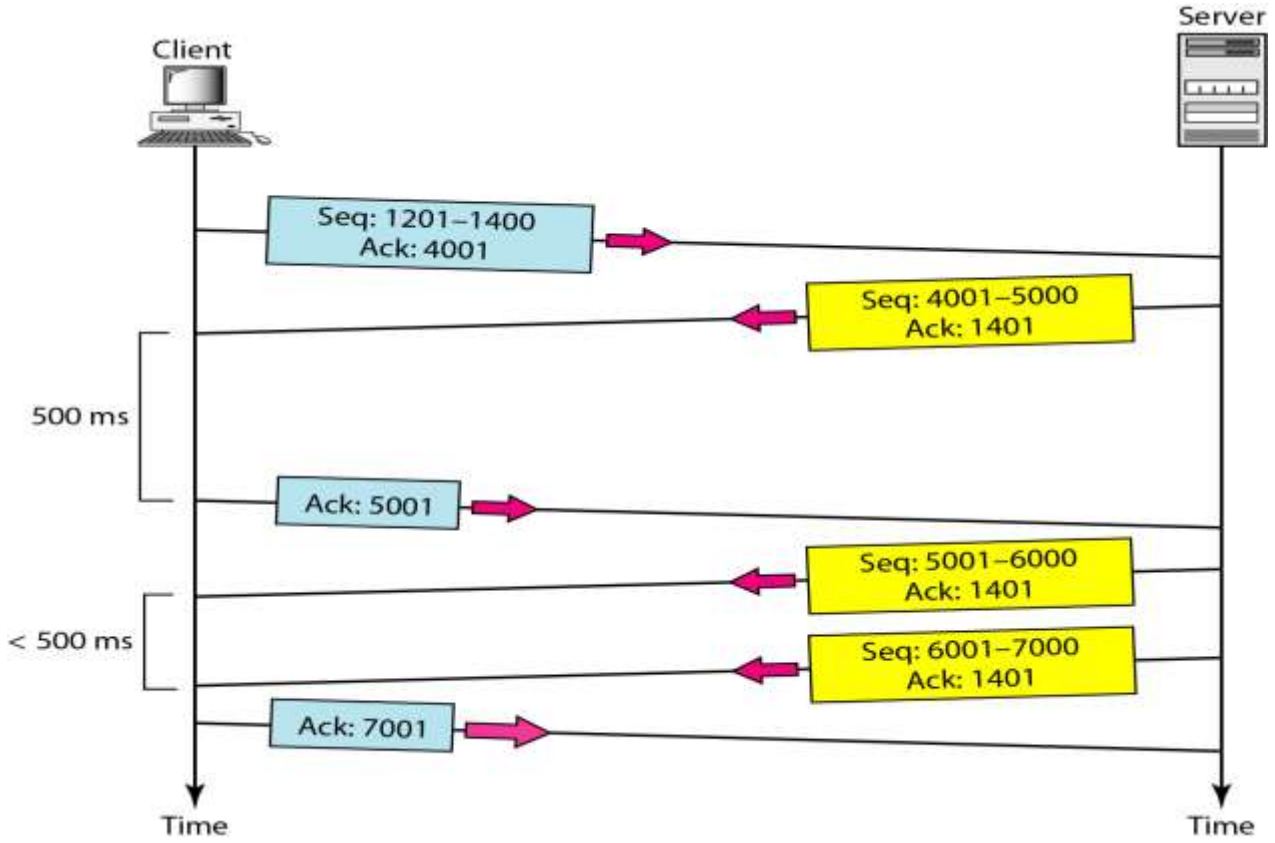


Figure 23.25 *Lost segment*

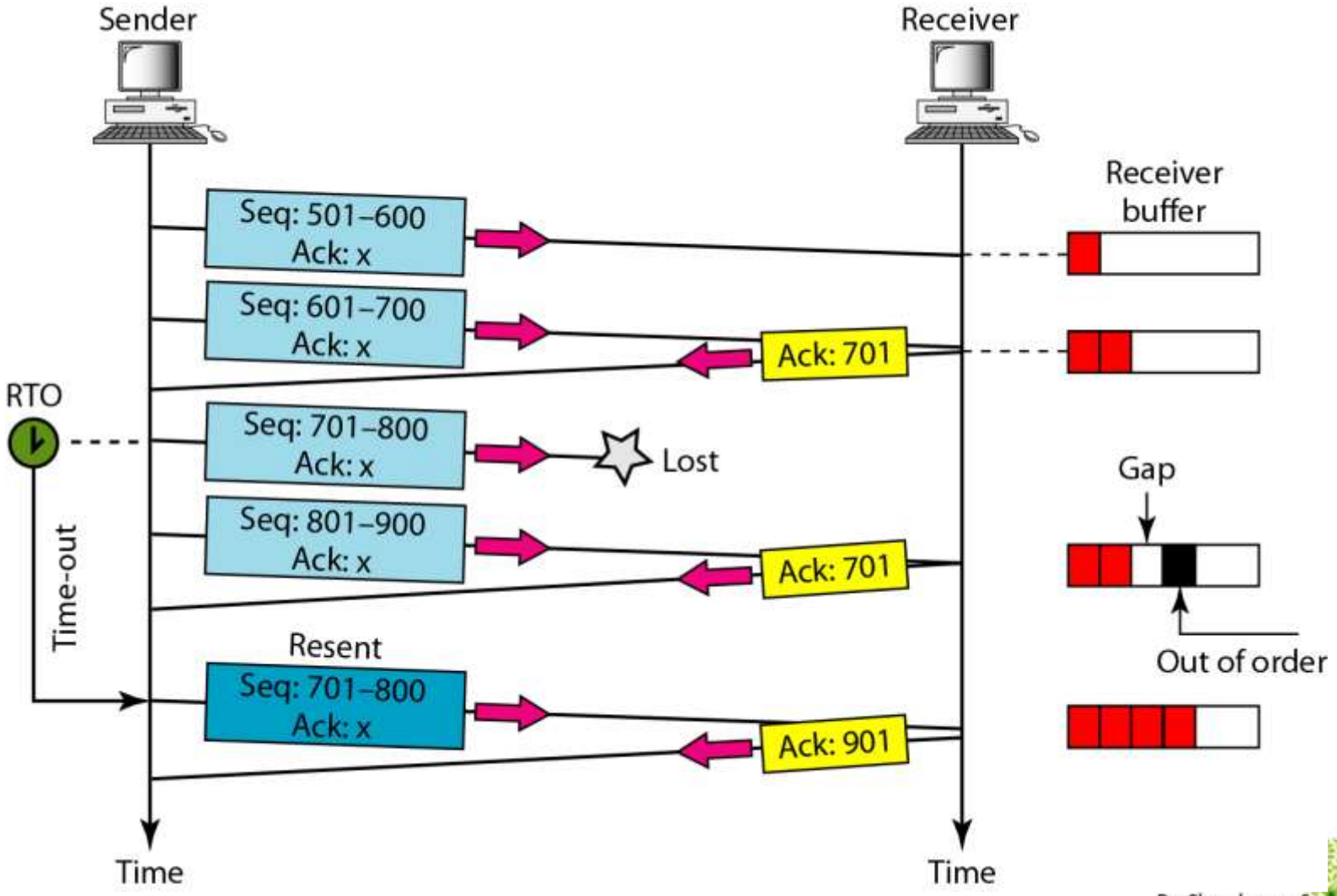
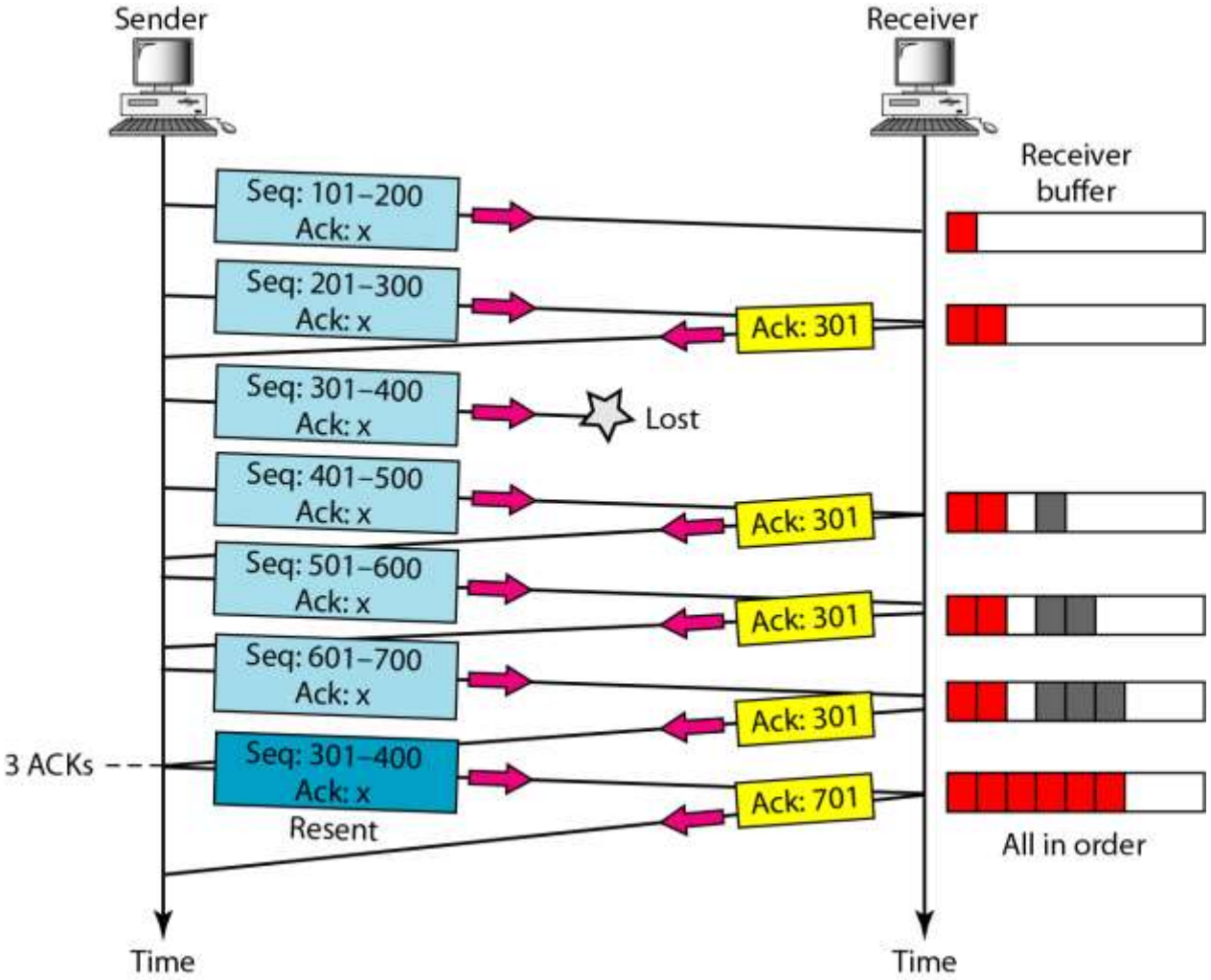


