Data and Computer Communications

Tenth Edition

by William Stallings

CHAPTER 20

Congestion Control

We were doing very well, up to the kind of sum when a bath is filling at the rate of so many gallons and two holes are letting the water out, and please to say how long it will take to fill the bath, when my mother put down the socks she was darning and clicked her tongue in impatience.

"Filling up an old bath with holes in it, indeed. Who would be such a fool?"

"A sum it is, girl," my father said. "A sum. A problem for the mind."

"Filling the boy with old nonsense," Mama said.

"Not nonsense, Beth," my father said. "A sum, it is. The water pours in and takes so long. It pours out and takes so long. How long to fill? That is all."

ho would pour water into an old bath with holes?" my mother said. "Who d think to do it, but a lunatic?"

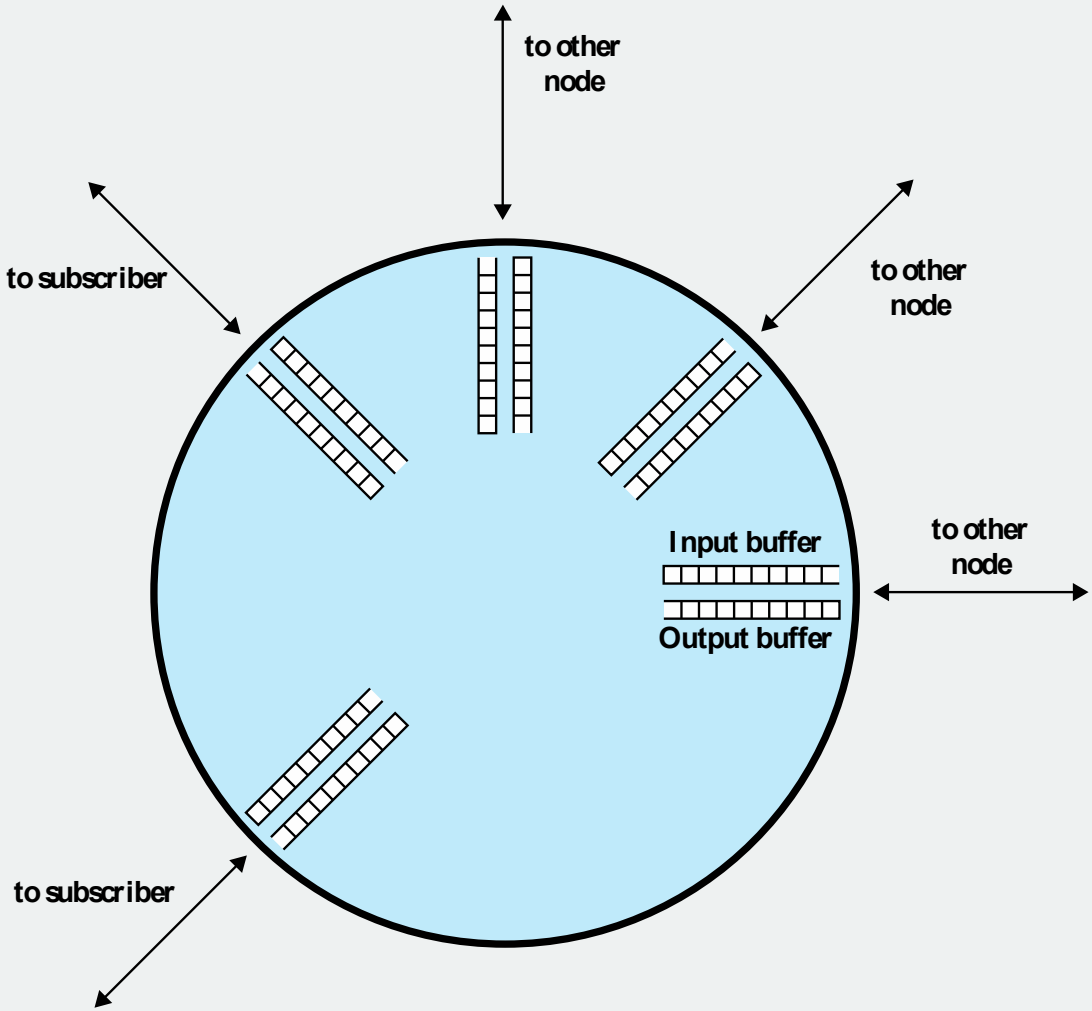


Figure 20.1 Input and Output Queues at Node

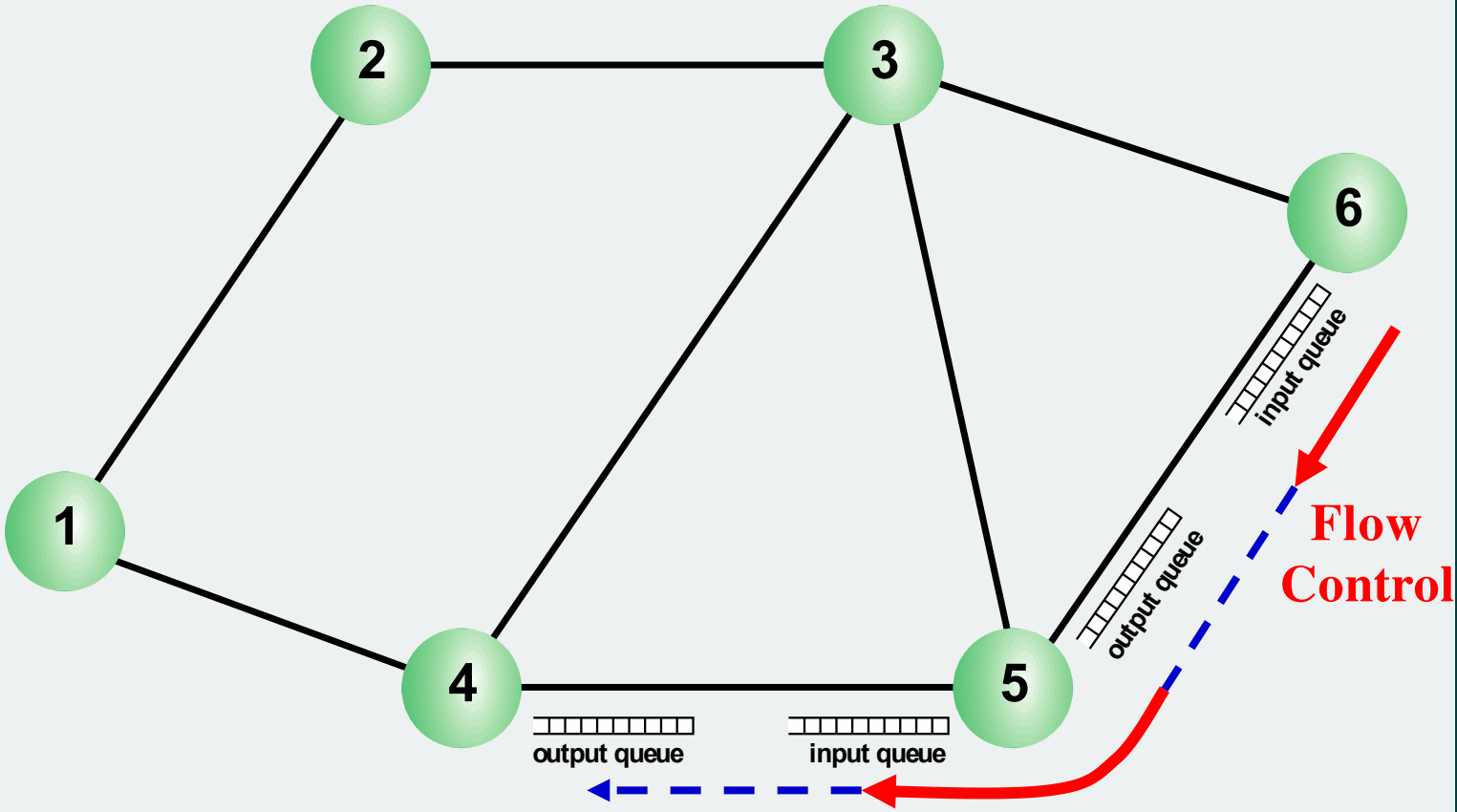


Figure 20.2 Interaction of Queues in a Data Network

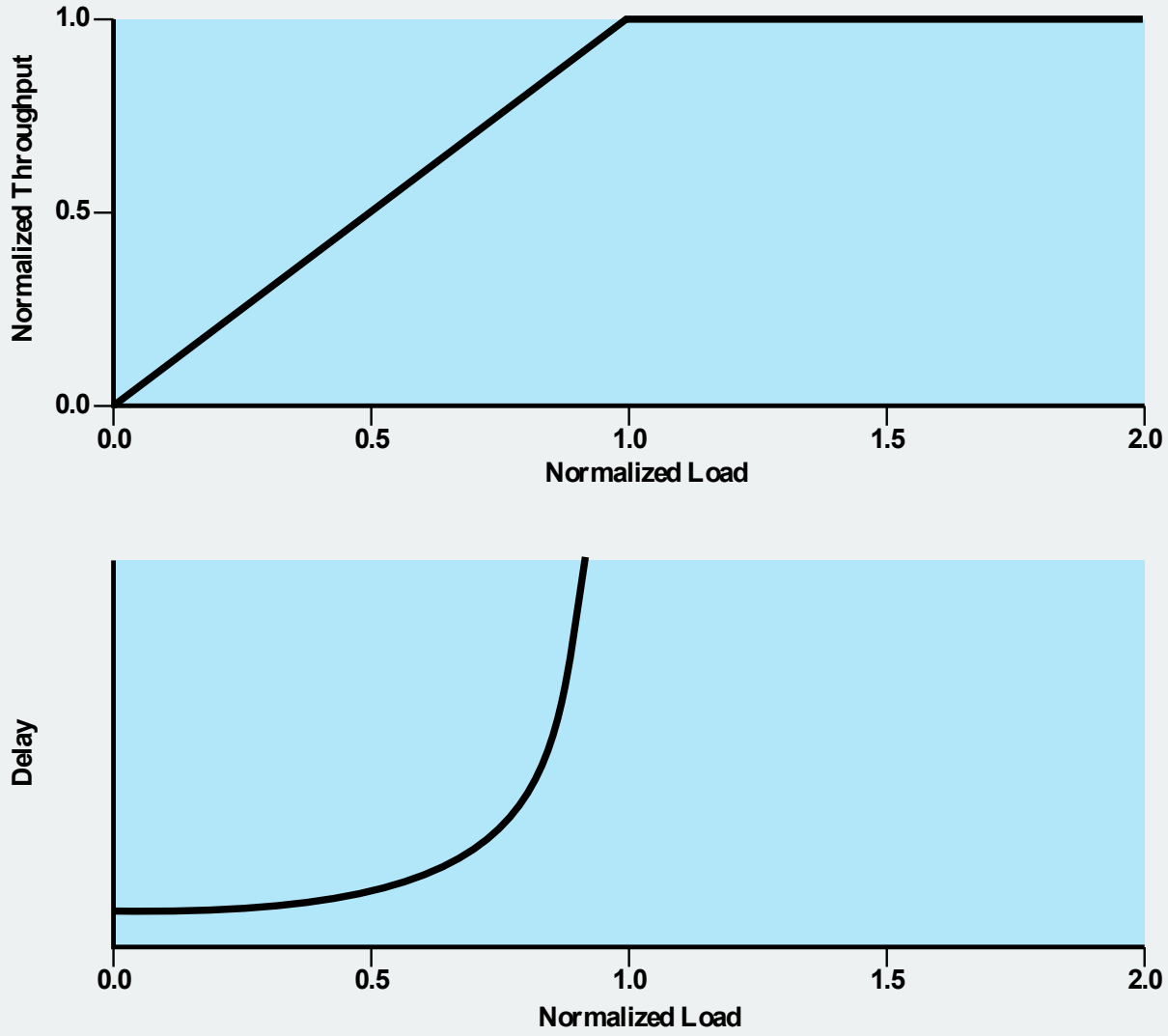


Figure 20.3 Ideal Network Utilization

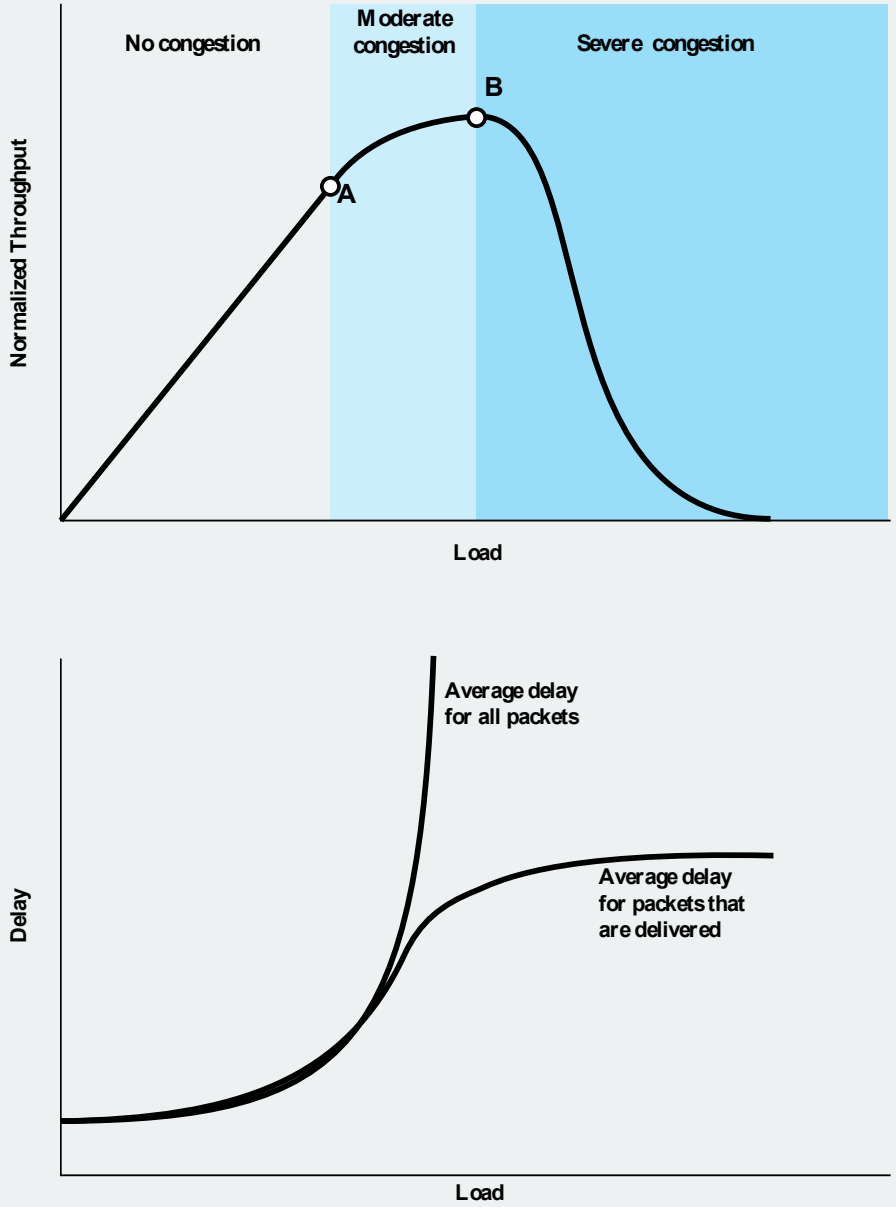


Figure 20.4 The Effects of Congestion

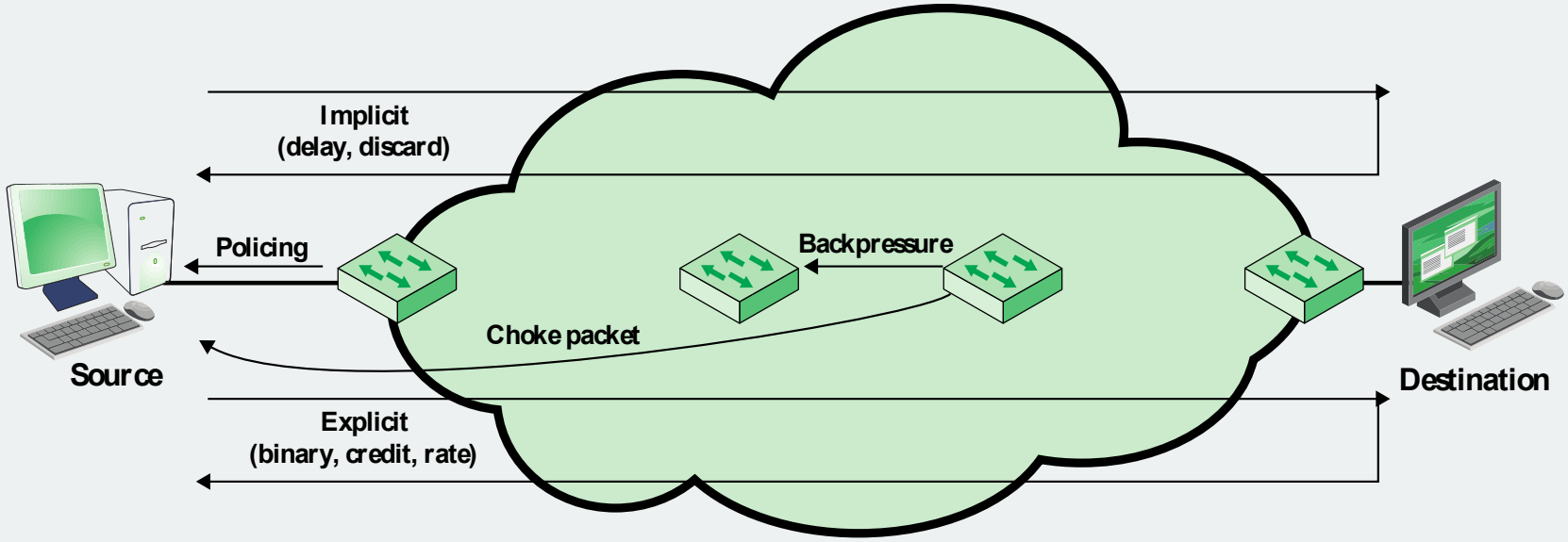


Figure 20.5 Mechanisms for Congestion Control

Backpressure

- If node becomes congested it can slow down or stop flow of packets from other nodes
- Can be exerted on the basis of links or logical connections
