Internet Fundamentals

Lecture-30

Internet Congestion Control

Opening

- What is the Internet?
  - A network is a group of connected, communicating devices.
  - An internet is two or more networks that can communicate with each other.
  - The most notable internet is called the Internet.
- How congestion control in the Internet?
  - Hosts & network devices

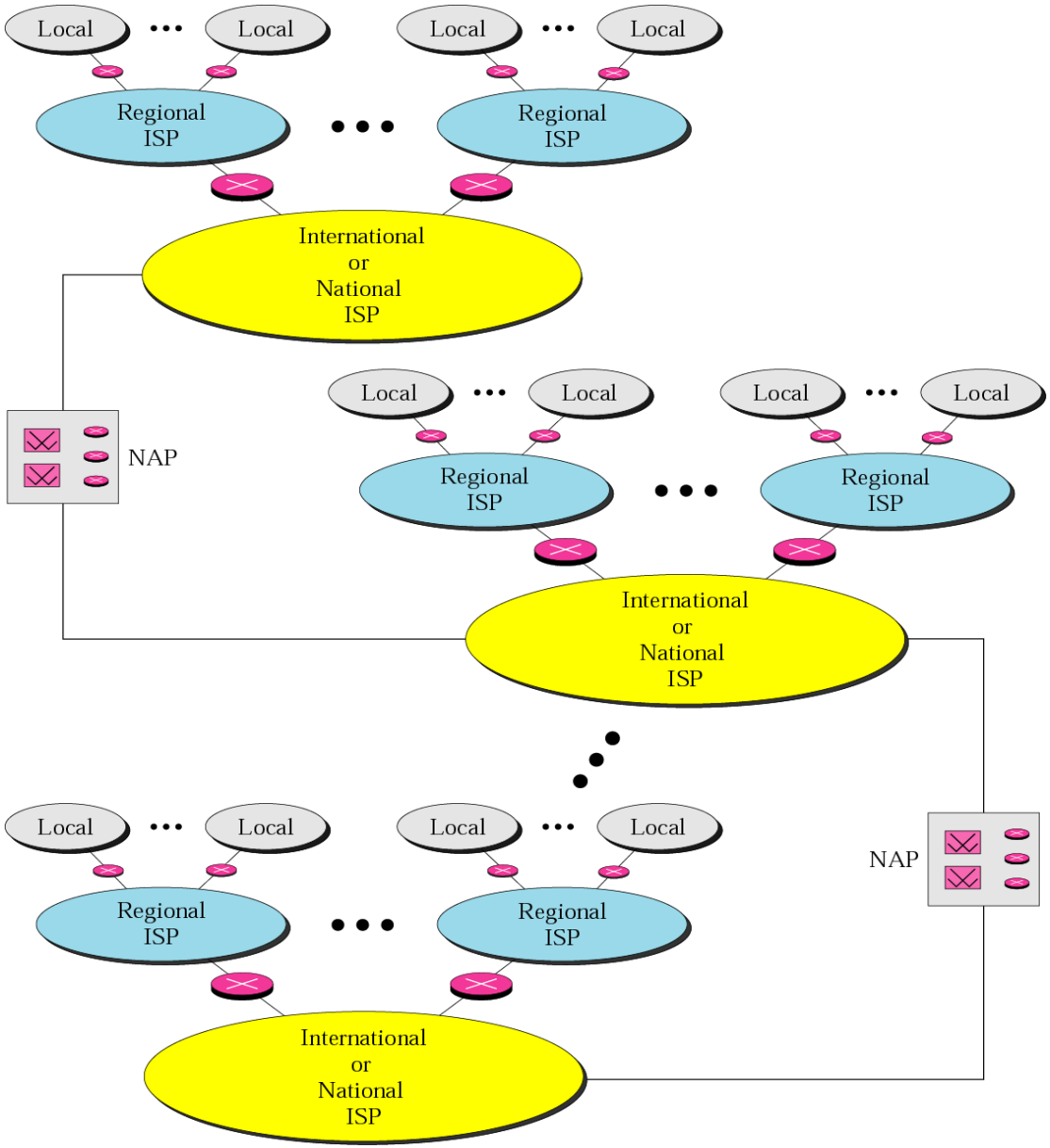
Outline

- The Internet
- Transmission Control Protocol - TCP
- Active Queue Management - AQM
- Admission Control
- Conclusions

The Internet

- Internet history
  - 1969: Four-node ARPANET established.
  - 1972: Vint Cerf and Bob Kahn's Internet Project
  - 1973: Development of TCP/IP begins.
  - 1981: UNIX operating system includes TCP/IP.
  - 1983: TCP/IP became the official protocol for the ARPANET.
  - 1986: NSFNET (sponsored by the National Science Foundation)
  - 1990: ARPANET retired.
  - 1991: A high-speed Internet backbone called ANSNET was build by IBM, Merit, and MCI.
  - 1995: Companies known as Internet Service Providers (ISPs) started.