Figure 23.26 *Fast retransmission*



23-4 SCTP

Stream Control Transmission Protocol (SCTP) is a new reliable, message-oriented transport layer protocol. SCTP, however, is mostly designed for Internet applications that have recently been introduced. These new applications need a more sophisticated service than TCP can provide.

Topics discussed in this section:

SCTP Services and Features

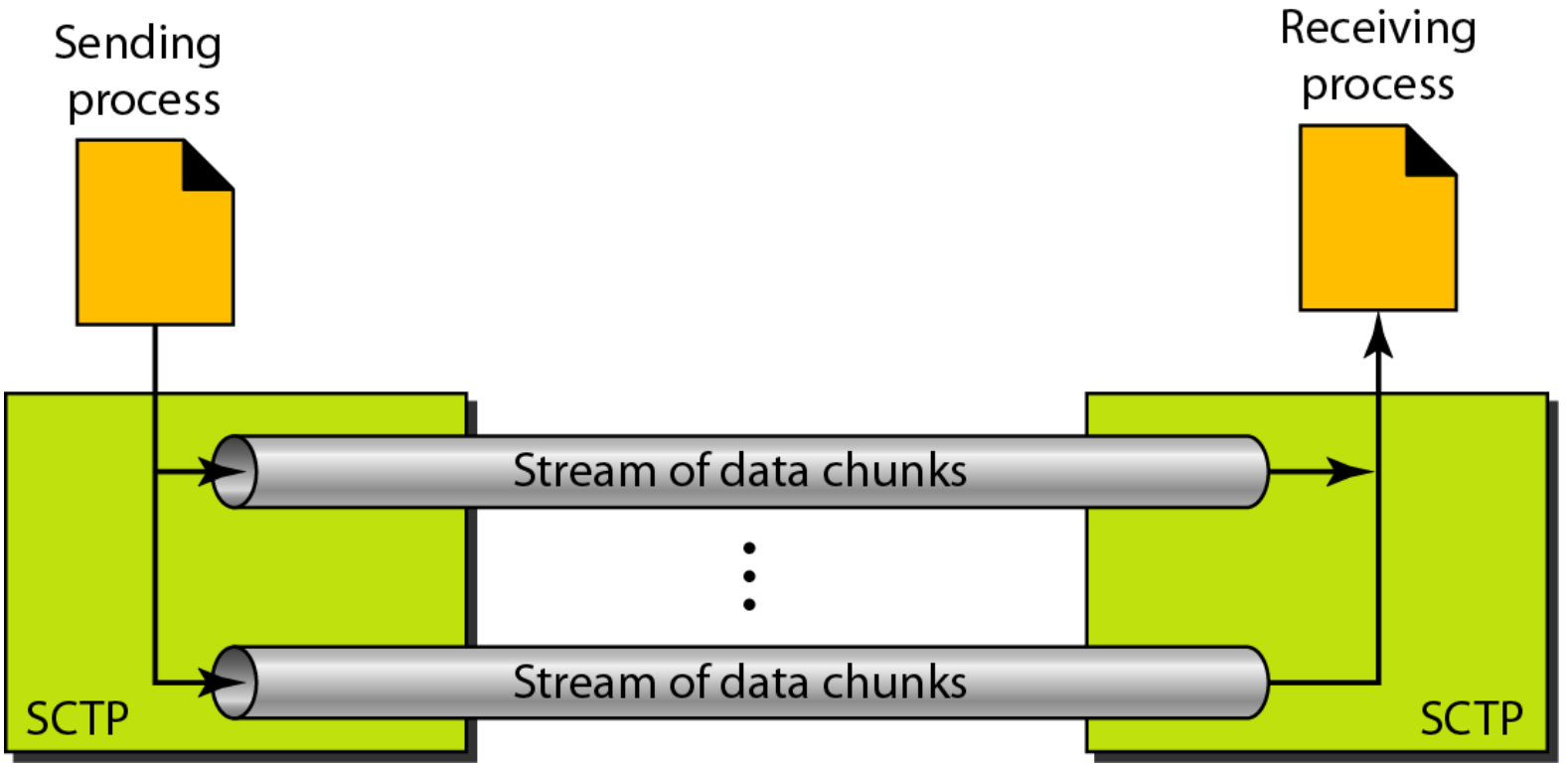
SCTP Services

1. *Process-to-Process Communication*
2. *Multiple Streams*
3. *Multihoming*
4. *Full-Duplex Communication*
5. *Connection-Oriented Service*
6. *Reliable Service*