- Flow restriction propagates backward to sources, which are restricted in the flow of new packets into the network
- Can be selectively applied to logical connections so that the flow from one node to the next is only restricted or halted on some connections

Choke Packet

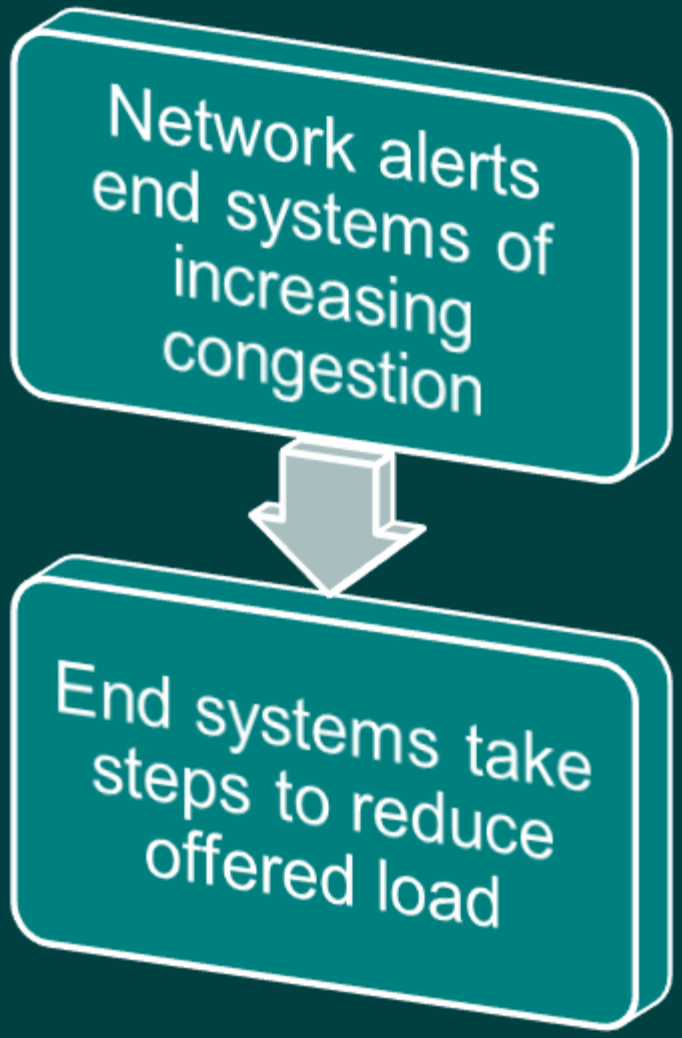
- A control packet
  - Generated at congested node
  - Sent back to source node
- An example is the Internet Control Message Protocol (ICMP) Source Quench packet
  - From router or destination end system
  - Source cuts back until it no longer receives quench messages
  - Message is issued for every discarded packet
  - Message may also be issued for anticipated congestion
- Is a crude technique for controlling congestion

Implicit Congestion Signaling

- With network congestion:
  - Transmission delay increases
  - Packets may be discarded
- Source can detect congestion and reduce flow
- Responsibility of end systems
- Effective on connectionless (datagram) networks
- Also used in connection-oriented networks
  - LAPF control is capable of detecting lost frames

Explicit Congestion Signaling

- Backward
  - Congestion avoidance notification in opposite direction to packet required
- Forward
  - Congestion avoidance notification in same direction as packet required



Explicit Signaling Categories

Binary

- A bit set in a packet indicates congestion

Credit based

