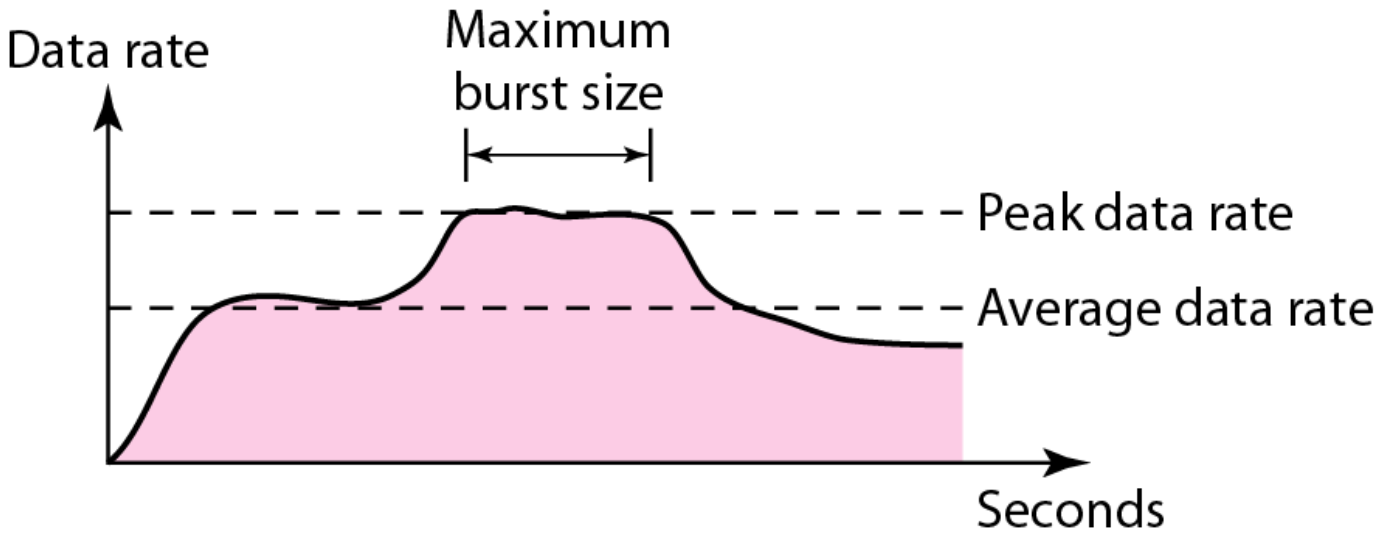
# Congestion Control
# And
# QoS

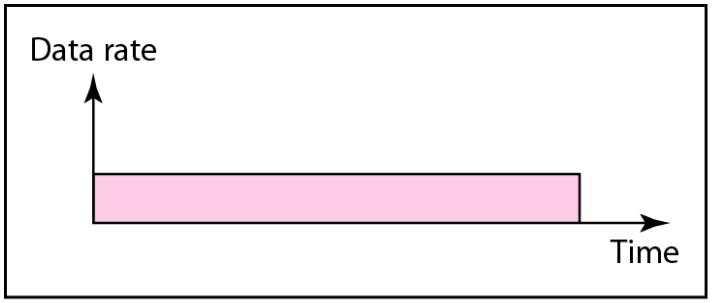
# 24-1 DATA TRAFFIC

*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.*

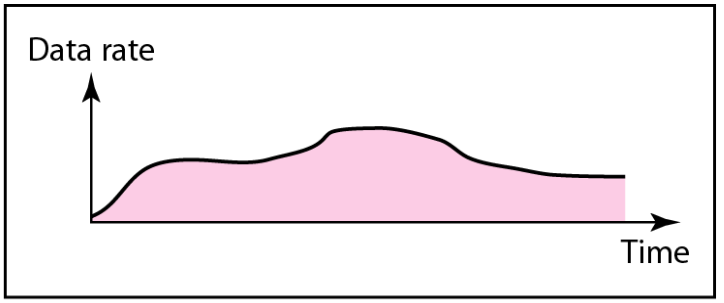
## Figure 24.1 *Traffic descriptors*



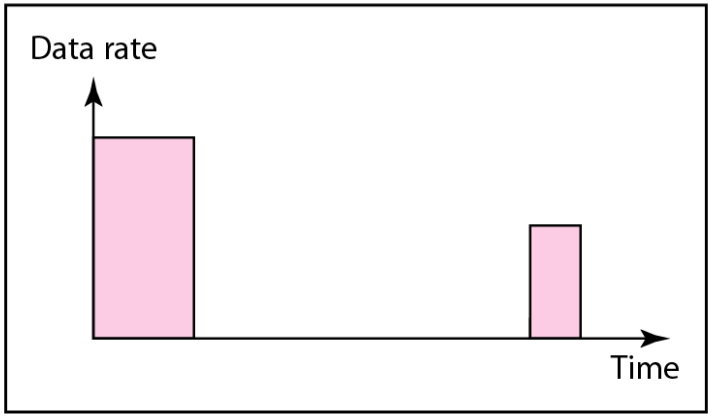
# Figure 24.2 *Three traffic profiles*




a. Constant bit rate




b. Variable bit rate




c. Bursty

# 24-2 CONGESTION

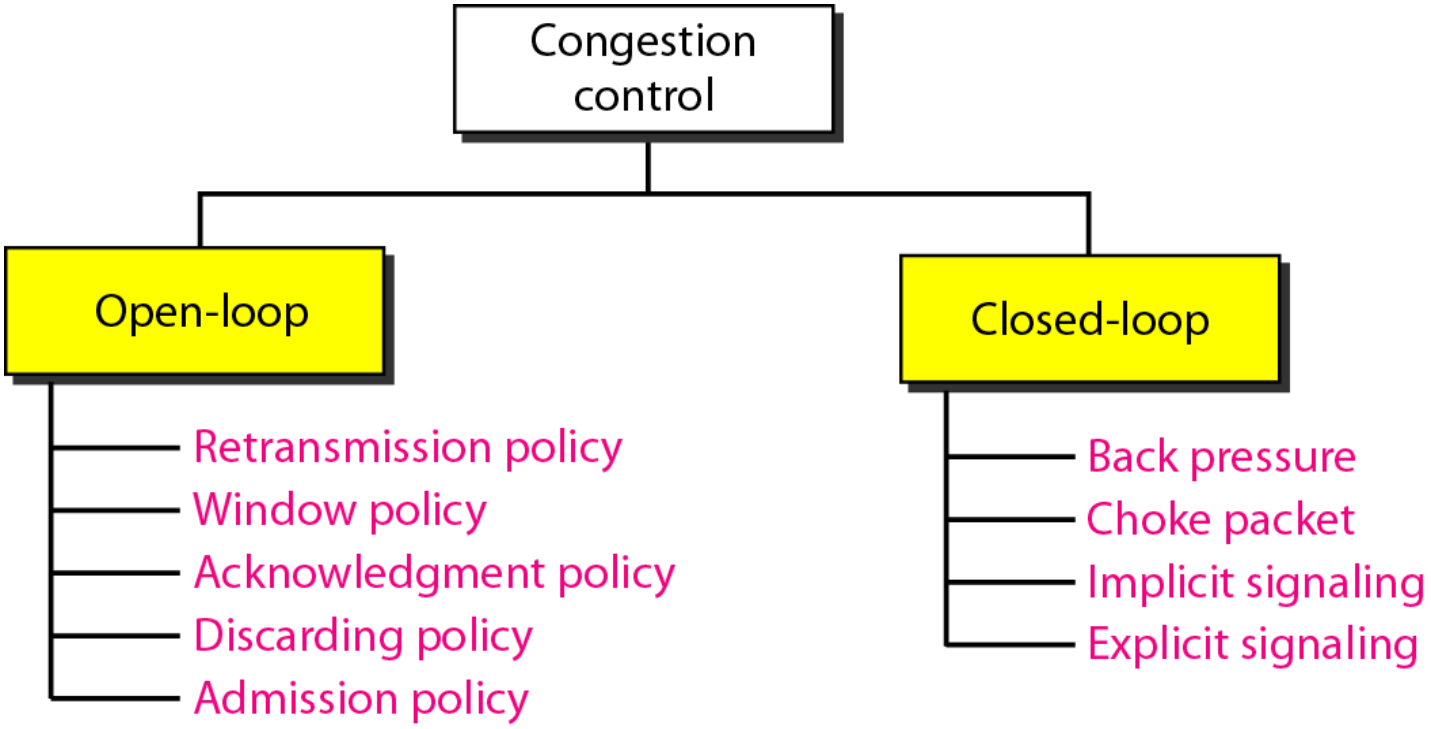
***Congestion in a network may occur if the load on the network—the number of packets sent to the network—is greater than the capacity of the network—the number of packets a network can