Table 23.4 ***Some SCTP applications***

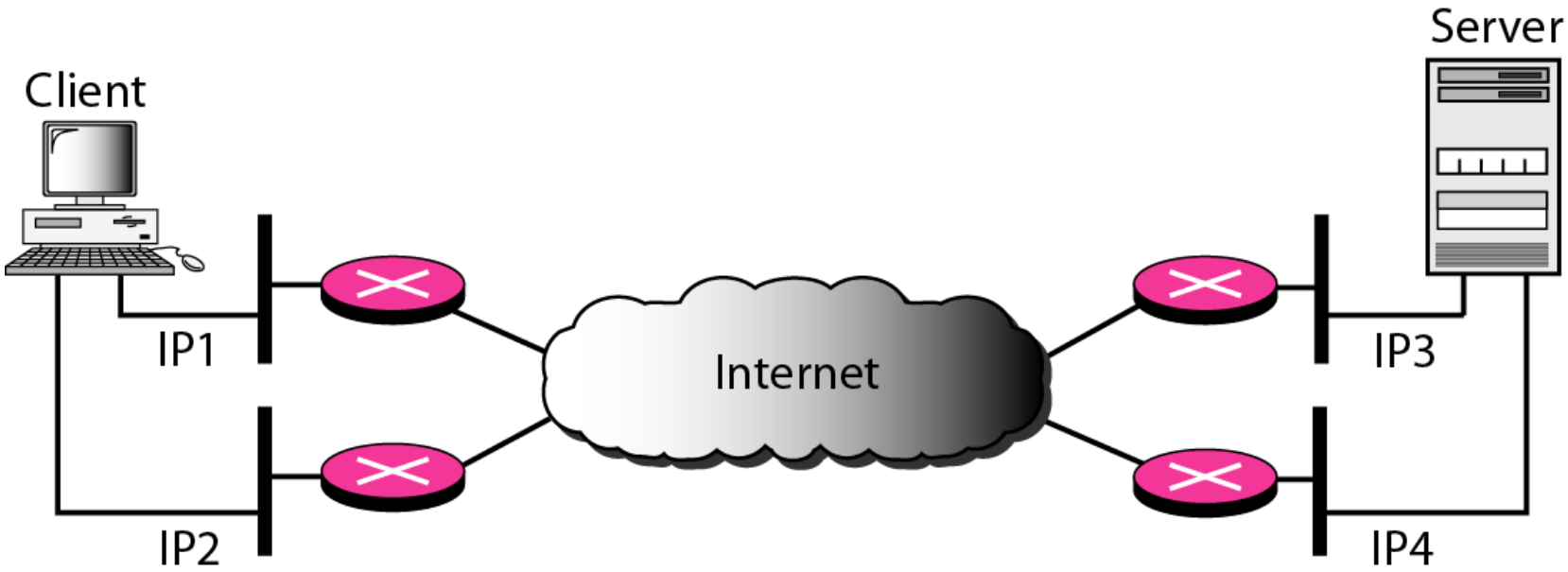
Protocol	*Port Number*	*Description*
IUA	9990	ISDN over IP
M2UA	2904	SS7 telephony signaling
M3UA	2905	SS7 telephony signaling
H.248	2945	Media gateway control
H.323	1718, 1719, 1720, 11720	IP telephony
SIP	5060	IP telephony

2. **Multiple Streams** -If one of the streams is blocked, the other streams can still deliver their data. The idea is similar to multiple lanes on a highway



Multihoming: The client and the server can make an association, using four different pairs of IP addresses, only one pair of IP addresses can be chosen for normal communication; the alternative is used if the main choice fails

Figure 23.28 ***Multihoming concept***

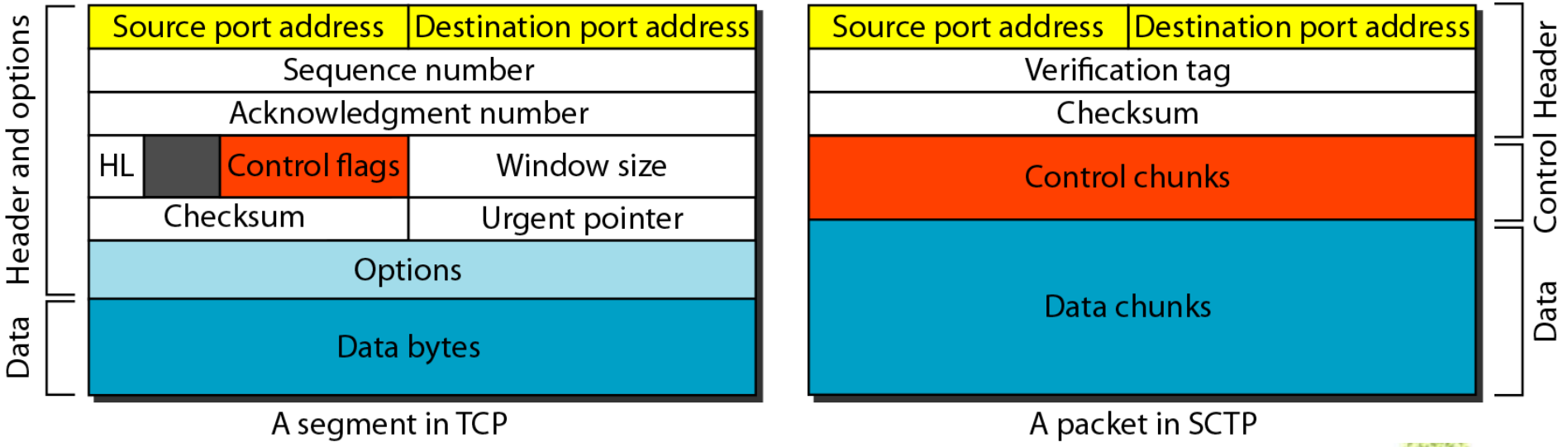


SCTP Features

1. *Transmission Sequence Number*
2. *Stream Identifier*
3. *Stream Sequence Number*
4. *Packets*
5. *Acknowledgment Number*
6. *Flow Control*
7. *Error Control*
8. *Congestion Control*

- In SCTP, a data chunk is numbered using a TSN- 0 and 2 32 - 1.
- In SCTP, there may be several streams in each association. Each stream in SCTP needs to be identified by using a stream identifier (SI)- 16-bit number starting from 0.
- In addition to an SI, SCTP defines each data chunk in each stream with a stream sequence number (SSN).
- In SCTP, acknowledgment numbers are used to acknowledge only data chunks.
- packets