- Indicates how many packets source may send
- Common for end-to-end flow control

Rate based

- Supply explicit data rate limit
- Nodes along path may request rate reduction

Traffic Management

Fairness

- Provide equal treatment of various flows

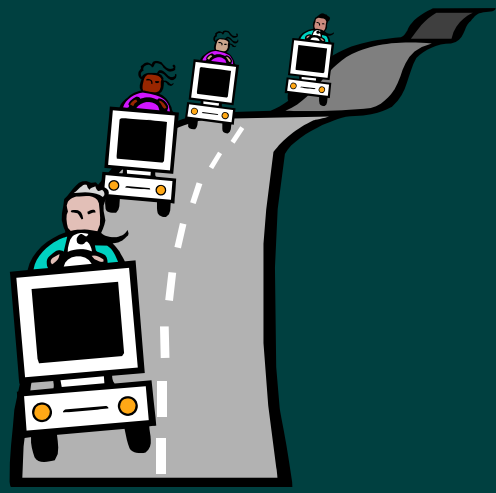
Quality of service

- Different treatment for different connections

Reservations

- Traffic contract between user and network
- Excess traffic discarded or handled on a best-effort basis

Traffic Shaping/Traffic Policing



- Two important tools in network management:
  - Traffic shaping
    - Concerned with traffic leaving the switch
    - Reduces packet clumping
    - Produces an output packet stream that is less burst and with a more regular flow of packets
  - Traffic policing
    - Concerned with traffic entering the switch
    - Packets that don't conform may be treated in one of the following ways:
      - Give the packet lower priority compared to packets in other output queues
      - Label the packet as nonconforming by setting the appropriate bits in a header
      - Discard the packet

Token Bucket

- Widely used traffic management tool
- Advantages:
  - Many traffic sources can be defined easily and accurately
  - Provides a concise description of the load to be imposed by a flow, enabling the service to determine easily the resource requirement
  - Provides the input parameters to a policing function