handle. Congestion control refers to the mechanisms and techniques to control the congestion and keep the load below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

# 24-3 CONGESTION CONTROL

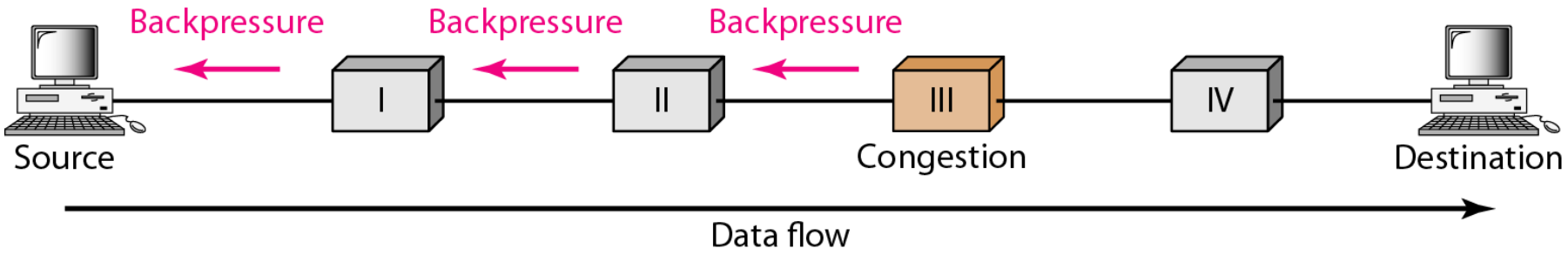
*Congestion control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened. In general, we can divide congestion control mechanisms into two broad categories: open-loop congestion control (prevention) and closed-loop congestion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

**Open-Loop Congestion Control**
**Closed-Loop Congestion Control**

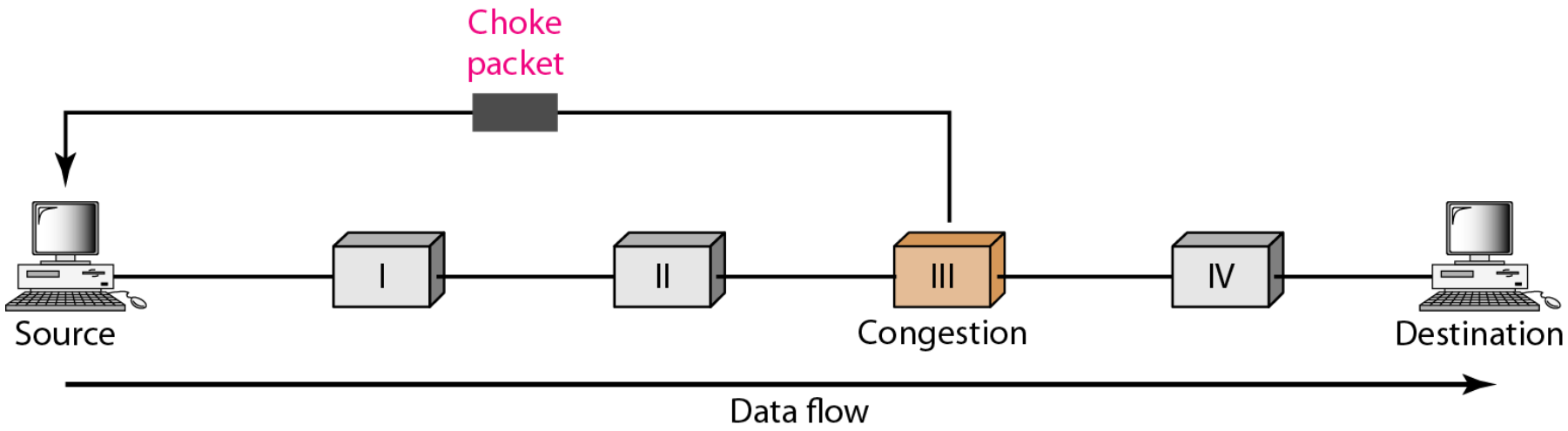
## Figure 24.5 *Congestion control categories*