Figure 23.29 ***Comparison between a TCP segment and an SCTP packet***

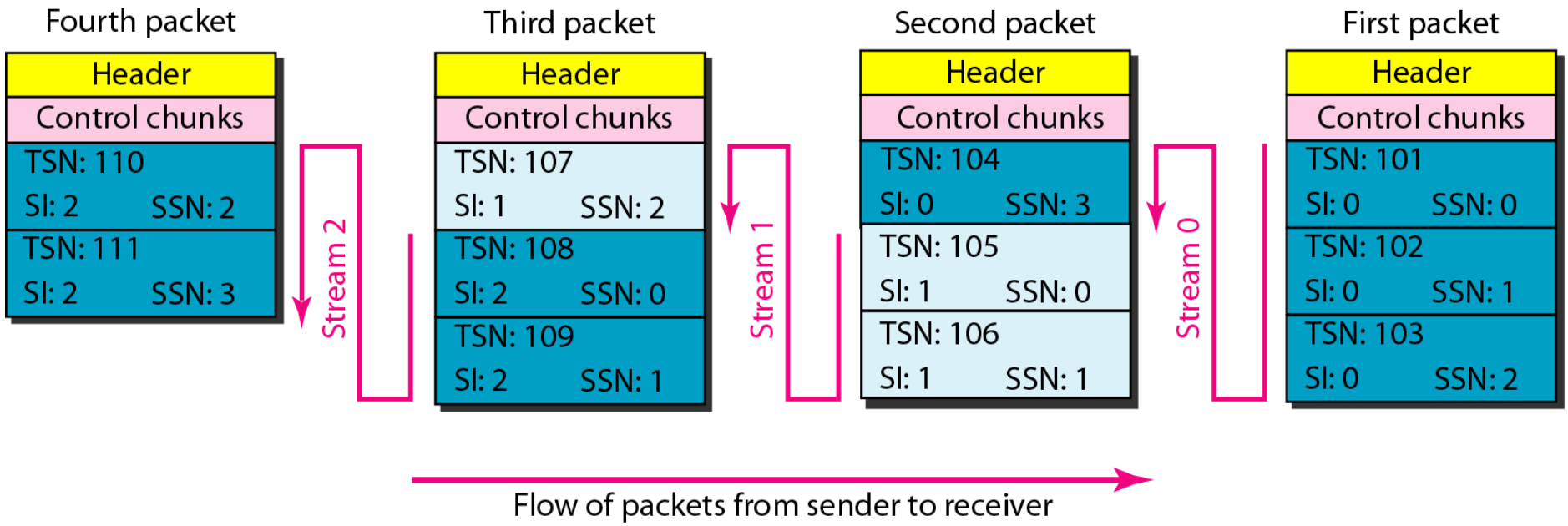


1. The control information in TCP is part of the header; the control information in SCTP is included in the control chunks
2. The data in a TCP segment treated as one entity; an SCTP packet can carry several data chunks; each can belong to a different stream
3. The mandatory part of the TCP header is 20 bytes, while the general header in SCTP is only 12 bytes
4. The options section, which can be part of a TCP segment, does not exist in an SCTP packet
5. The checksum in TCP is 16 bits; in SCTP, it is 32 bits
6. The verification tag in SCTP is an association identifier, which does not exist in TCP
7. TCP includes one sequence number in the header, which defines the number of the first byte in the data section. An SCTP packet can include several different data chunks. TSNs, SIs, and SSNs define each data chunk.
8. TCP that carry control information (such as SYN and FIN) need to consume one sequence number; control chunks in SCTP never use a TSN, SI, or SSN

Note

In SCTP, control information and data information are carried in separate chunks.

Figure 23.30 *Packet, data chunks, and streams*



Chapter 24

Congestion Control and Quality of Service

Dr. Chandrappa S
NAGARJUNA
COLLEGE OF ENGINEERING & TECHNOLOGY

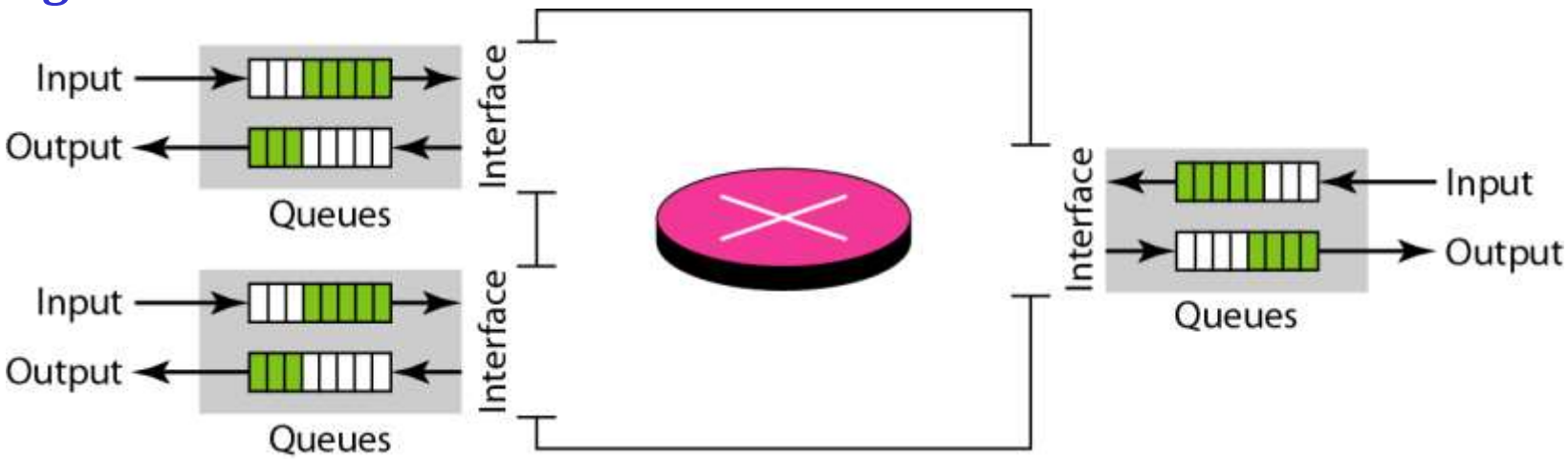
24-2 CONGESTION

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

Figure 24.3 ***Queues in a router***

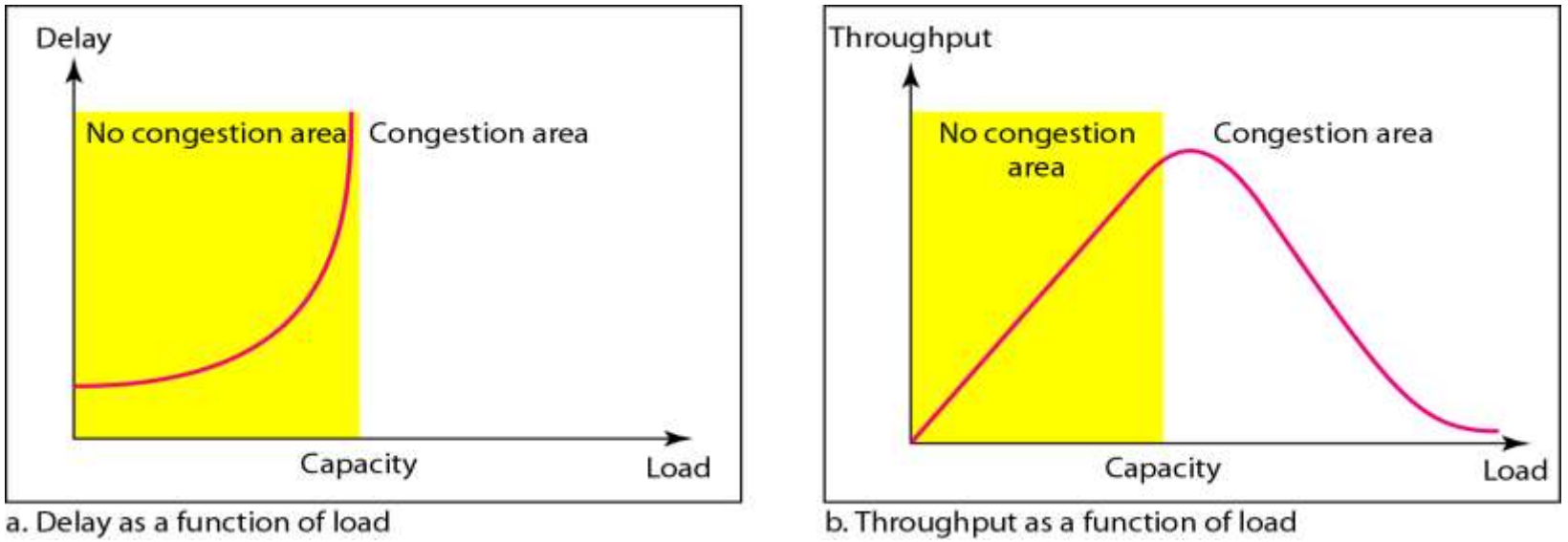


Congestion in a network or internetwork occurs because routers and switches have queues-buffers that hold the packets before and after processing

1. The packet is put at the end of the input queue while waiting to be checked.
2. The processing module of the router removes the packet from the input queue once it reaches the front of the queue and uses its routing table and the destination address to find the route.
3. The packet is put in the appropriate output queue and waits its tum to be sent.