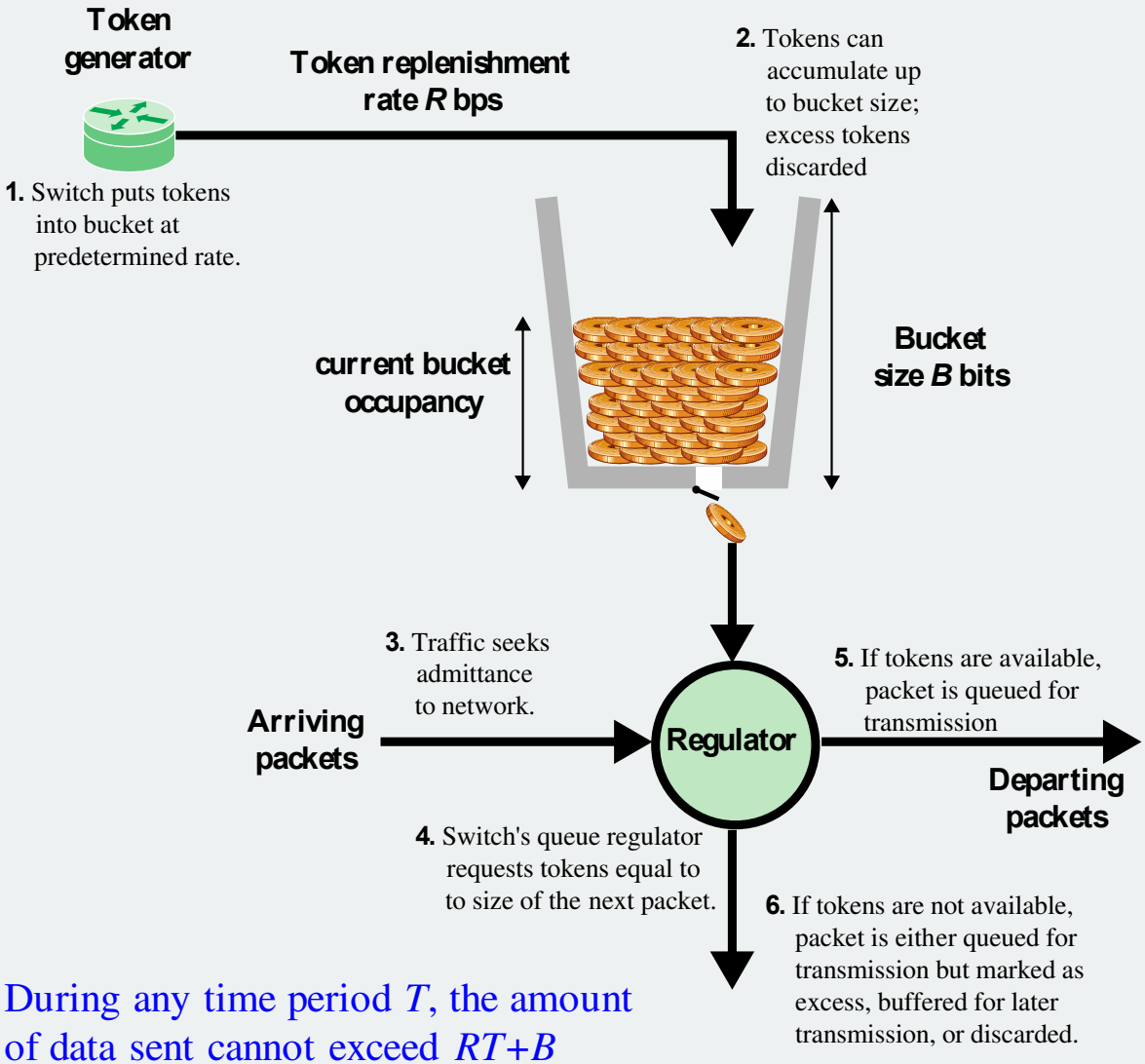


Figure 20.6 Token Bucket Scheme

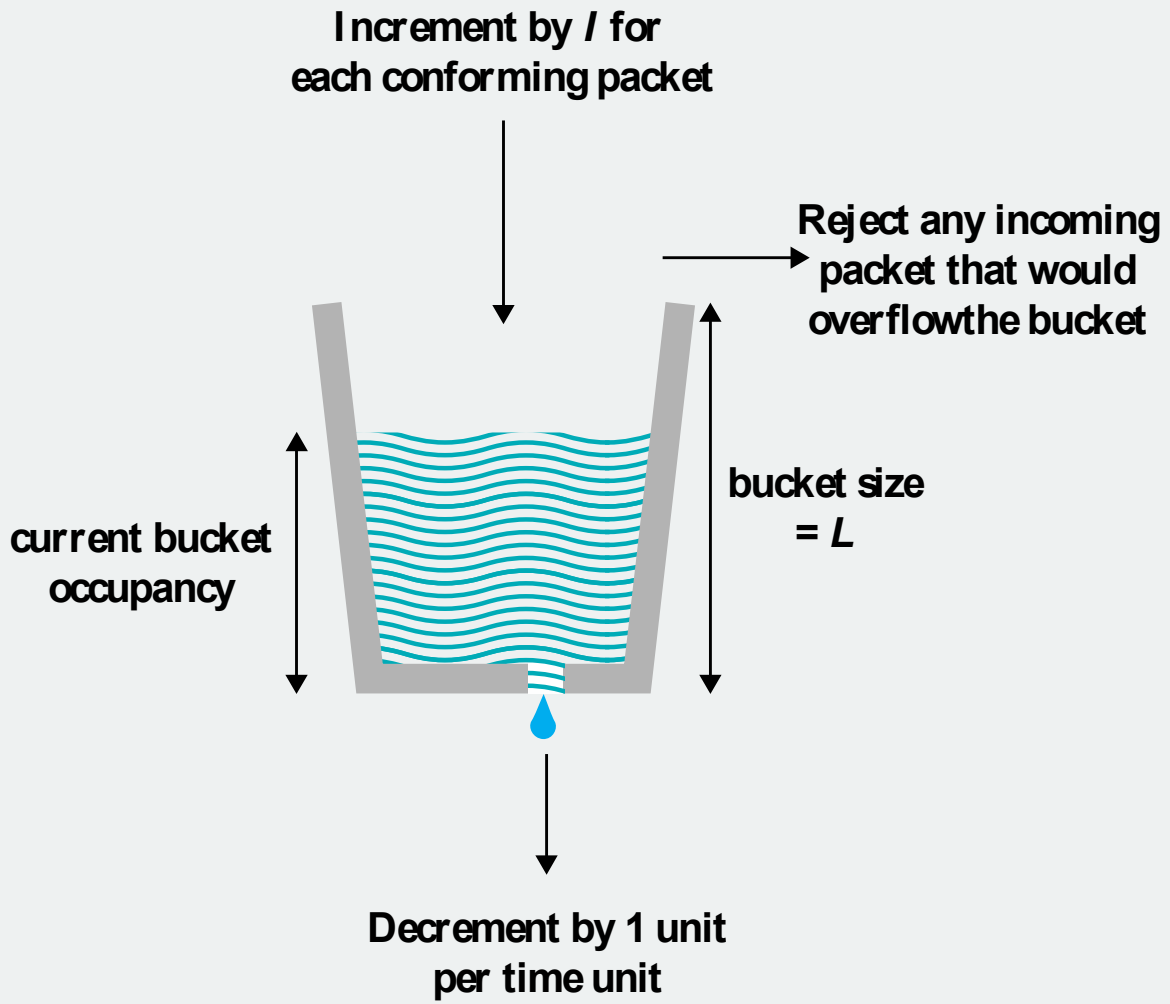


Figure 20.7 Leaky Bucket Algorithm

Congestion Control in Packet-Switching Networks

Send control packet to some or all source nodes

- Requires additional traffic during congestion

Rely on routing information

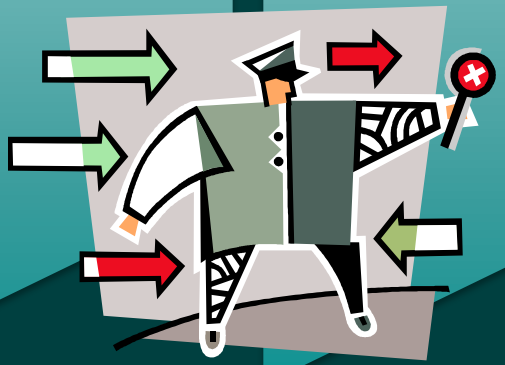
- May react too quickly

End to end probe packets

- Adds to overhead

Add congestion information to packets in transit

- Either backwards or forwards



TCP Congestion Control

- Flow control also used for congestion control
  - Recognize increased transit times & dropped packets
  - React by reducing flow of data
- RFC's 1122 & 2581 detail extensions
  - Tahoe, Reno & New Reno implementations
- Two categories of extensions:
  - Retransmission timer management
  - Window management

Flow Ctrl vs. Congestion Ctrl

Why Flow Control?

⇩

Prevent Receiver Buffer Overflow

⇩

Receiver-based window size (*rwnd*)

Why Congestion Control?

⇩

Try Not To Cause Congestion

⇩

Network-based window size (*cwnd*)

Sender's window = Min (*cwnd*, *rwnd*)

Table 20.1
Implementation of TCP Congestion Control Measures

Error Control ?!