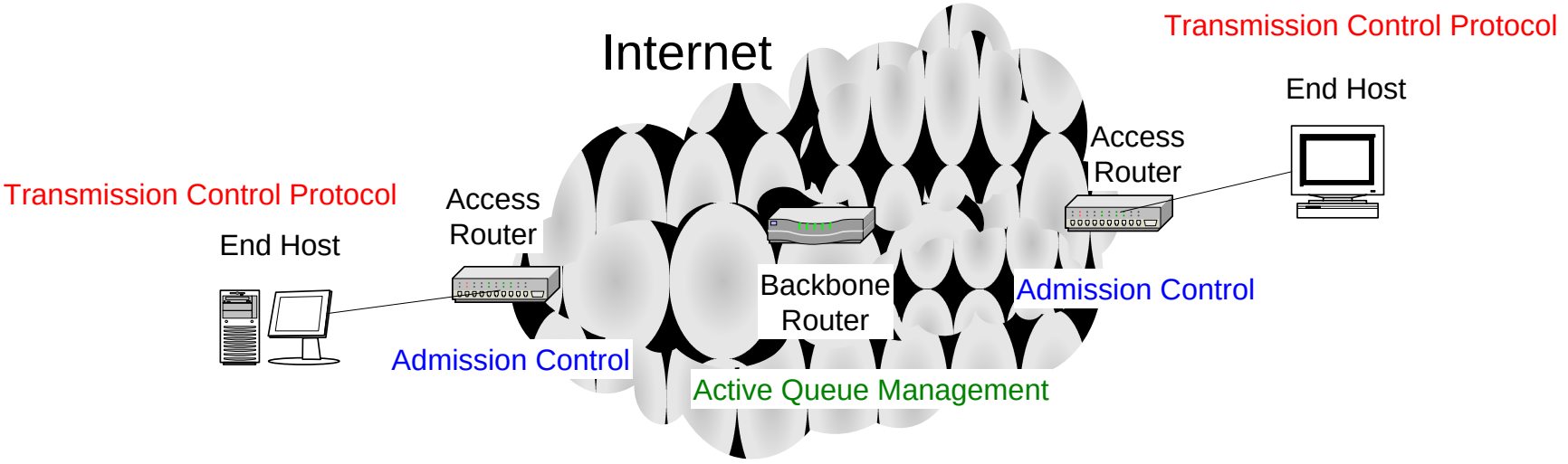
Internet Today



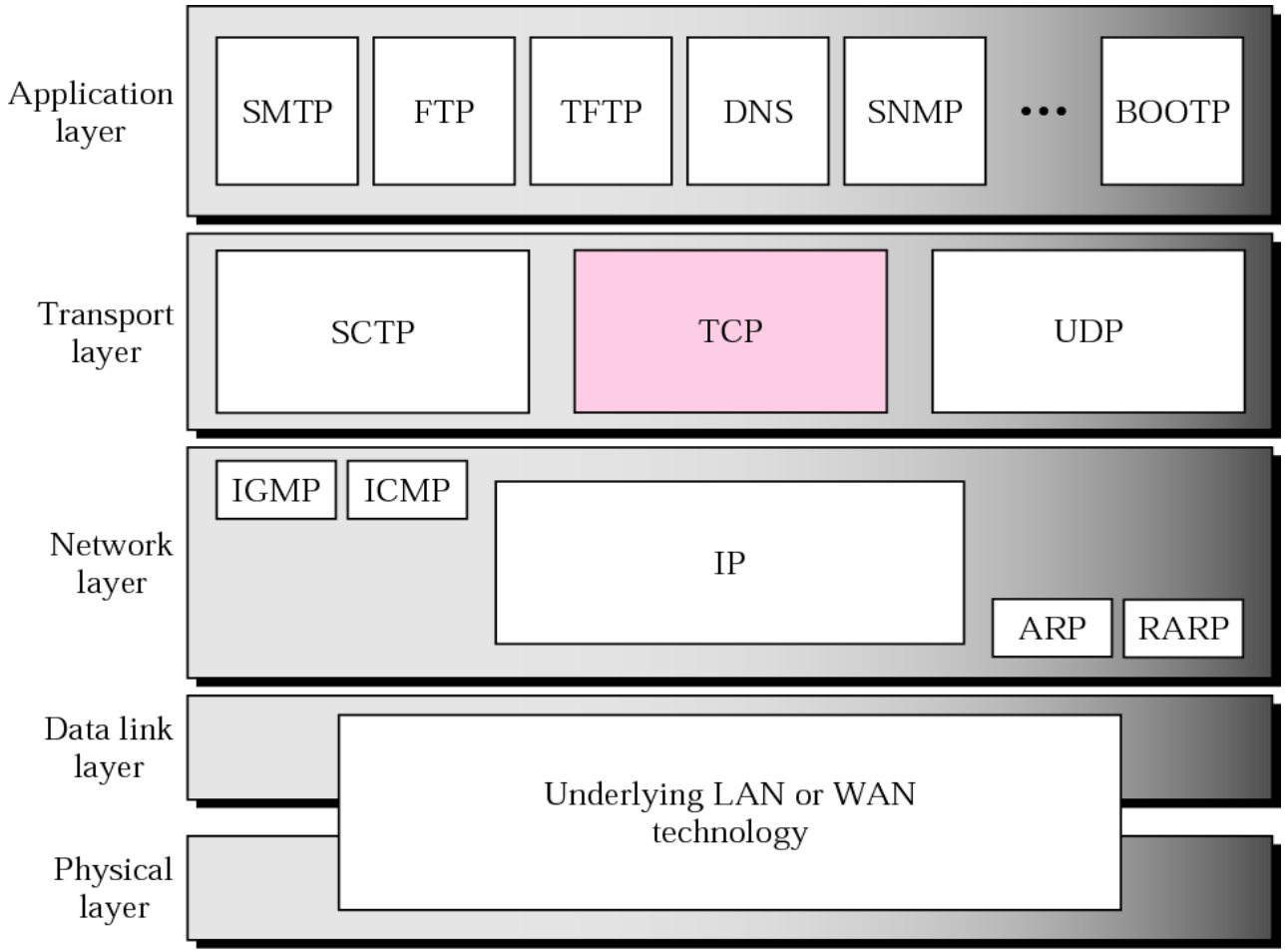
Growth of the Internet

- The Internet has grown tremendously.
- The Internet is still growing.
  - New Protocols
  - New Technology
  - Increasing use of multimedia

Internet Congestion Control Components



Transmission Control Protocol - TCP



TCP Services (1/2)