Network Performance

Figure *Packet delay and throughput as functions of load*



a. Delay as a function of load

b. Throughput as a function of load

- Congestion control involves two factors that measure the performance of a network: *delay* and *throughput.*
- **Throughput: This refers to the amount of data that can be transmitted successfully over the network in a given period of time. It's usually measured in bits per second (bps) or multiples thereof (kbps, Mbps, etc.)**
- **Delay: This is the amount of time it takes for a packet of data to travel from the source to the destination. Delay is usually measured in milliseconds (ms).**

24-3 CONGESTION CONTROL

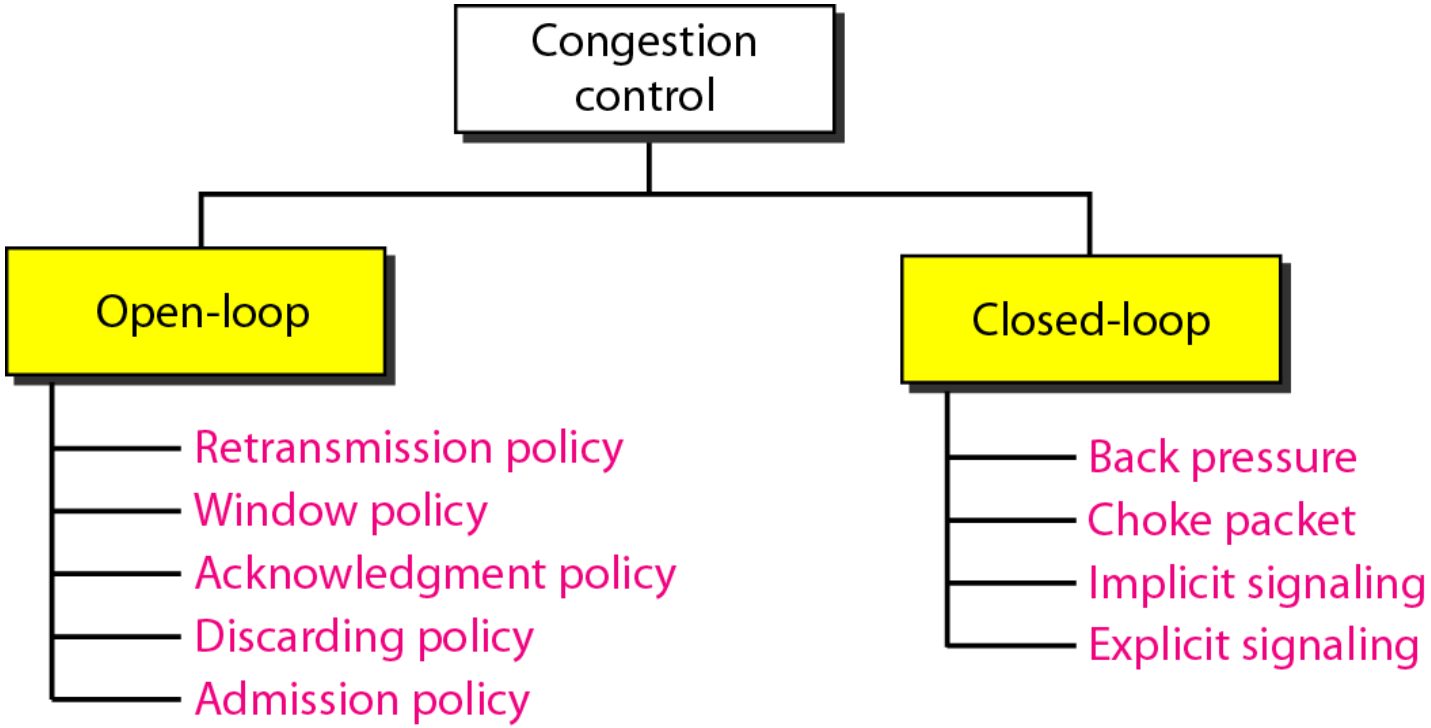
Congestion control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened. In general, we can divide congestion control mechanisms into two broad categories: open-loop congestion control (prevention) and closed-loop congestion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

Topics discussed in this section:

Open-Loop Congestion Control
Closed-Loop Congestion Control

Dr. Chandrappa S
NAGARJUNA
College of Engineering & Technology

Figure 24.5 *Congestion control categories*



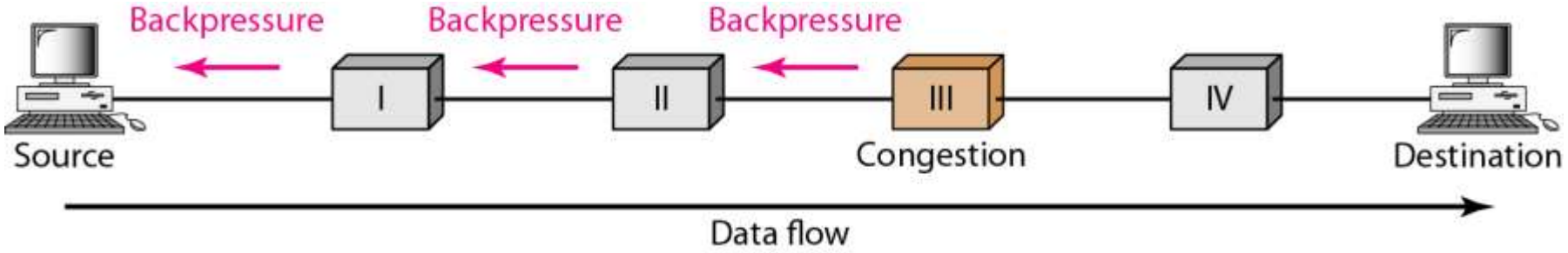
In open-loop congestion control, policies are applied to prevent congestion before it happens. In these mechanisms, congestion control is handled by either the source or the destination.

1. The retransmission policy and the retransmission timers must be designed to optimize efficiency and at the same time prevent congestion
2. The Selective Repeat window is better than the Go-Back-N window for congestion control
3. Sending fewer acknowledgments means imposing less load on the network
4. Discard less sensitive packets when congestion is likely to happen
5. An admission policy, which is a quality-of-service mechanism, Switches in a flow first check the resource requirement of a flow before admitting it to the network

Closed-loop congestion control mechanisms try to alleviate congestion after it happens.

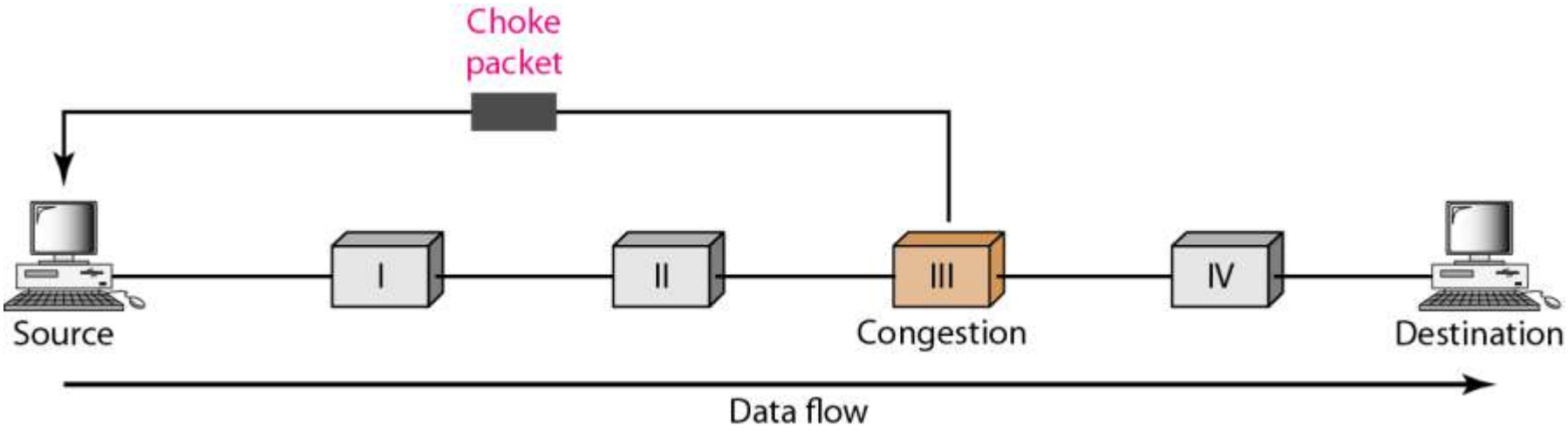
1. *Backpressure-* Backpressure is a node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow, to the source.
2. *choke packet method*, the warning is from the router, which has encountered congestion, to the source station directly.
3. *implicit signaling,* there is no communication between the congested node or nodes and the source. The source guesses that there is a congestion somewhere in the network from other symptoms (ex. delay in receiving an acknowledgment).
4. *explicit signaling method,* the signal is included in the packets that carry data.
   - **Backward Signaling:** A bit can be set in a packet moving in the direction opposite to the congestion. This bit can warn the source
   - **Forward Signaling** A bit can be set in a packet moving in the direction of the congestion. This bit can warn the destination

Figure 24.6 *Backpressure method for alleviating congestion*



Node III in the figure has more input data than it can handle. It drops some packets in its input buffer and informs node II to slow down. Node II, in turn, may be congested because it is slowing down the output flow of data. If node II is congested, it informs node I to slow down, which in turn may create congestion. If so, node I informs the source of data to slow down.