## Figure 24.6 *Backpressure method for alleviating congestion*



## Figure 24.7 *Choke packet*



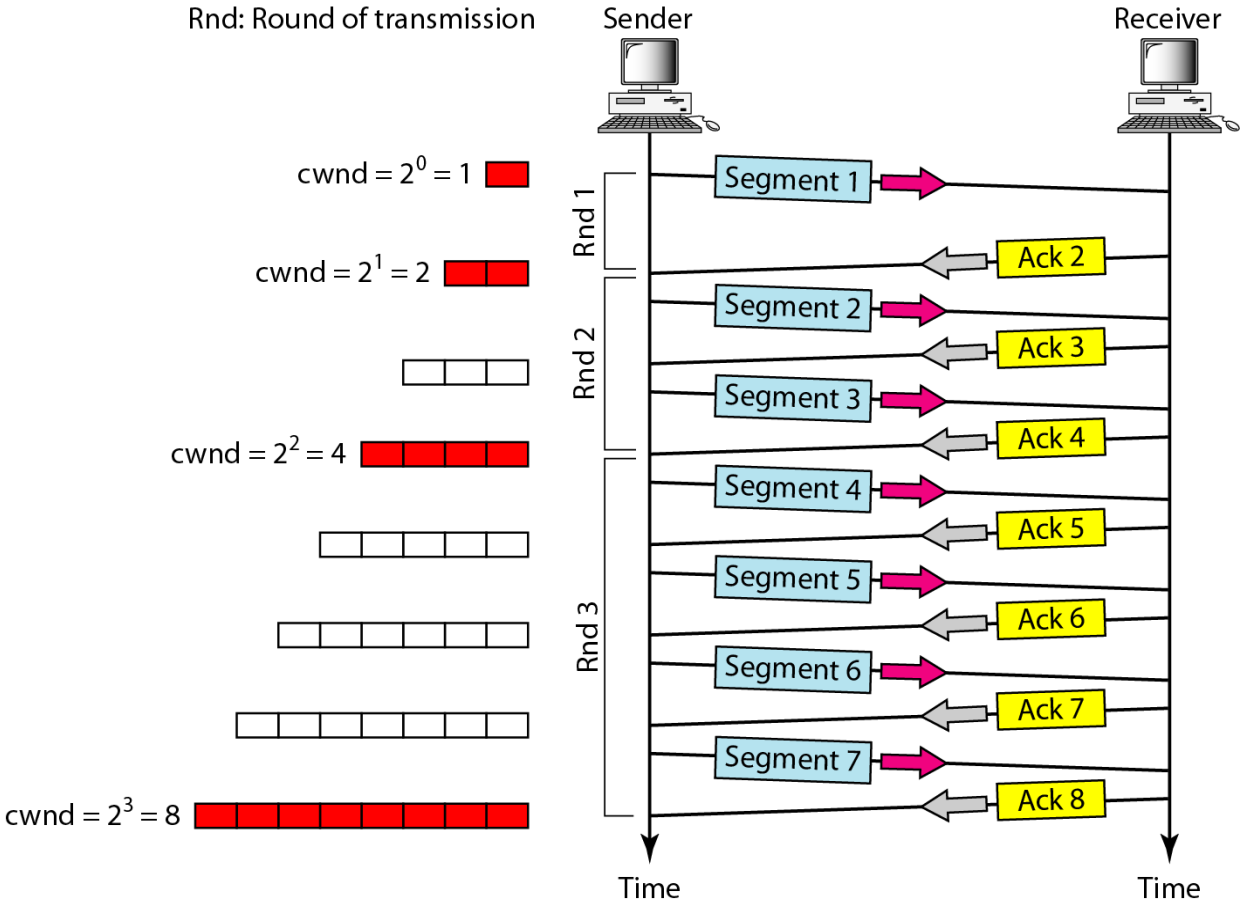
# 24-4 TWO EXAMPLES

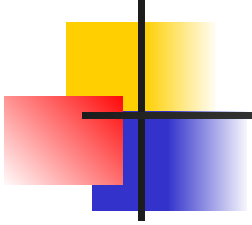
*To better understand the concept of congestion control, let us give two examples: one in TCP and the other in Frame Relay.*

***Topics discussed in this section:***

**Congestion Control in TCP**
**Congestion Control in Frame Relay**

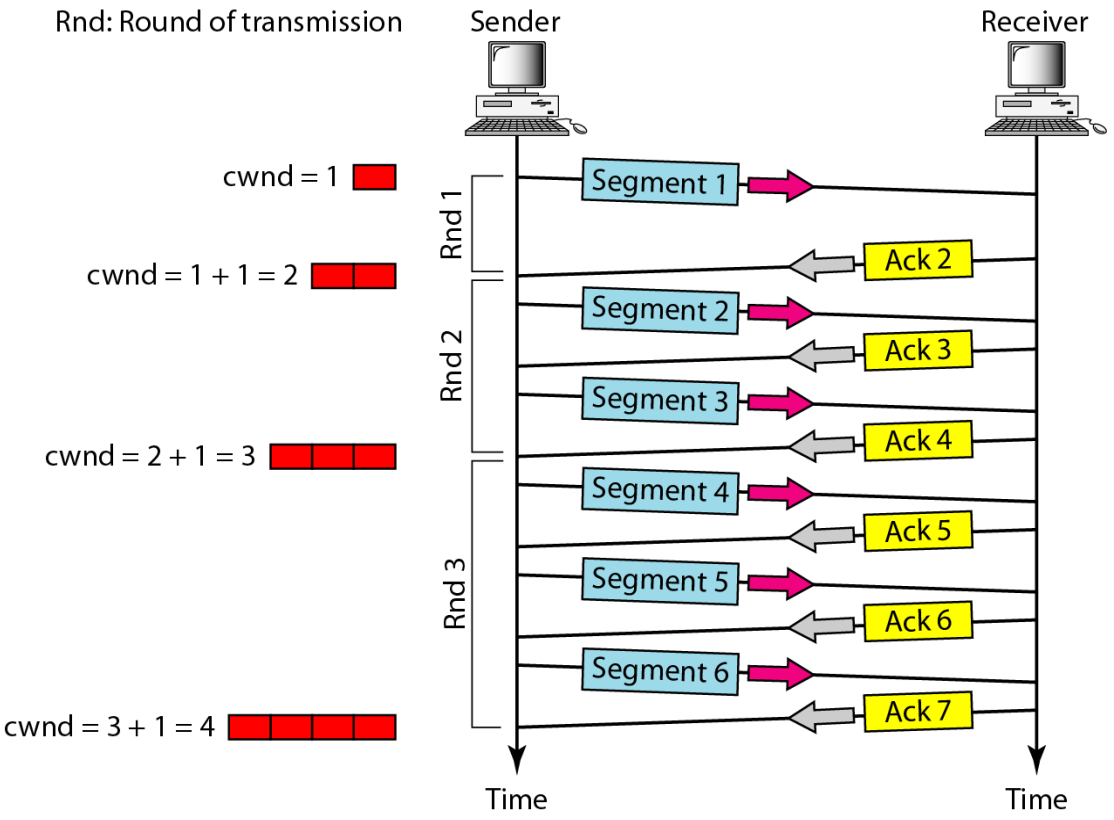
# Figure 24.8 *Slow start, exponential increase*

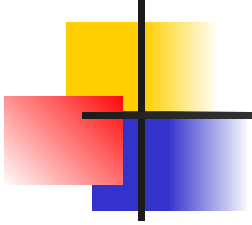




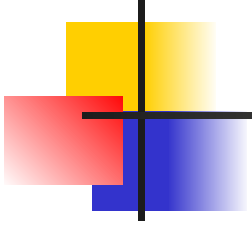
**In the slow-start algorithm, the size of the congestion window increases exponentially until it reaches a threshold.**