Measure	RFC 1122	TCP Tahoe	TCP Reno	NewReno
RTT Variance Estimation	✓	✓	✓	✓
Exponential RTO Backoff	✓	✓	✓	✓
Karn's Algorithm	✓	✓	✓	✓
Slow Start	✓	✓	✓	✓
Dynamic Window Sizing on Congestion	✓	✓	✓	✓
Fast Retransmit		✓	✓	✓
Fast Recovery			✓	✓
Modified Fast Recovery				✓

Retransmission Timer Management

As network or internet conditions change a static retransmission timer is likely to be either too long or too short

Virtually all TCP implementations attempt to estimate the current round-trip time and then set the timer to a value somewhat greater than the estimated time

Simple average:

- Take the average of observed round-trip times over a number of segments
- If the average accurately predicts future round-trip times, then the resulting retransmission timer will yield good performance

Exponential average:

- Technique for predicting the next value on the basis of a time series of past values
- Specified in RFC 793

Retransmission Timer (cont)

- **Simple Average**
  - **RTT(i): round-trip time observed for the *i*th transmitted segment**
  - **ARTT(K): average round-trip time for the first *K* segments**

$$ARTT(K+1) = \frac{1}{K+1}\sum_{i=1}^{K+1} RTT(i)$$ **or**

$$ARTT(K+1) = \frac{K}{K+1} ARTT(K) + \frac{1}{K+1} RTT(K+1)$$

Retransmission Timer (cont)

- **Exponential Average**
  - **SRTT: smoothed round-trip time estimate**
  - **RTO: retransmission timer**

$$SRTT(K+1) = \alpha \times SRTT(K) + (1-\alpha) \times RTT(K+1)$$

$$RTO(K+1) = SRTT(K+1) + \Delta$$

RFC793:

$$RTO(K+1) = Min(UBOUND, MAX(LBOUND, \beta \times SRTT(K+1)))$$

Example values: α: 0.8 ~ 0.9, β: 1.3 ~ 2.0

RTT Variance Estimation

- **AERR(K): sample mean deviation measured at time K**

$$AERR(K+1) = RTT(K+1) - ARTT(K)$$

$$ADEV(K+1) = \frac{1}{K+1}\sum_{i=1}^{K+1}|AERR(i)|$$

$$= \frac{K}{K+1}ADEV(K) + \frac{1}{K+1}|AERR(K+1)|$$

RTT Variance Estimation (cont)

- **Jacobson's Algorithm**

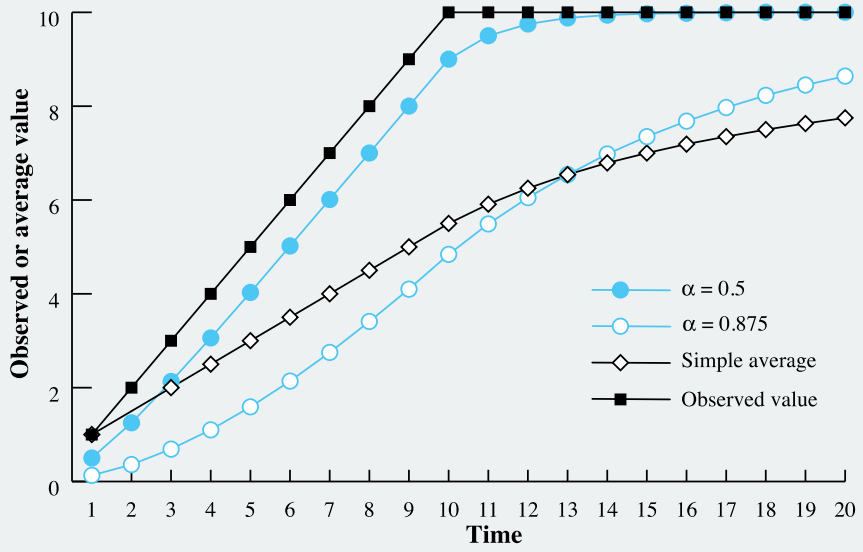
$$SRTT(K+1) = (1-g) \times SRTT(K) + g \times RTT(K+1)$$

$$SERR(K+1) = RTT(K+1) - SRTT(K)$$

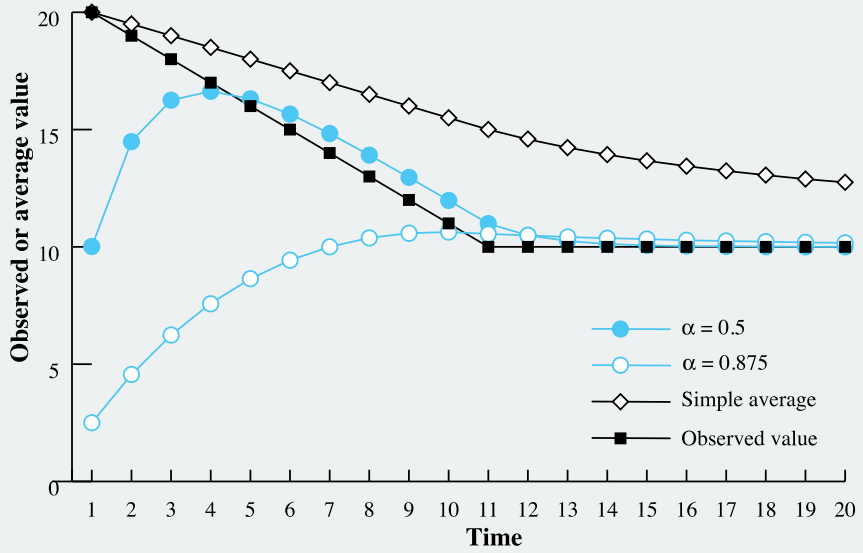
$$SDEV(K+1) = (1-h) \times SDEV(K) + h \times \left| SERR(K+1) \right|$$