Figure 24.7 *Choke packet*



24-5 QUALITY OF SERVICE

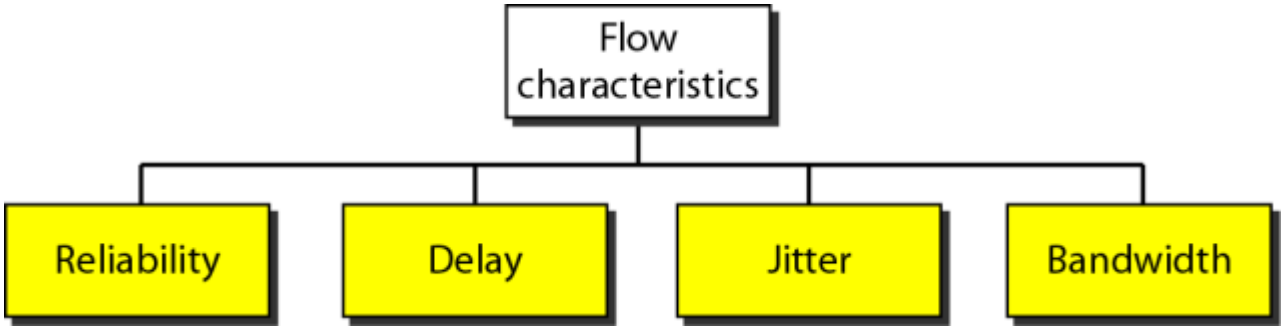
Quality of Service (QoS) refers to how well a network can provide the needed performance for different types of data. It focuses on ensuring that data flows (like video calls, online gaming, or file transfers) get the right treatment in terms of speed, reliability, and priority.

Topics discussed in this section:

Flow Characteristics

Flow Classes

Figure 24.15 *Flow characteristics*



- **Reliability: This measures how consistently data is transmitted without errors or loss. A reliable network ensures that packets of data arrive intact and in the correct order.**
- **Delay: Delay refers to the time it takes for data to travel from the source to the destination. This can be influenced by factors such as the network's length, routing, and congestion.**
- **Jitter: Jitter is the variation in delay. It occurs when packets experience different delays along their route.**
- **Bandwidth: Bandwidth is the maximum rate at which data can be transmitted across a network, often measured in bits per second (bps). High bandwidth means more data can be transferred quickly,**

Flow Classes-

a. ***Constant Bit Rate (CBR)***.
b. ***Variable Bit Rate-Non Real Time (VBR-NRT)***.
c. ***Variable Bit Rate-Real Time (VBR-RT)***.
d. ***Available Bit Rate (ABR)***.
e. ***Unspecified Bit Rate (UBR)***.

24-6 TECHNIQUES TO IMPROVE QoS

In Section 24.5 we tried to define QoS in terms of its characteristics. In this section, we discuss some techniques that can be used to improve the quality of service. We briefly discuss four common methods: scheduling, traffic shaping, admission control, and resource reservation.

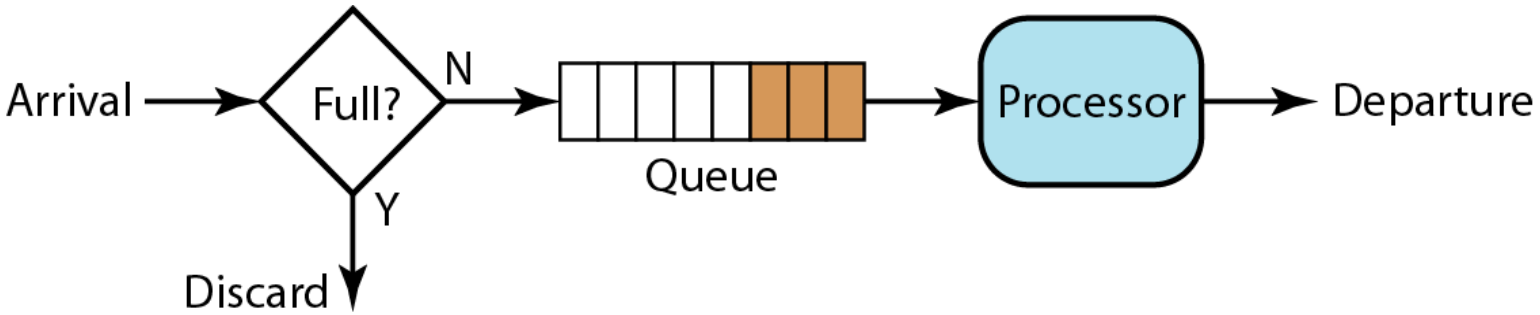
Topics discussed in this section:

Scheduling
Traffic Shaping
Resource Reservation
Admission Control

Scheduling-Scheduling refers to how a network allocates resources (like bandwidth) to different data flows or packets over time

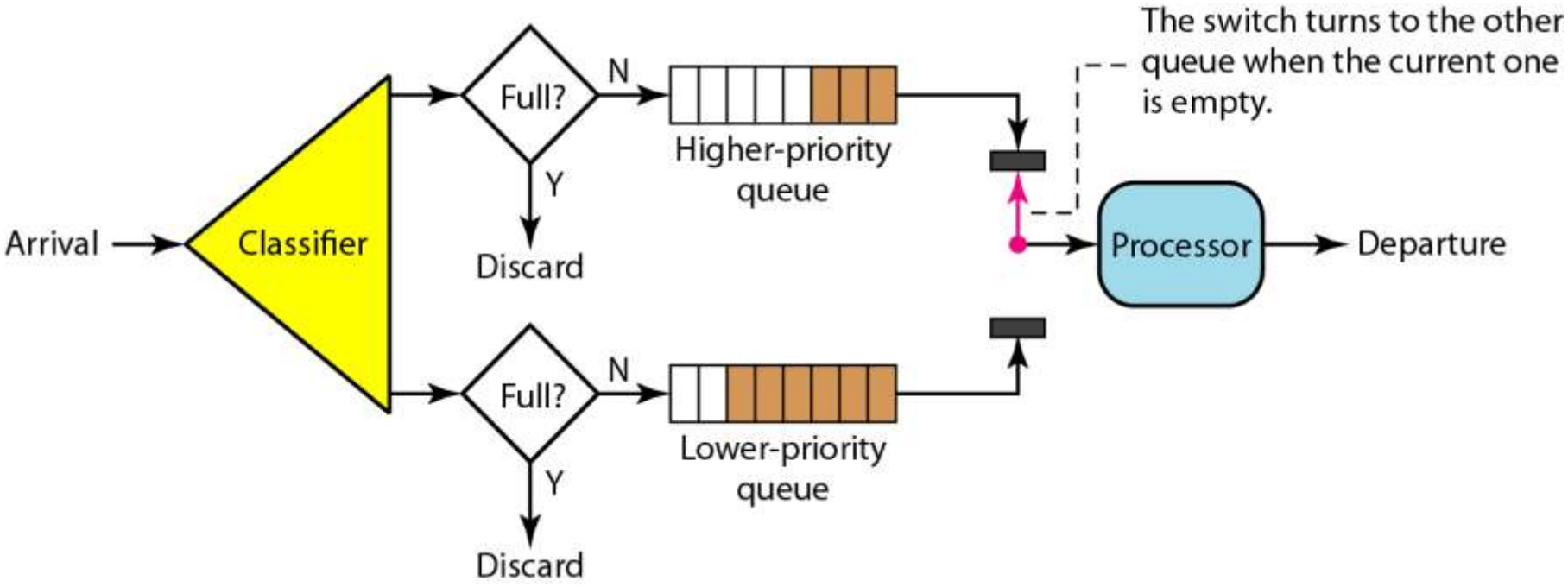
1. FIFO Queuing

In first-in, first-out (FIFO) queuing, packets wait in a buffer (queue) until the node (router or switch) is ready to process them. If the average arrival rate is higher than the average processing rate, the queue will fill up and new packets will be discarded