# Figure 24.9 *Congestion avoidance, additive increase*





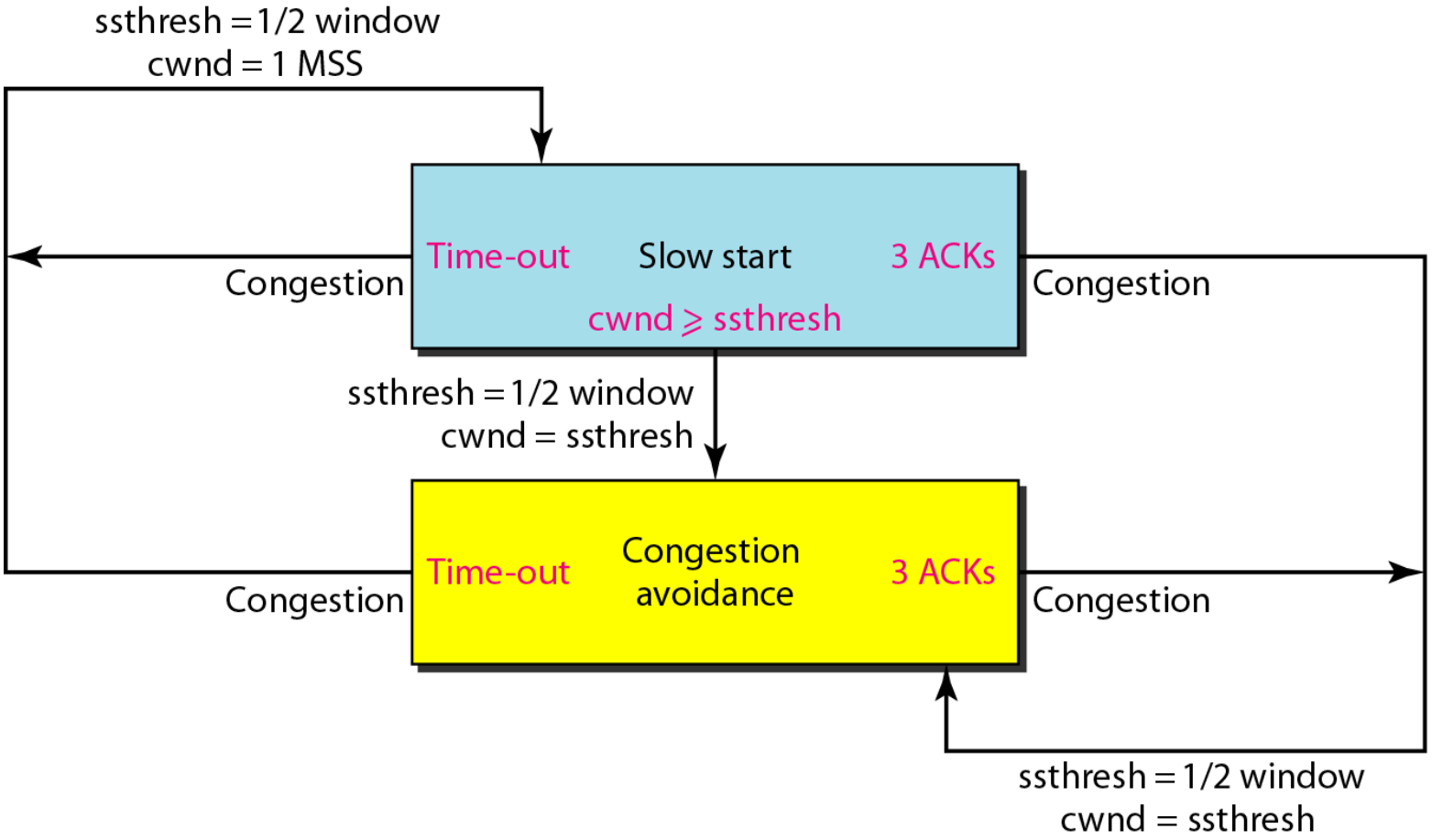
**In the congestion avoidance algorithm, the size of the congestion window increases additively until congestion is detected.**



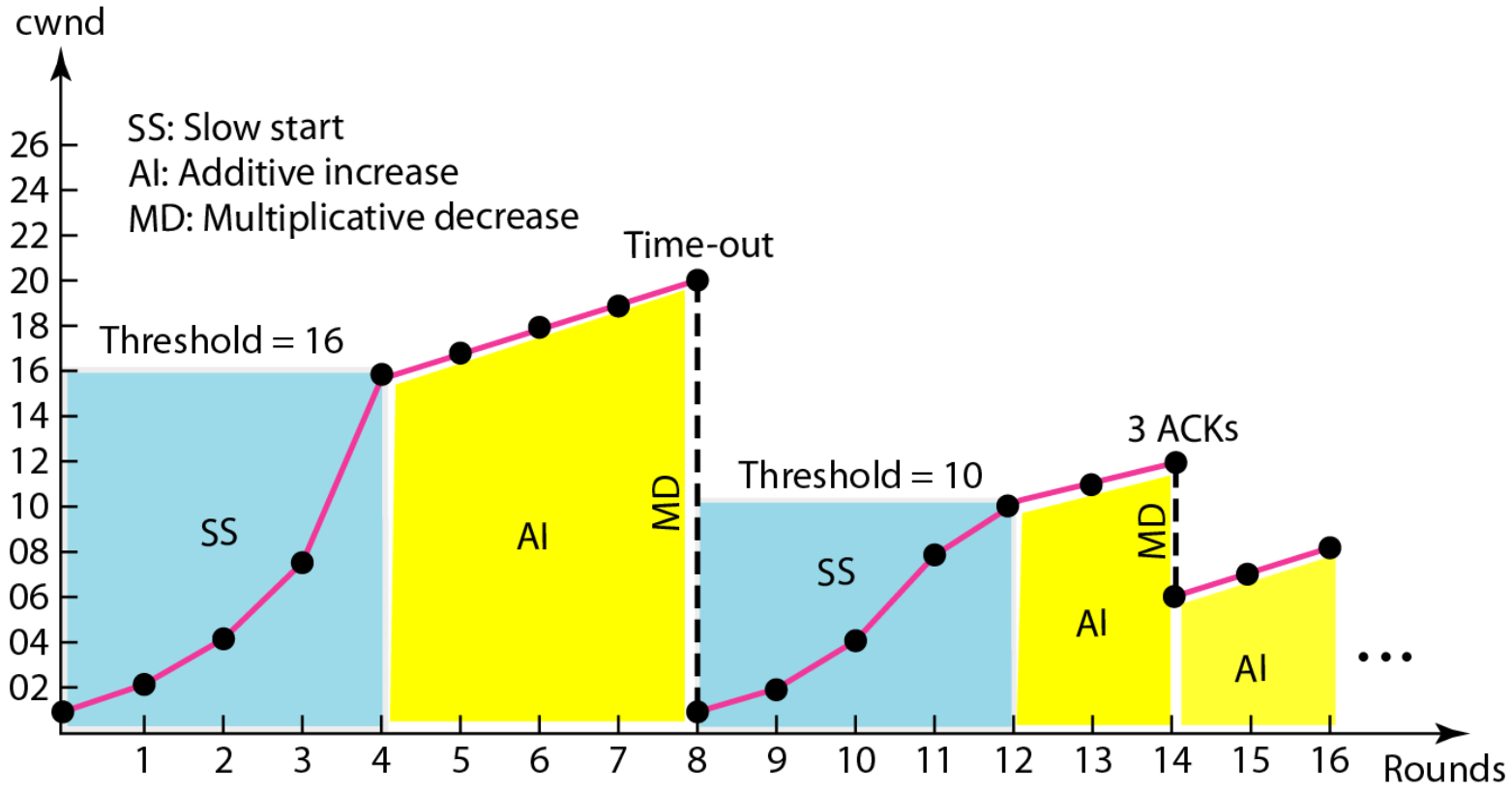
An implementation reacts to congestion detection in one of the following ways:

- If detection is by time-out, a new slow start phase starts.
- If detection is by three ACKs, a new congestion avoidance phase starts.

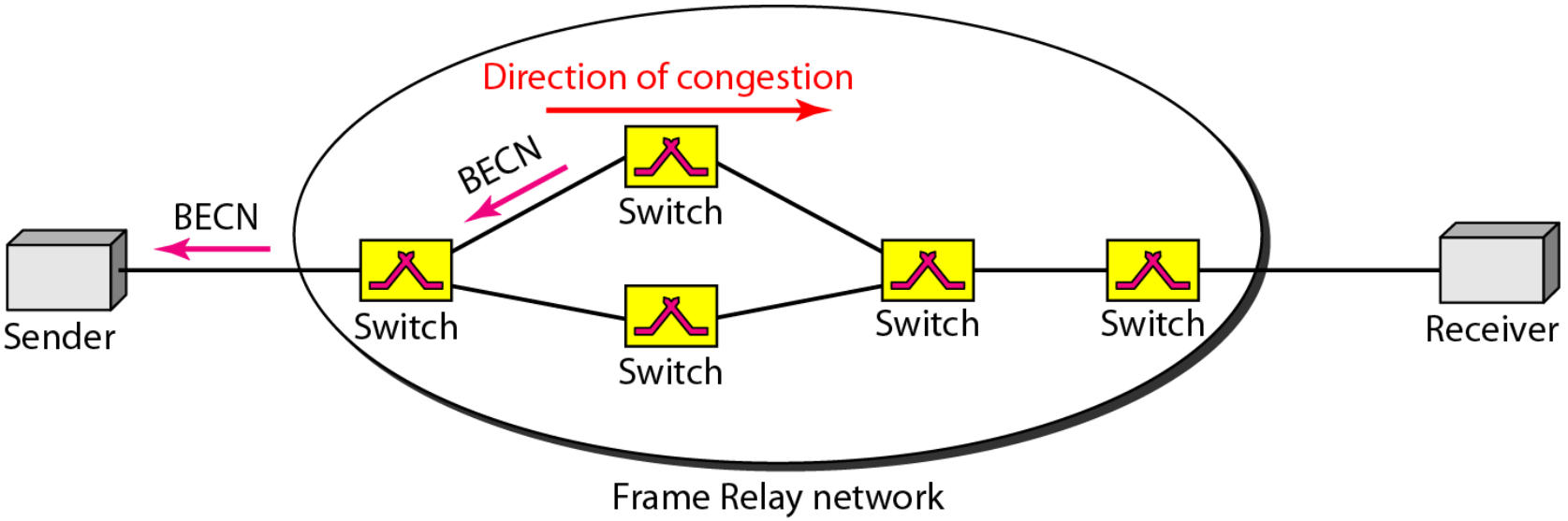
## Figure 24.10 *TCP congestion policy summary*