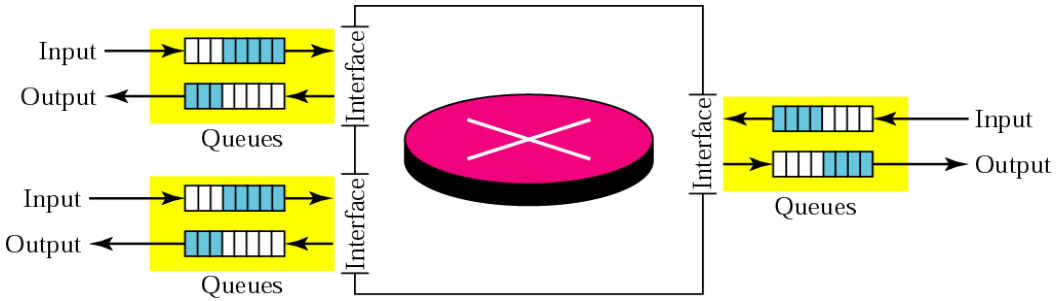
- Process-to-process communication

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

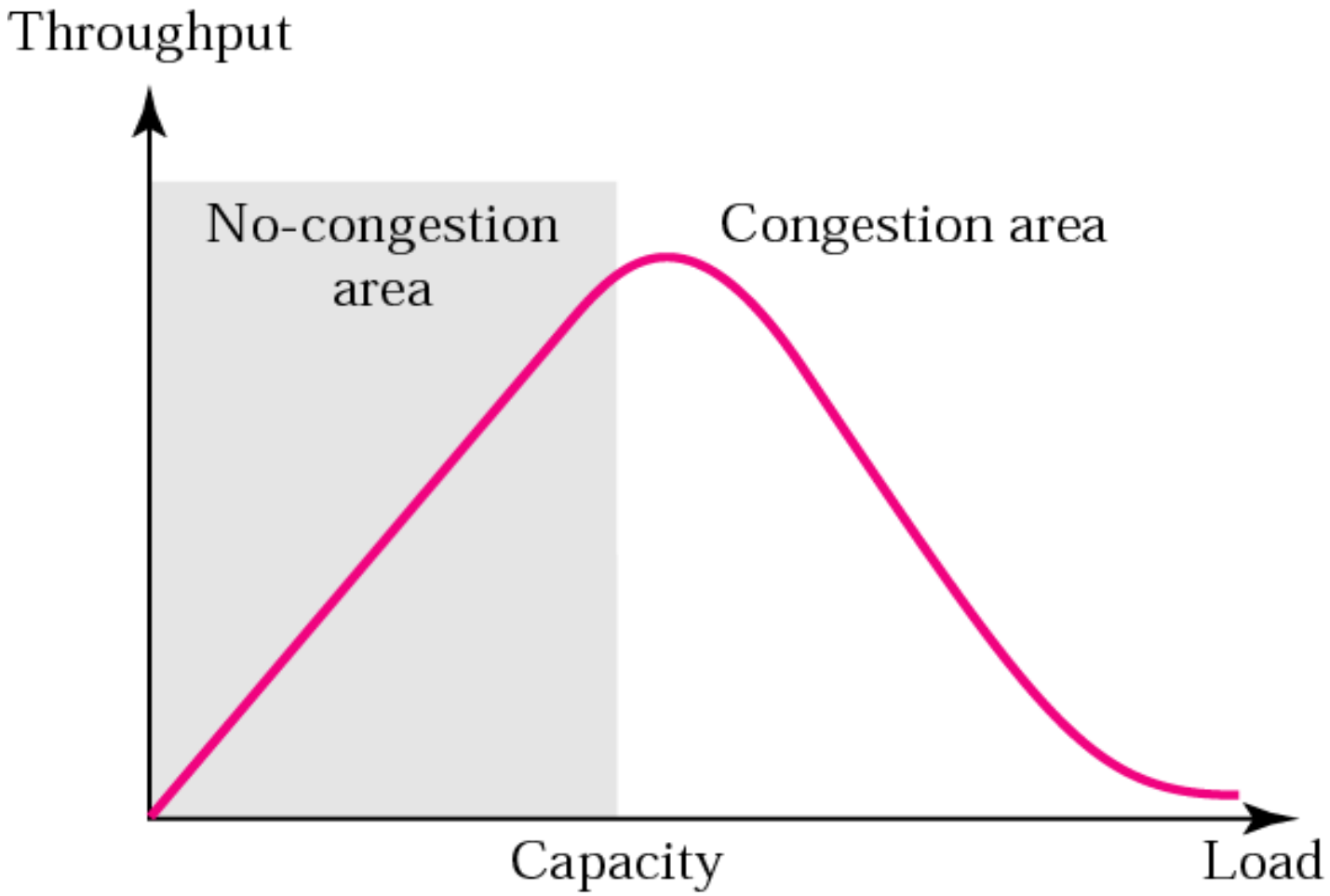
Well-known ports used by TCP

TCP Services (2/2)

- Virtual connection
  - Three-way handshaking
  - Flow control
    - It regulates the amount of data a source can send before receiving an ACK from the destination.
  - Error control
    - Checksum, ACK, retransmission time-out.
  - Congestion control
    - The mechanism to control the congestion and keep the load below the capacity.