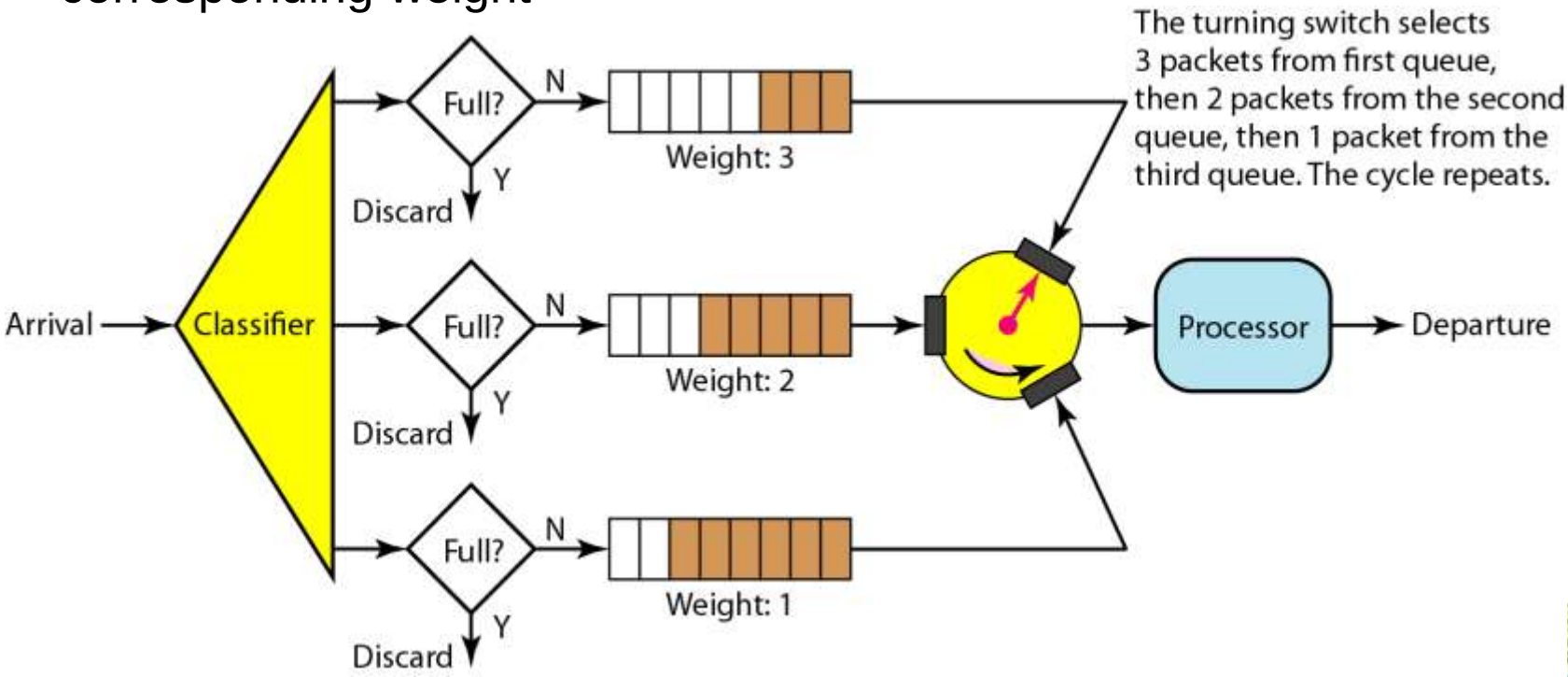
2.Priority Queuing

In **priority queuing,** packets are first assigned to a priority class. Each priority class has its own queue. The packets in the highest-priority queue are processed first. Packets in the lowest-priority queue are processed last



3. Weighted Fair Queuing

- the packets are still assigned to different classes and admitted to different queues.
- The queues, however, are weighted based on the priority of the queues; higher priority means a higher weight.
- The system processes packets in each queue in a round-robin fashion with the number of packets selected from each queue based on the corresponding weight



Traffic Shaping-is a network management technique used to control the flow of data into a network to ensure that traffic conforms to a specific rate or profile

1. Leaky Bucket

If a bucket has a small hole at the bottom, the water leaks from the bucket at a constant rate as long as there is water in the bucket. The rate at which the water leaks does not depend on the rate at which the water is input to the bucket unless the bucket is empty. The input rate can vary, but the output rate remains constant

Figure 24.19 *Leaky bucket*

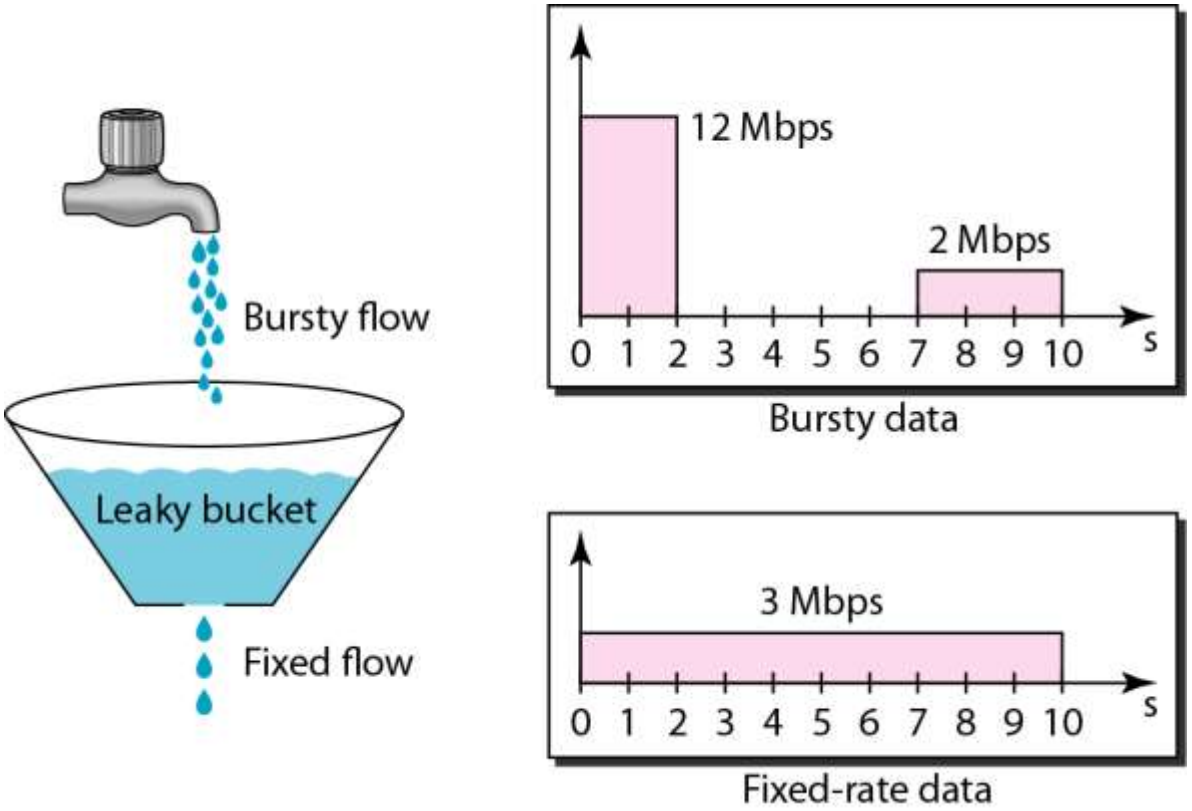
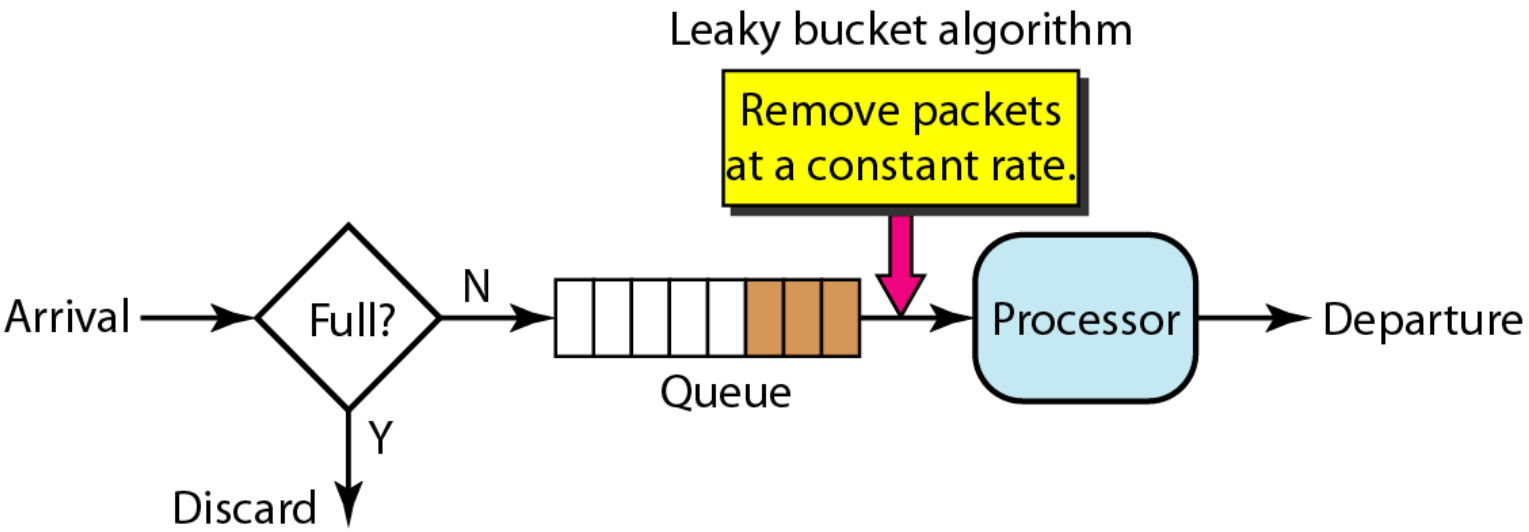