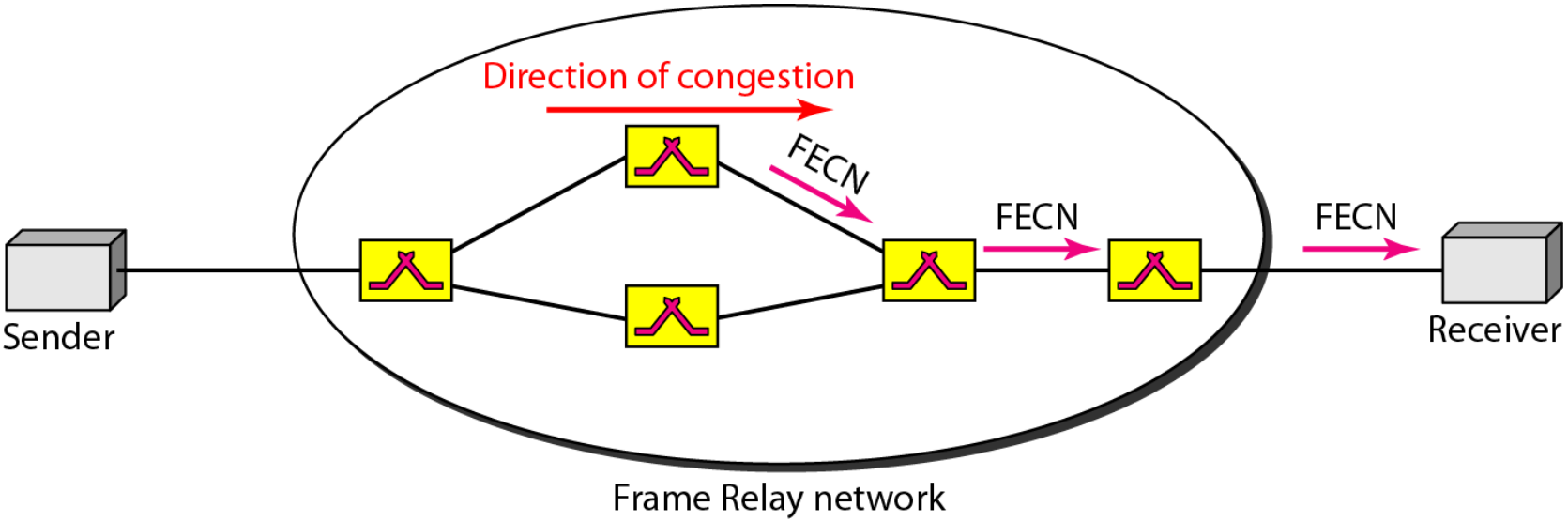
## Figure 24.11 *Congestion example*



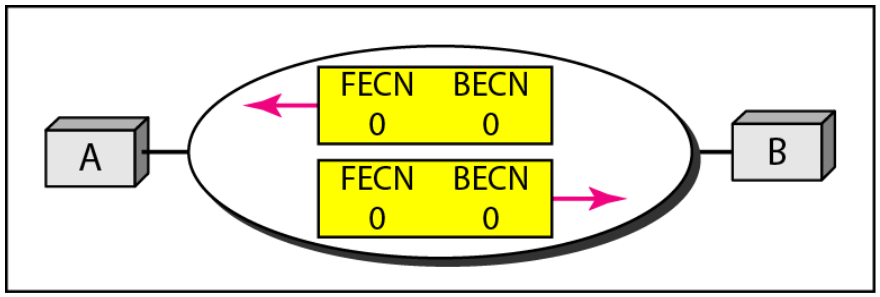
# Figure 24.12 *BECN*



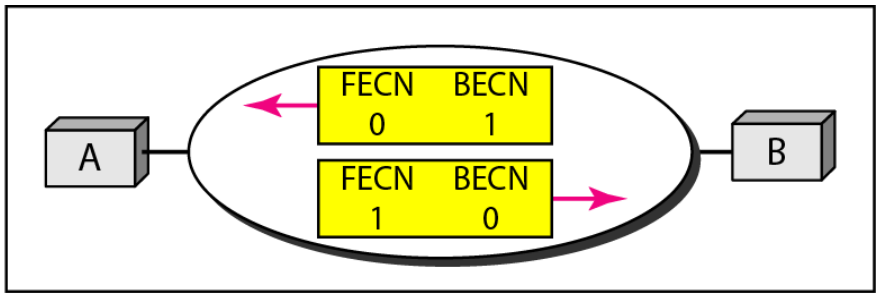
# Figure 24.13 *FECN*



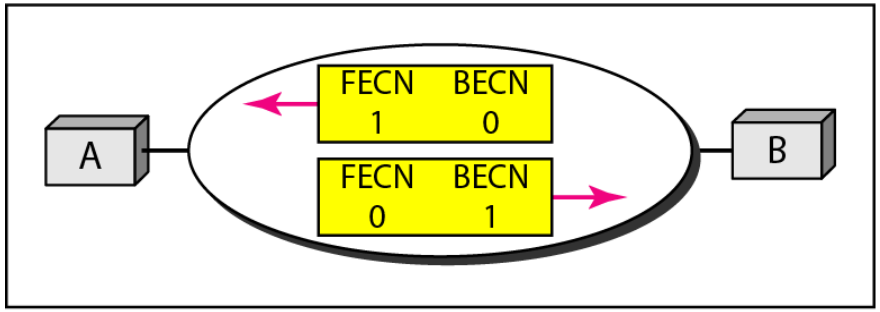
# Figure 24.14 *Four cases of congestion*



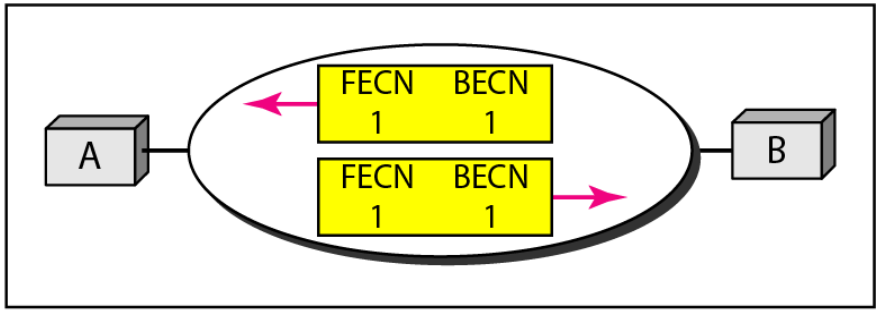

a. No congestion




b. Congestion in the direction A–B




c. Congestion in the direction B–A




d. Congestion in both directions

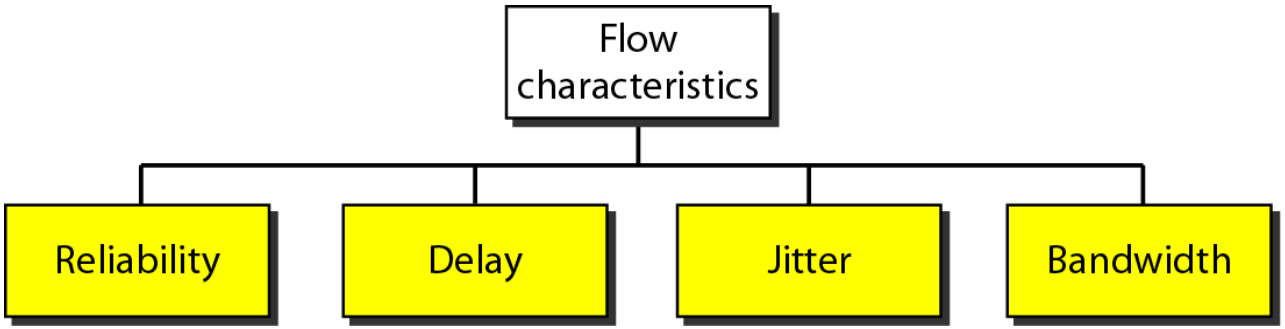
# 24-5 QUALITY OF SERVICE

*Quality of service (QoS) is an internetworking issue that has been discussed more than defined. We can informally define quality of service as something a flow seeks to attain.*