$$RTO(K+1) = SRTT(K+1) + f \times SDEV(K+1)$$

- **g = 1/8 = 0.125, h = ¼ = 0.25, f = 2**

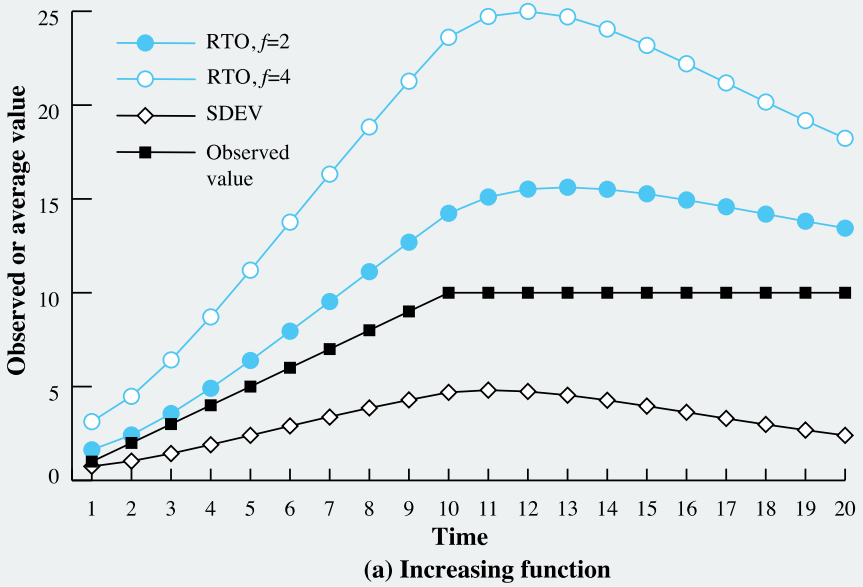


(a) Increasing function

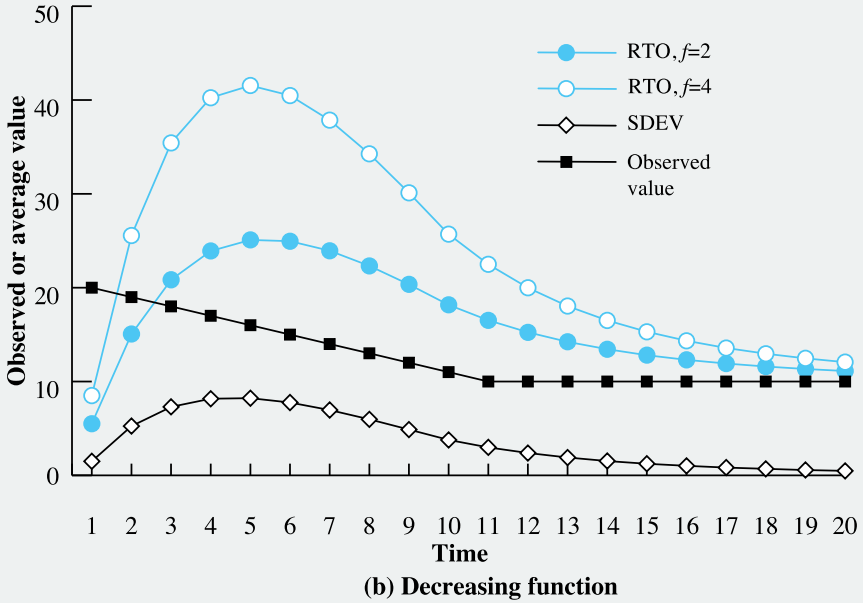


(b) Decreasing function

Figure 20.8 Use of Exponential Averaging



(a) Increasing function



(b) Decreasing function

Figure 20.9 Jacobson's RTO Calculation

Exponential RTO Backoff

- When a TCP sender times out on a segment, it must retransmit that segment
  - RFC 793 assumes that the same RTO value will be used
  - Because the time-out is probably due to network congestion, maintaining the same RTO value is ill advised
- A more sensible policy dictates that a sending TCP entity increase its RTO each time a segment is retransmitted

Karn's Algorithm

Do not use the measured RTT for a retransmitted segment to update SRTT and SDEV

Use the backoff RTO value for succeeding segments until an acknowledgment arrives for a segment that has not been retransmitted

Calculate the backoff RTO using the equation RTO = q * RTO when a retransmission occurs

Window Management

- The size of TCP's send window can have a critical effect on whether TCP can be used efficiently without causing congestion
- Two techniques found in virtually all modern implementation of TCP are:
  - Slow start
  - Dynamic window sizing on congestion

Slow Start

- Gradually expand the window until acknowledgments are received
- TCP transmission is constrained by

$$awnd = \text{MIN}[credit, cwnd]$$

where

- awnd = allowed window, in segment
  - This is the number of segments that TCP is currently allowed to send without receiving further acknowledgments
- *cwnd*= congestion window, in segments
  - A window used by TCP during startup and to reduce flow during periods of congestion
- *credit* = the amount of unused credit granted in the most recent acknowledgment, in segments
  - When an acknowledgment is received, this value is calculated as window/segment_size, where window is a field in the incoming TCP segment (the amount of data the peer TCP entity is willing to accept)