Figure 24.20 *Leaky bucket implementation*



Token bucket

1.Tokens in the Bucket:

- **The bucket accumulates tokens at a fixed rate, usually representing a specific amount of data (e.g., 1 token = 1 byte of data or 1 packet).**
- **If tokens are available in the bucket, data can be sent. If no tokens are available, the data must wait until more tokens are generated.**

2.Token Generation:

- **Tokens are generated at a fixed rate, such as 1 token per millisecond, up to a maximum capacity of the bucket (i.e., the maximum number of tokens the bucket can hold).**

3. Burst Traffic:

- **if no traffic is being sent for a while, tokens accumulate in the bucket. Once traffic resumes, the bucket can allow a burst (sending data faster than the average rate), as long as there are tokens available.**

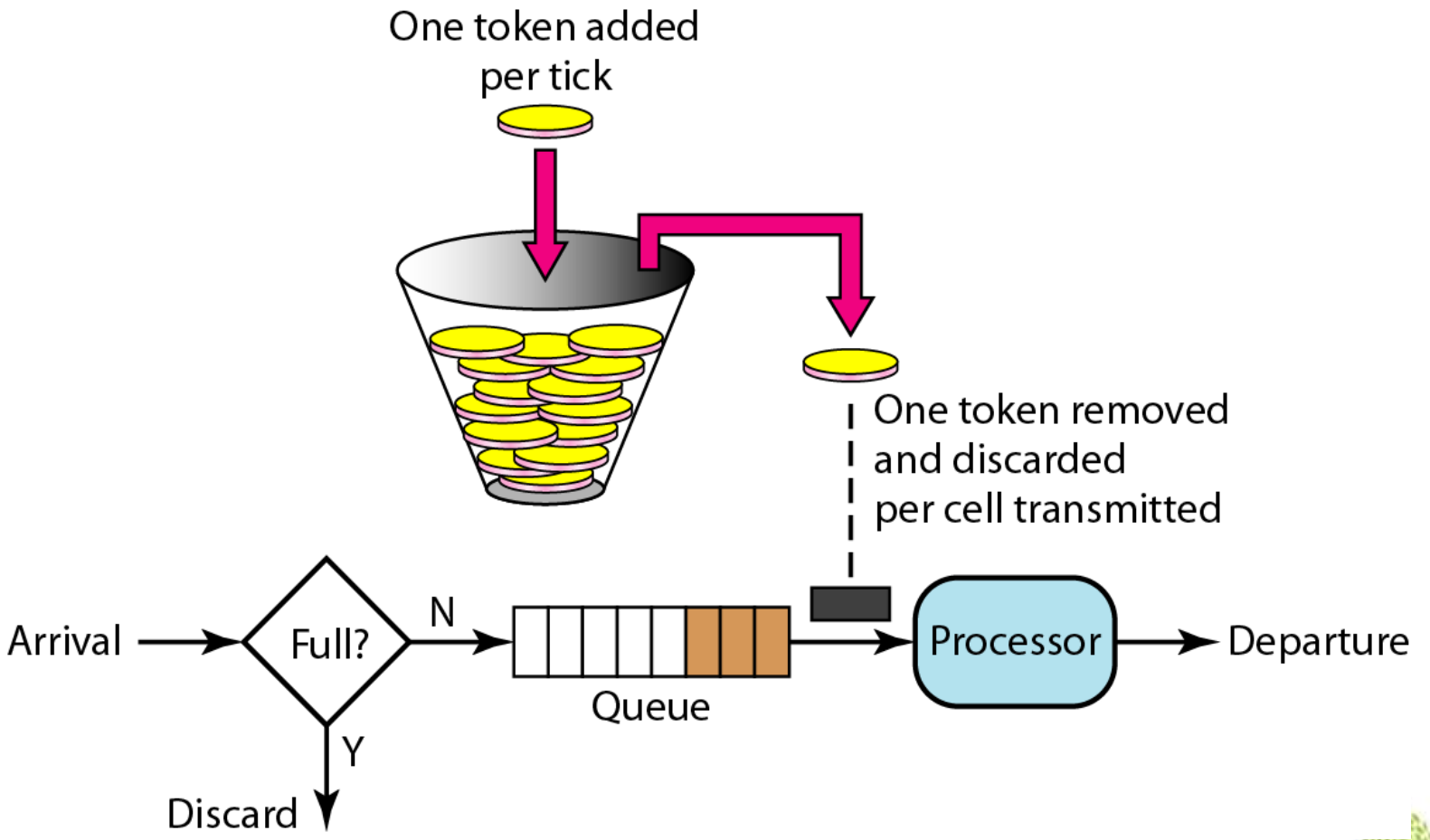
4. Sending Data:

- **When data needs to be sent, it checks whether there are enough tokens in the bucket.**
  - **If enough tokens are present, the data is sent, and the corresponding number of tokens are removed from the bucket.**
  - **If there aren't enough tokens, the data must wait until enough tokens are generated to allow transmission.**

5. Bucket Capacity:

- **The token bucket has a maximum capacity. If the bucket is full, no more tokens can be accumulated, and excess tokens are discarded.**

Figure 24.21 *Token bucket*



Resource Reservation-A flow of data needs resources such as a buffer, bandwidth, CPU time, and so on. These should be reserved.

Admission Control-Admission control refers to the mechanism used by a router or a switch to accept or reject a flow based on predefined parameters called *flow specifications*