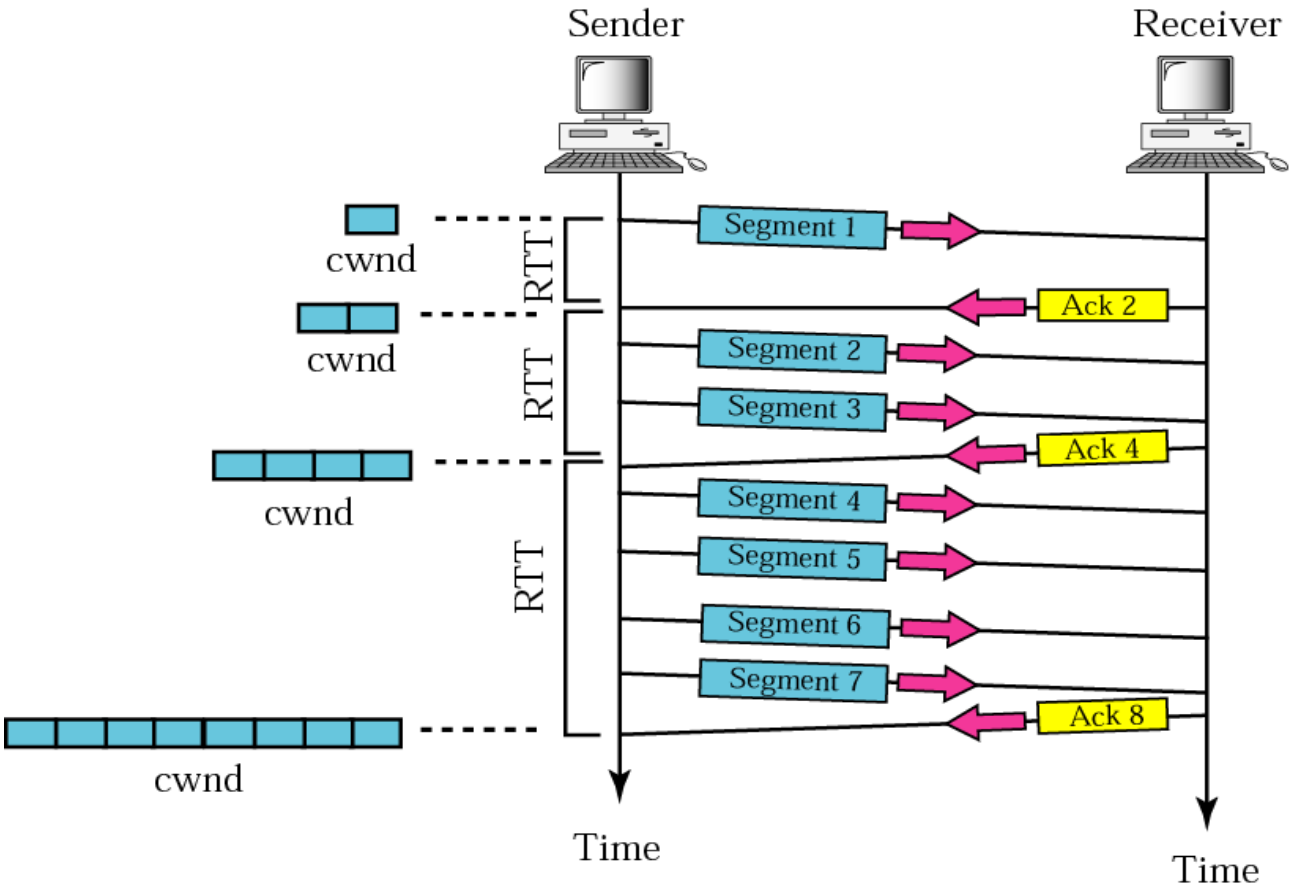


Throughput versus Network Load



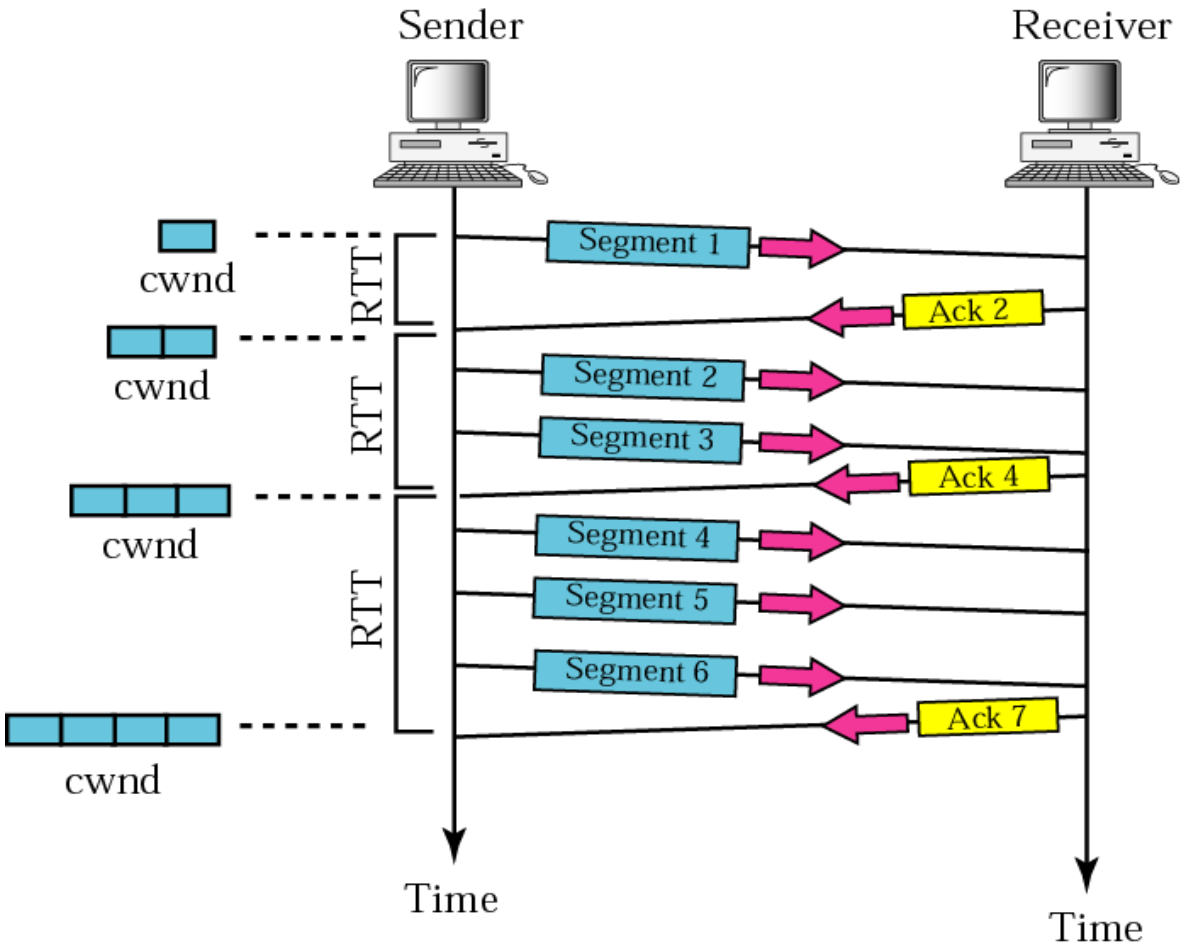
Congestion Control in TCP (1/5)