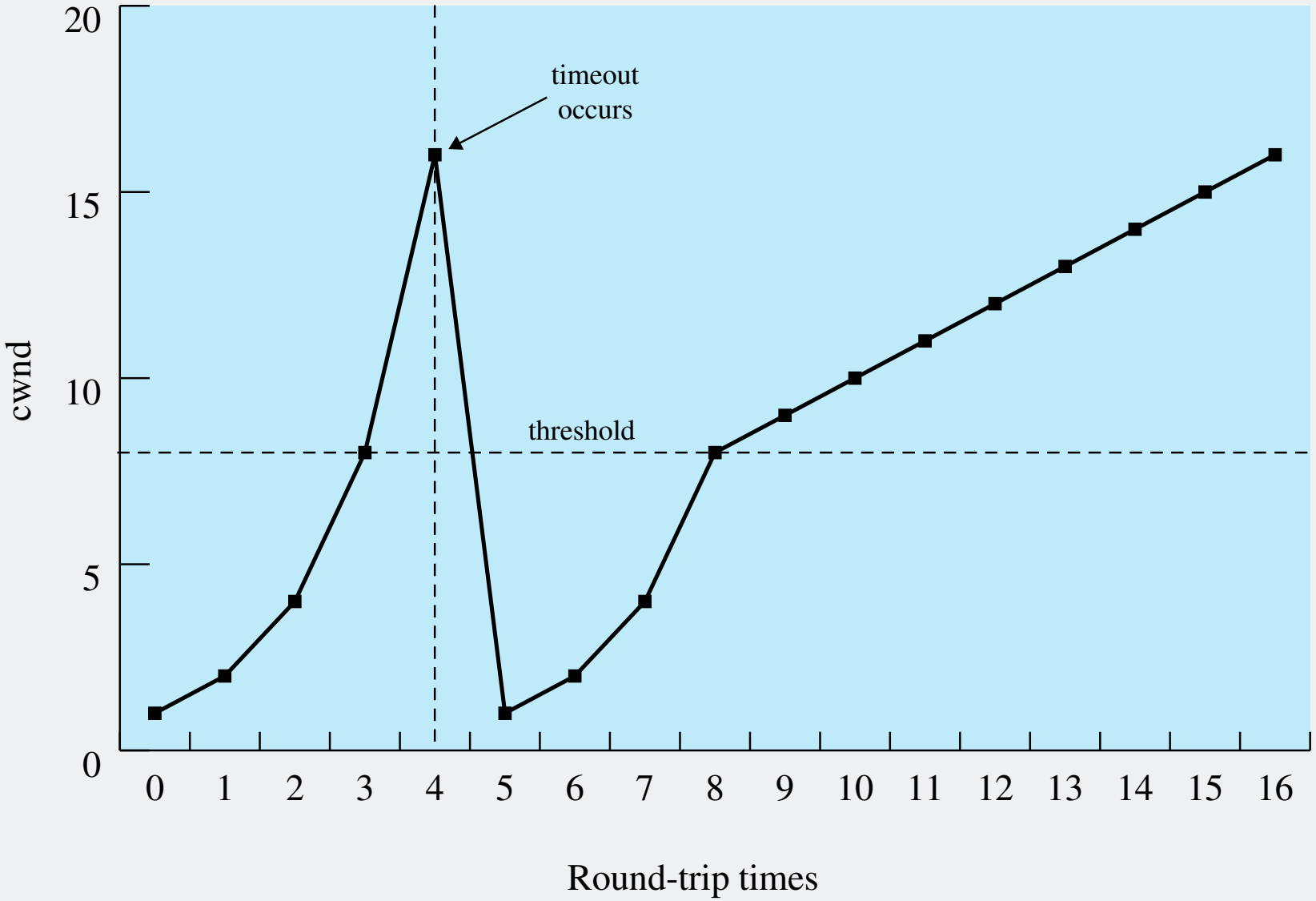


Figure 20.10 Illustration of Slow Start and Congestion Avoidance

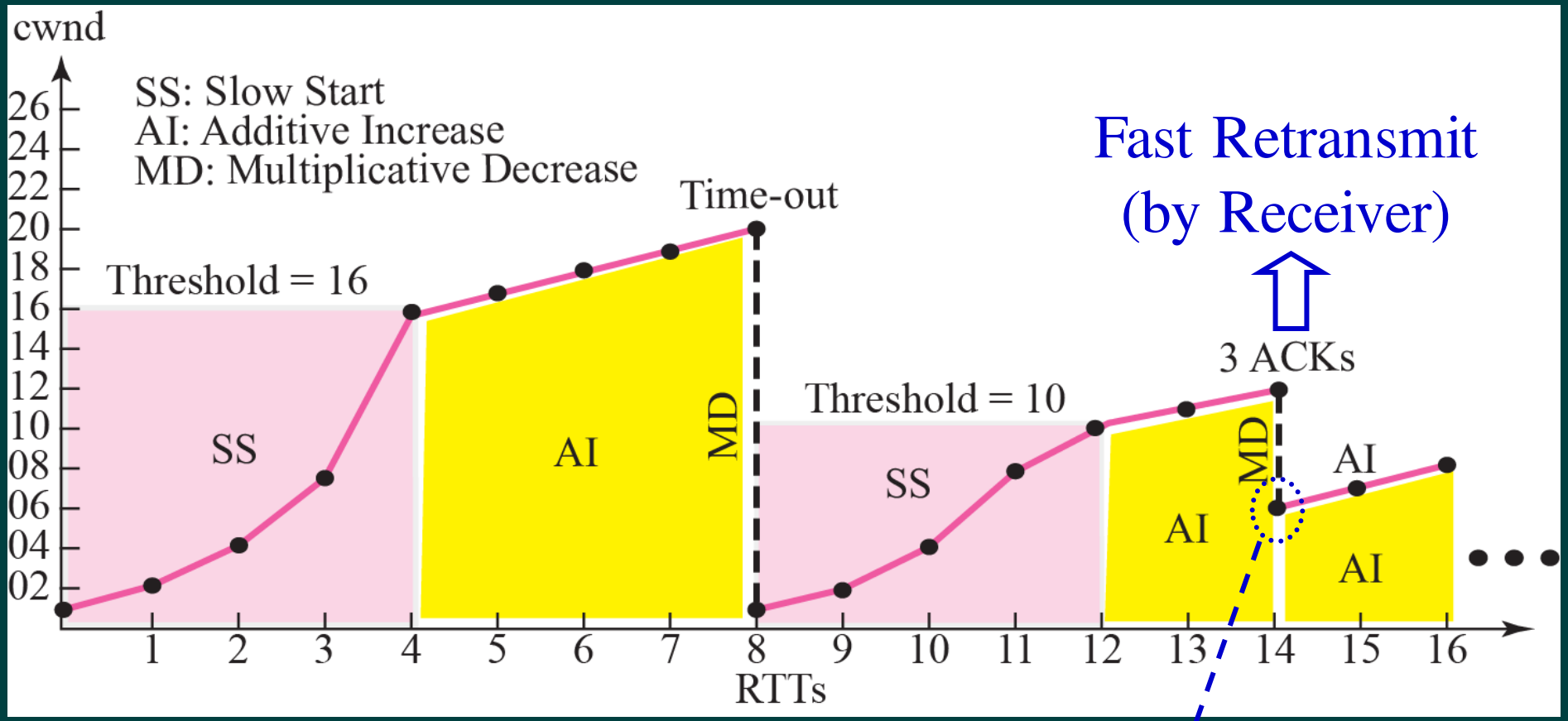
Fast Retransmit

- Under some circumstances improve on the performance provided by RTO
- Takes advantage of the rule that if a TCP entity receives a segment out of order, it must immediately issue an ACK for the last in-order segment that was received

Fast Recover

- Retransmit the lost segment, cut *cwnd* in half, and then proceed with the linear increase of *cwnd*
- RFC 3782 modifies the fast recovery algorithm to improve the response when two segments are lost within a single window

TCP Congestion Control



Explicit Congestion Notification (ECN)

An example of explicit congestion signaling is the ECN capability provided in IP and TCP and defined in RFC 3168

IP header includes the 2-bit ECN field, which enables routers to indicate to end nodes packets that are experiencing congestion, without the necessity of immediately dropping such packets

- A value of 00 indicates a packet that is not using ECN
- As value of 01 or 10 is set by the data sender to indicate that the end-points of the transport protocol are ECN-capable
- A value of 11 is set by a router to indicate congestion has been encountered

TCP header includes two one-bit flags:

- ECN-Echo flag
- CWR (congestion window reduced) flag

See the typical sequence of events on P678.

TCP Friendliness

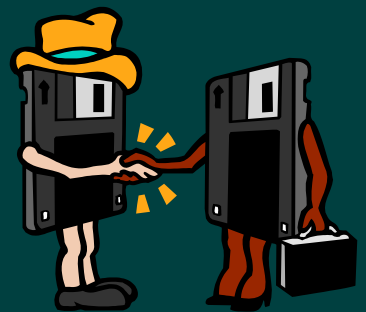
- Also known as TCP compatible, refers to flows that will back off appropriately in the face of congestion
- Congestion collapse
  - Too many traffic flows operating independently and without regard to their effect on overall Internet performance
- TCP-friendly equation (P679)

$$T \leq \frac{1.22 \times B}{R \times \sqrt{P}}$$

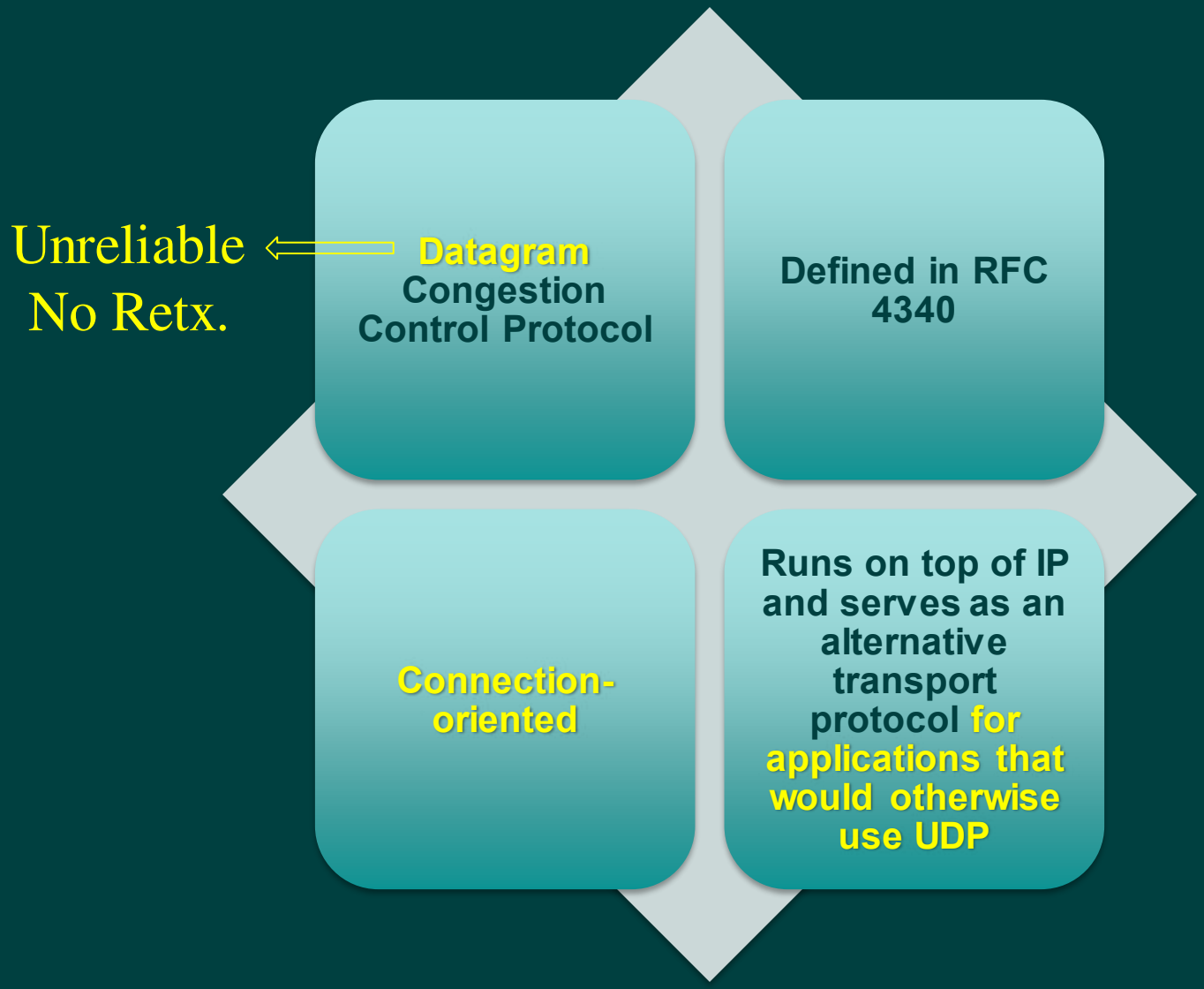
Max pkt length

RTT

Pkt drop rate



DCCP Operation



DCCP Packet Types

DCCP-Request:
Sent by the client to initiate a connection (the first part of the three-way initiation handshake)

DCCP-Response:
Sent by the server in response to a DCCP-Request (the second part of the three-way initiation handshake)

DCCP-Data:
Used to transmit application data

DCCP-Ack:
Used to transmit pure acknowledgments

DCCP-DataAck:
Used to transmit application data with piggybacked acknowledgment information

DCCP-CloseReq:
Sent by the server to request that the client close the connection

DCCP-Close:
Used by the client or the server to close the connection; elicits a DCCP-Reset in response

DCCP-Reset:
Used to terminate the connection, either normally or abnormally

DCCP-Sync, DCCP-SyncAck:
Used to resynchronize sequence numbers after large bursts of loss