***Topics discussed in this section:***

**Flow Characteristics**
**Flow Classes**

## Figure 24.15 *Flow characteristics*

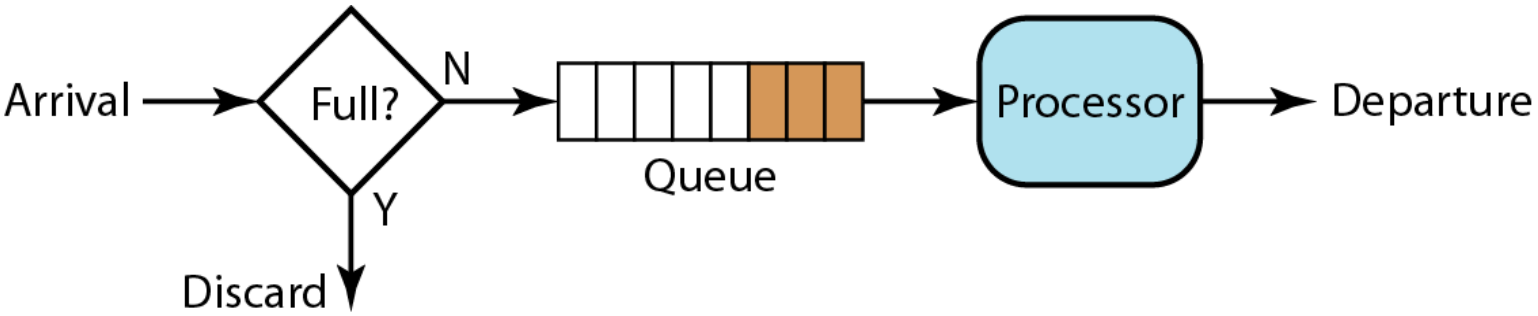


# 24-6 TECHNIQUES TO IMPROVE QoS

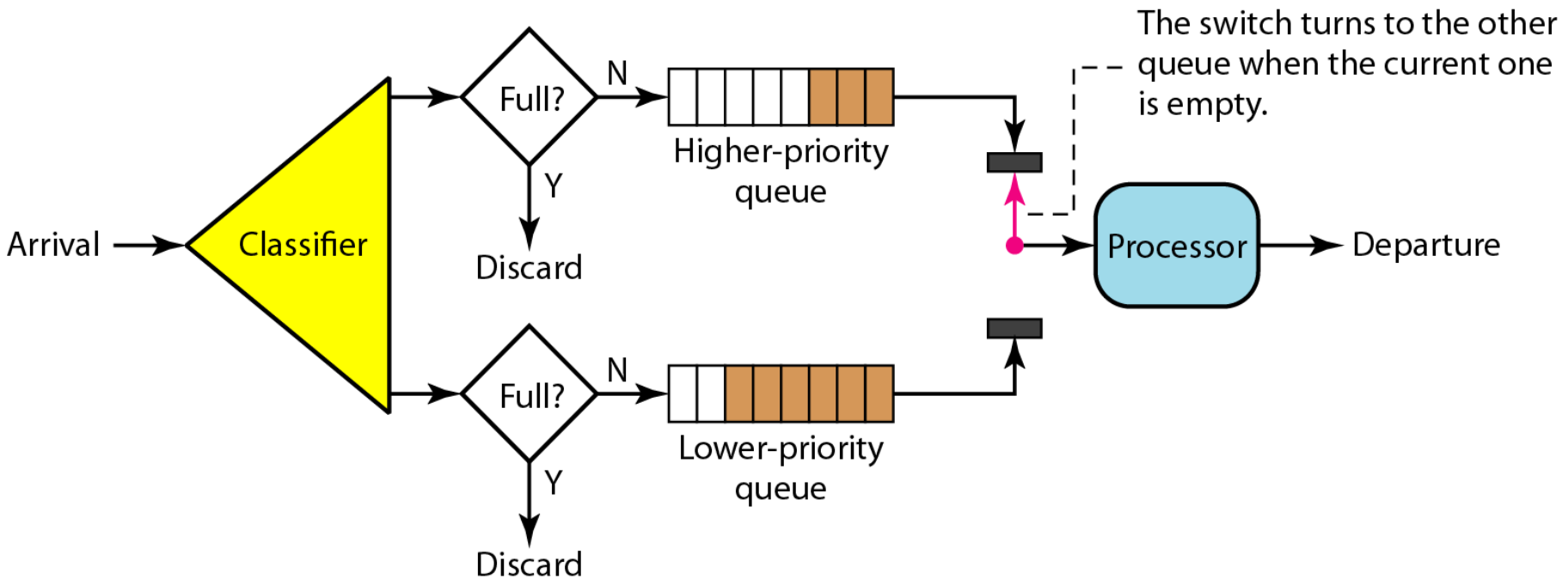
*In Section 24.5 we tried to define QoS in terms of its characteristics. In this section, we discuss some techniques that can be used to improve the quality of service. We briefly discuss four common methods: scheduling, traffic shaping, admission control, and resource reservation.*