- Congestion window
- Slow start: exponential increase



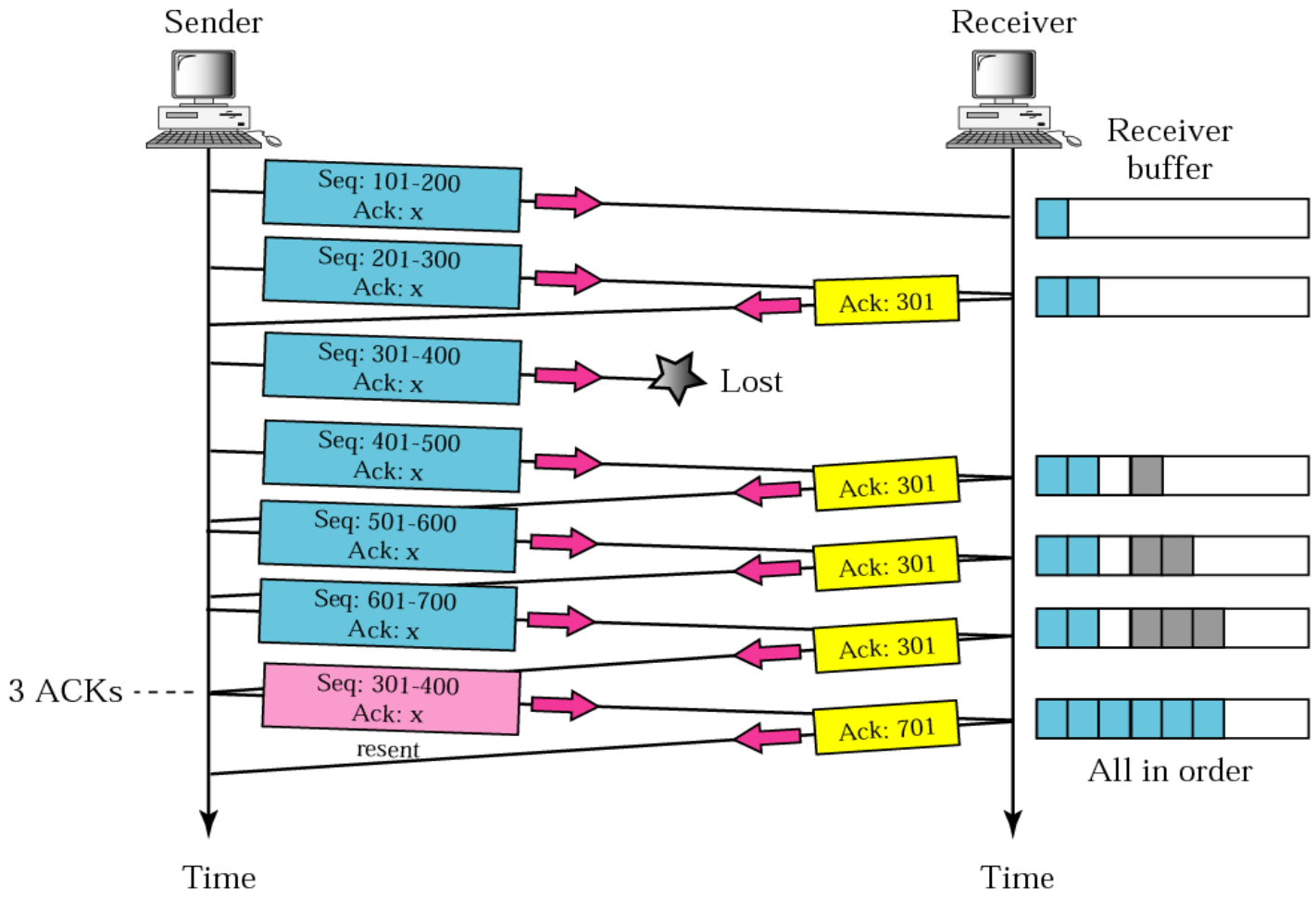
Congestion Control in TCP (2/5)