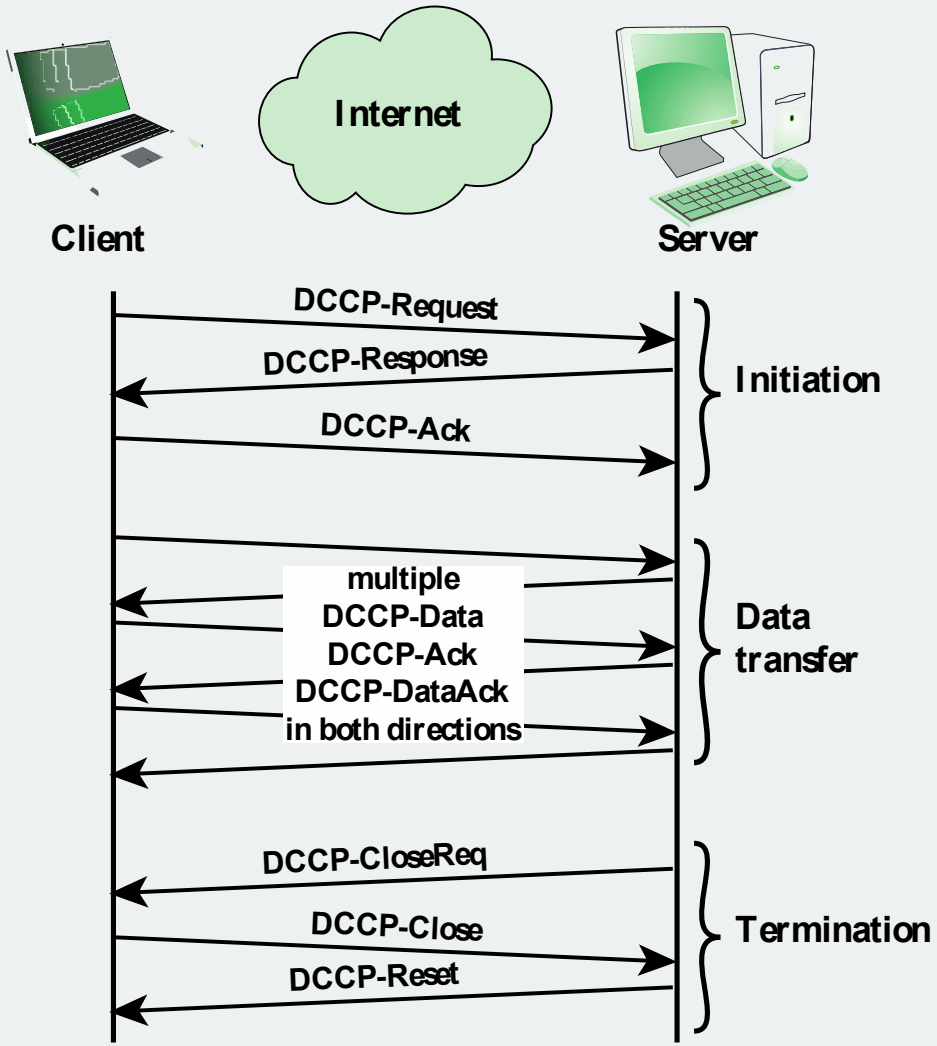


Figure 20.11 DCCP Packet Exchange

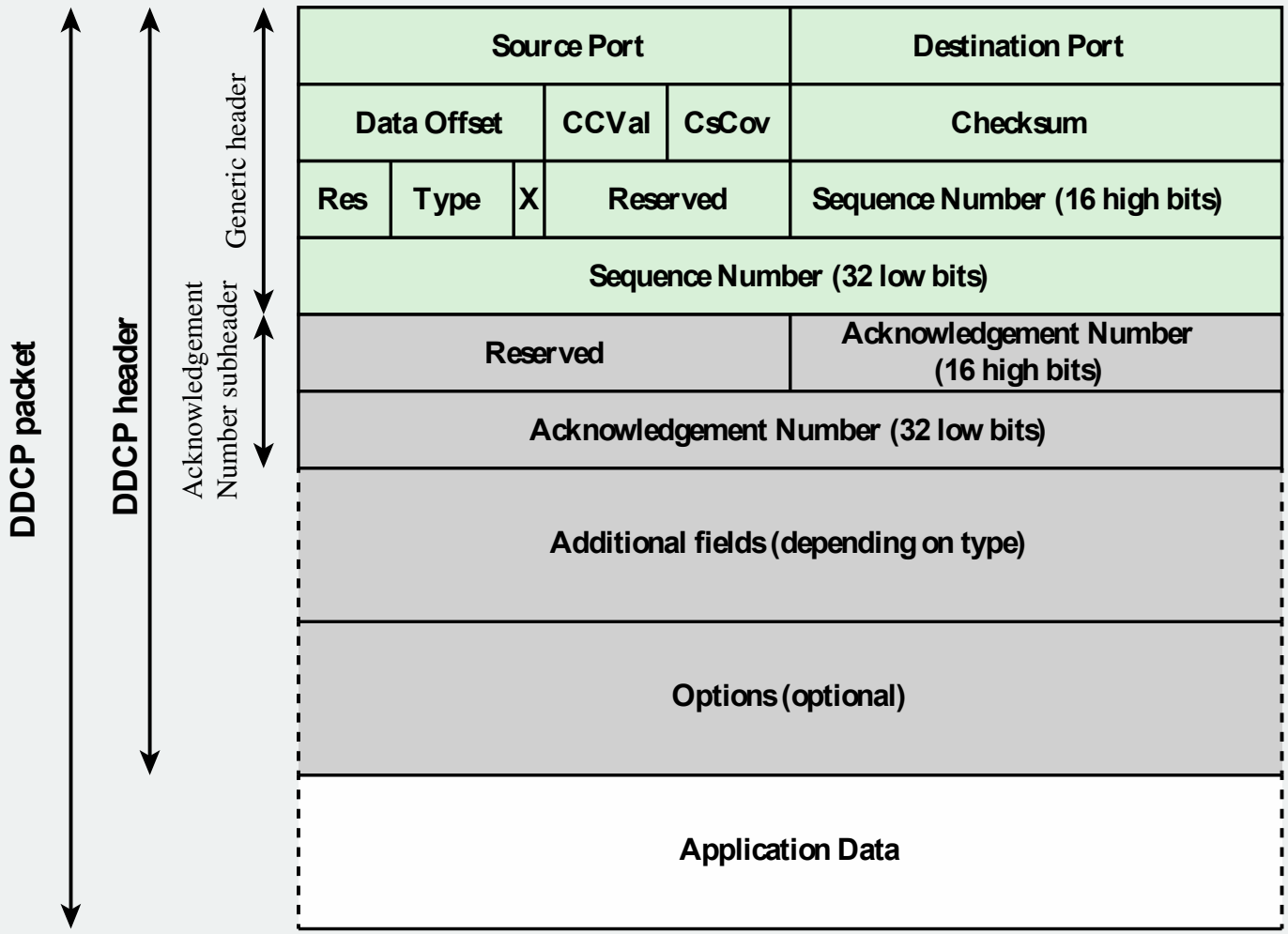


Figure 20.12 DCCP Packet Format (X = 1)

DCCP Congestion Control

- The Internet DCCP Working Group has defined two congestion control mechanisms, each assigned a congestion control identifier (CCID):

TCP-like congestion control (CCID 2)

- This mechanism has behavior modeled directly on TCP, including congestion window, slow start, and time-outs
- Applications should use CCID 2 if they prefer maximum bandwidth utilization to steadiness of rate

TCP-friendly rate control (TFDC) (CCID 3)

- This mechanism is based on the use of a variant of the TCP-friendly equation
- Suitable for applications such as streaming media, where a relatively smooth sending rate is important

Summary

- Effects of congestion
  - Ideal performance
  - Practical performance
- Congestion control
  - Backpressure
  - Choke packet
  - Implicit and explicit congestion signaling
- Traffic management
  - Fairness
  - Quality of service
  - Reservations
  - Traffic shaping and policing
- Congestion control in packet-switching networks
- TCP congestion control
  - Retransmission timer management
  - Window management
  - Explicit congestion notification
- Datagram congestion control protocol
  - TCP friendliness
  - DCCP operation
  - DCCP packet format
  - DCCP congestion control