**Scheduling**
**Traffic Shaping**
**Resource Reservation**
**Admission Control**

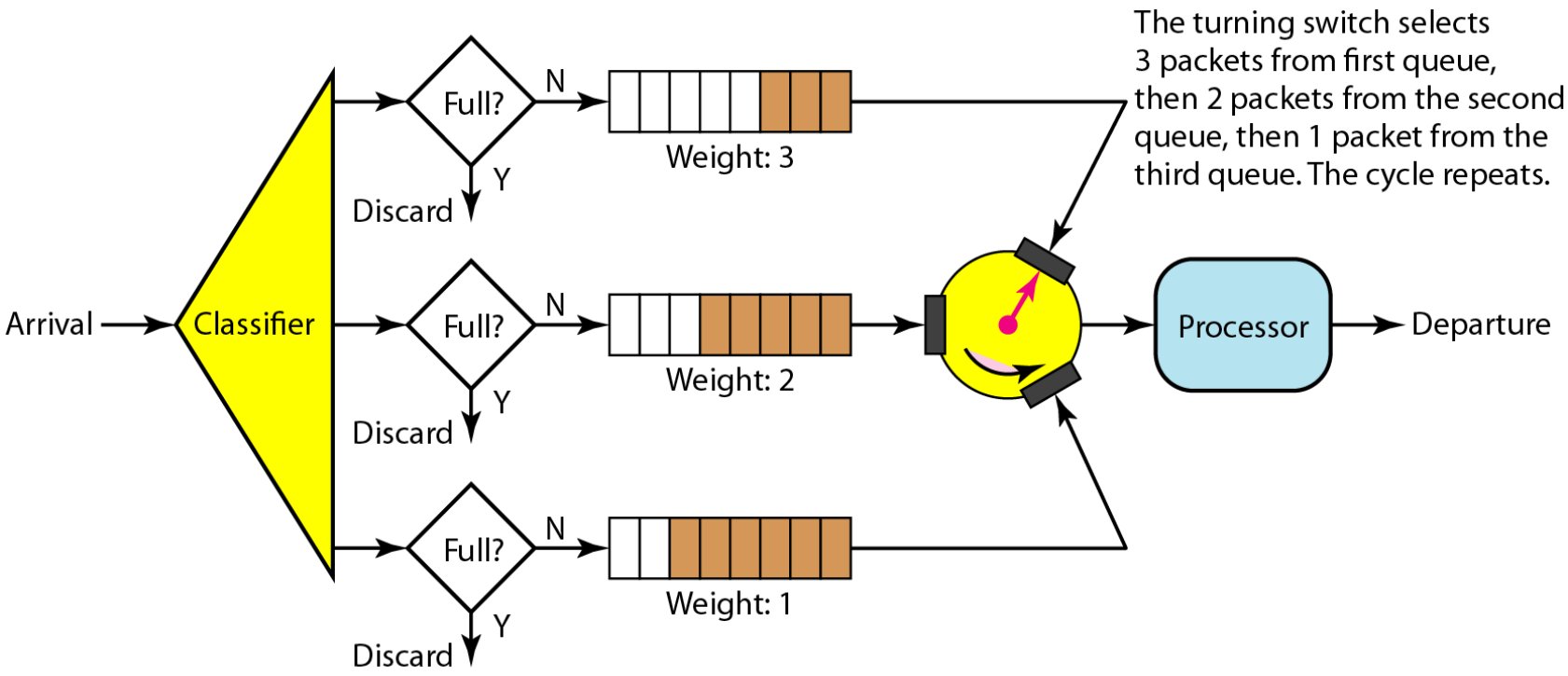
## Figure 24.16 *FIFO queue*



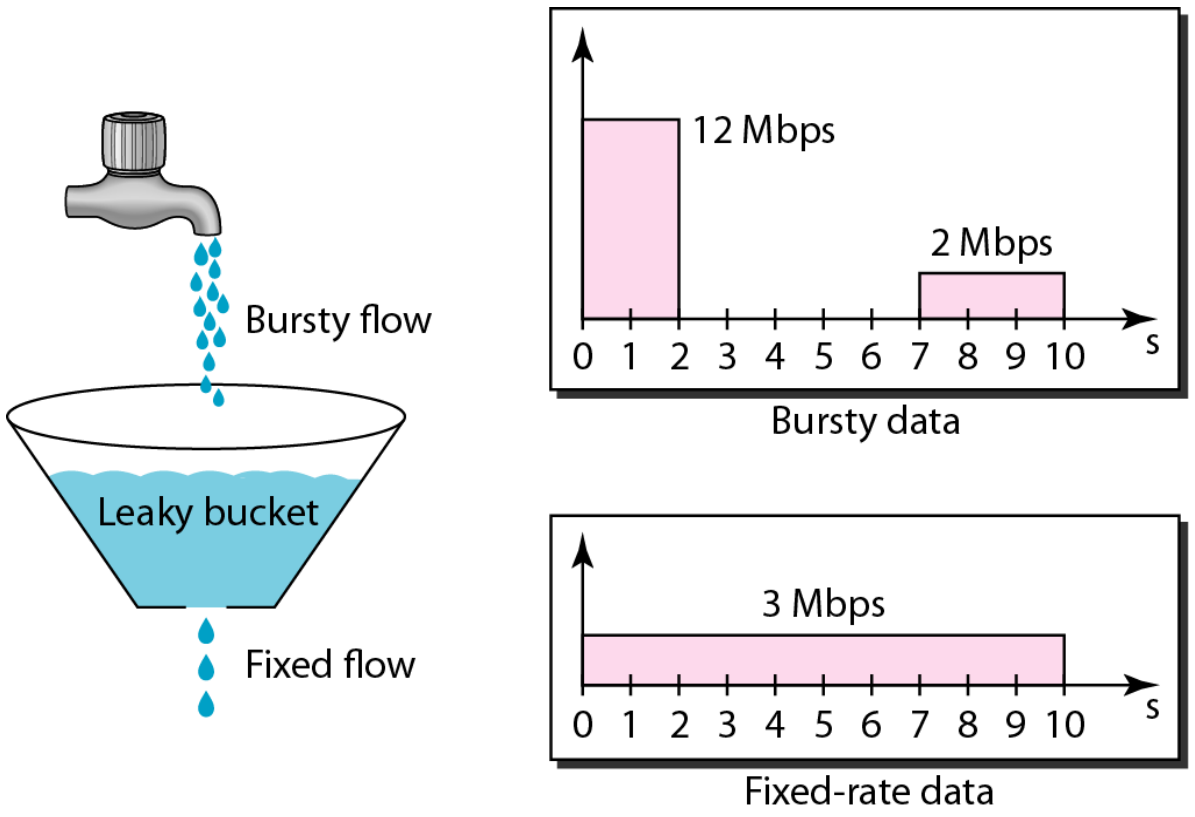
## Figure 24.17 *Priority queuing*



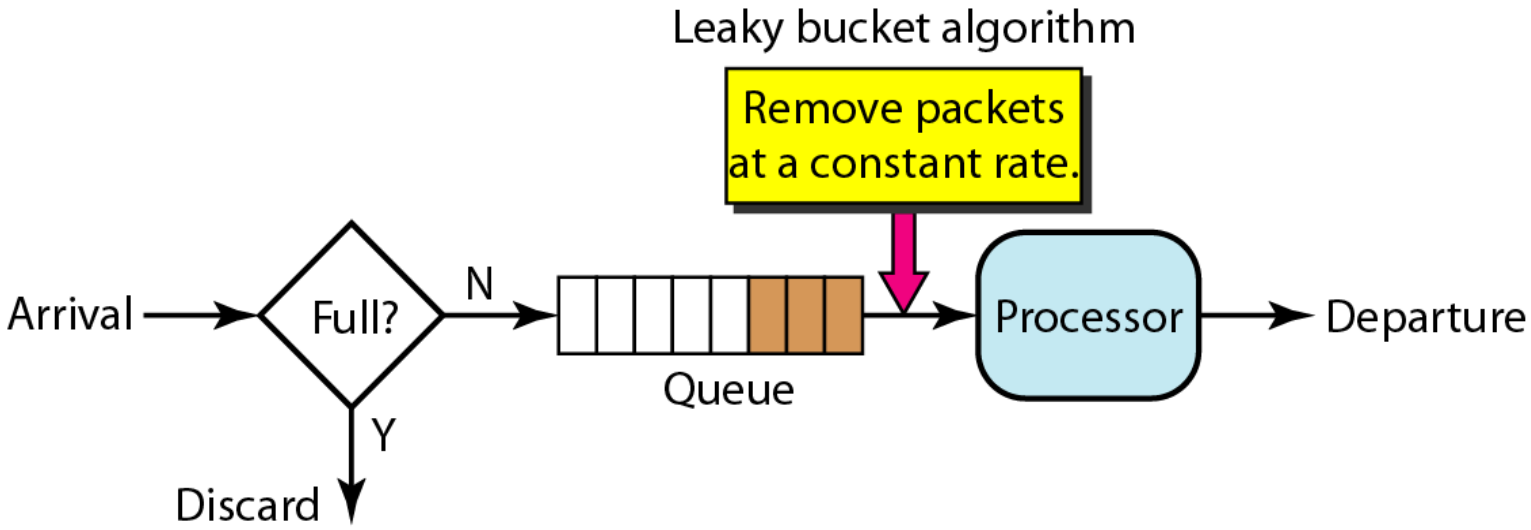
# Figure 24.18 *Weighted fair queuing*

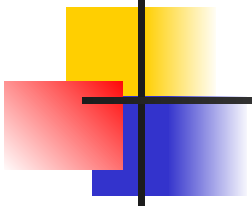


# Figure 24.19 *Leaky bucket*

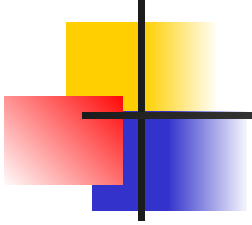


## Figure 24.20 *Leaky bucket implementation*





**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.**



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

## Figure 24.21 *Token bucket*