- Congestion avoidance: additive increase



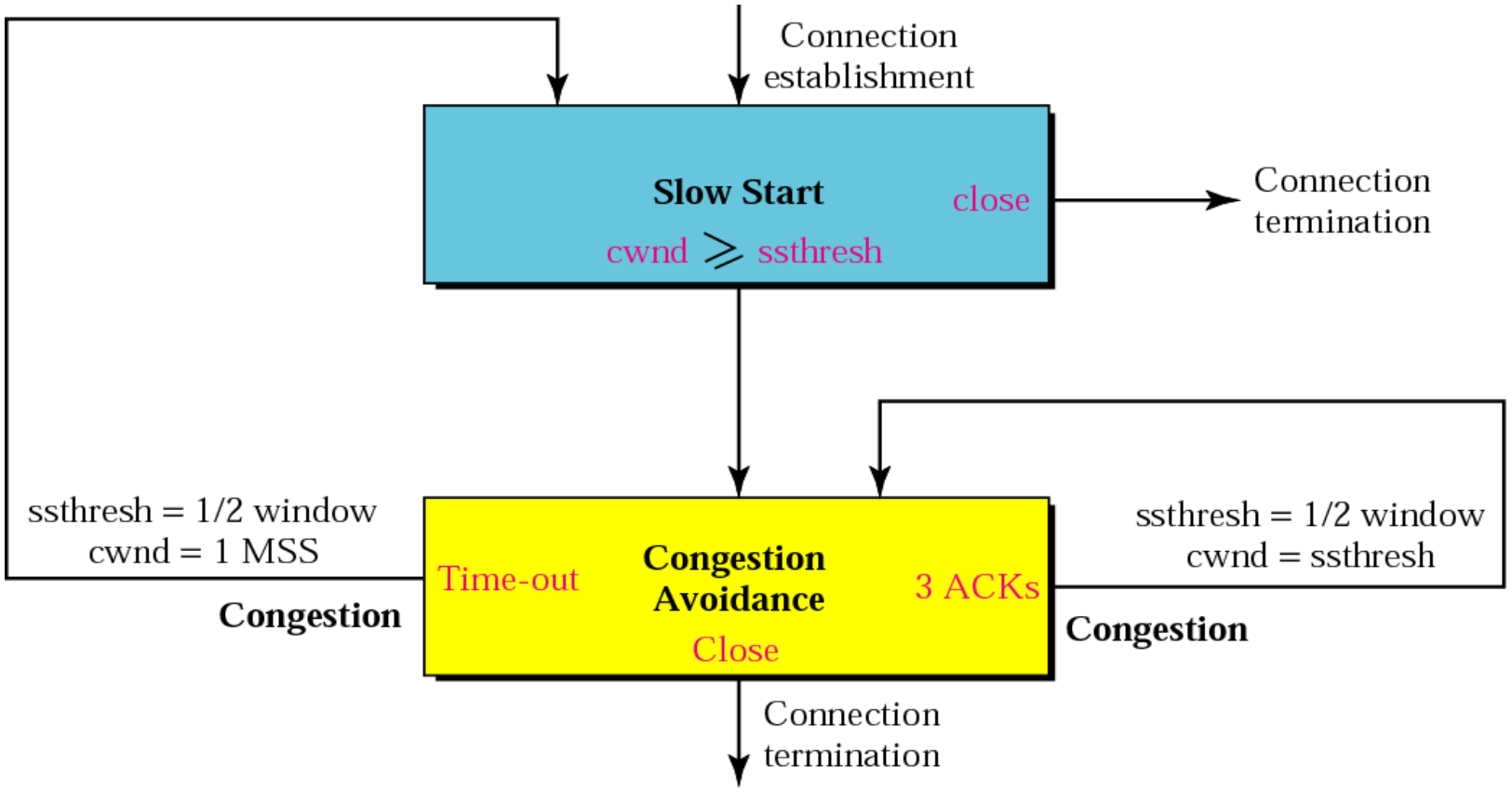
Congestion Control in TCP (3/5)

- Fast retransmission



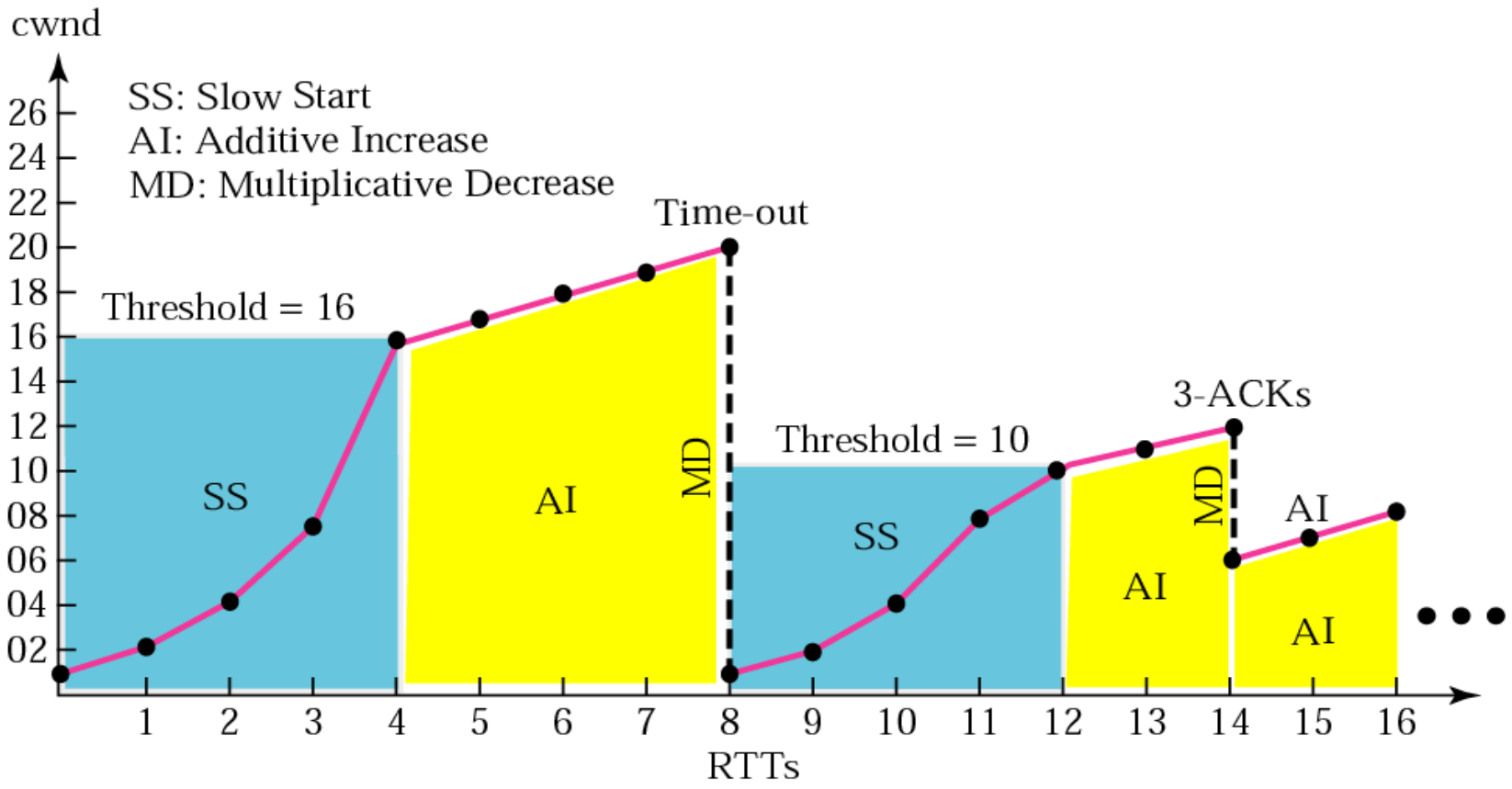
Congestion Control in TCP (4/5)

- TCP congestion control policy summary



Congestion Control in TCP (5/5)

- Congestion example



Evolutions of TCP

- RFC 793 (1981)
  - A simple sliding window flow control mechanism.
- Tahoe (1988)
  - Slow start, congestion avoidance, fast retransmit.
- Reno (1990)
  - Fast recovery.
- New Reno (1995)
  - Refine the fast retransmit.

Advanced Enhancements

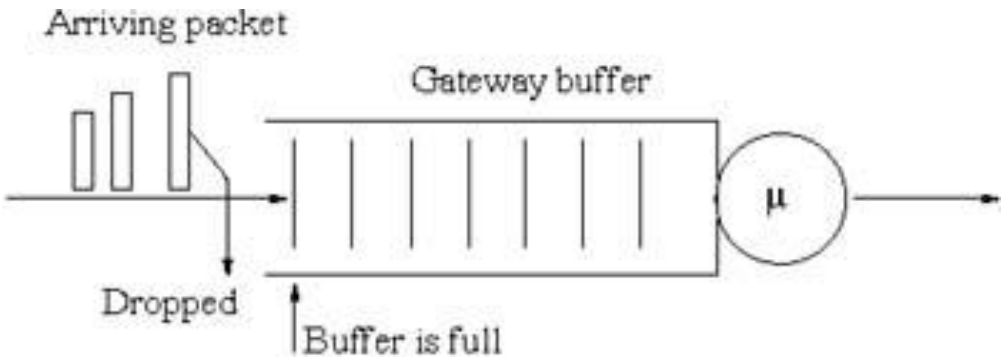
- TCP with selective acknowledgments (1996)
- TCP Vegas (1994)
- Compound TCP (2006)
- And …

Active Queue Management - AQM

- **Queue management** is defined as the algorithms that manage the length of queues by dropping packets when necessary or appropriate.
- Why Queue Management is needed?
- Passive Queue Management (**PQM**)
  - Drop packets when router buffer gets full.
- Active Queue Management (**AQM**)
  - Employ preventive packet drop before the router buffer gets full.

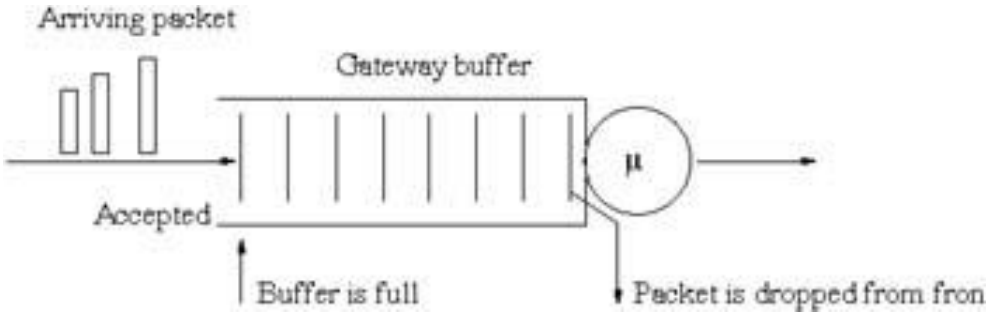
Passive Queue Management (1/2)

- Two states of PQM:
  - No packet drop
  - 100% packet drop
- Drop-Tail
  - Drop packets from the tail of the queue.
  - All arriving packets are dropped once the queue size reaches a certain threshold.

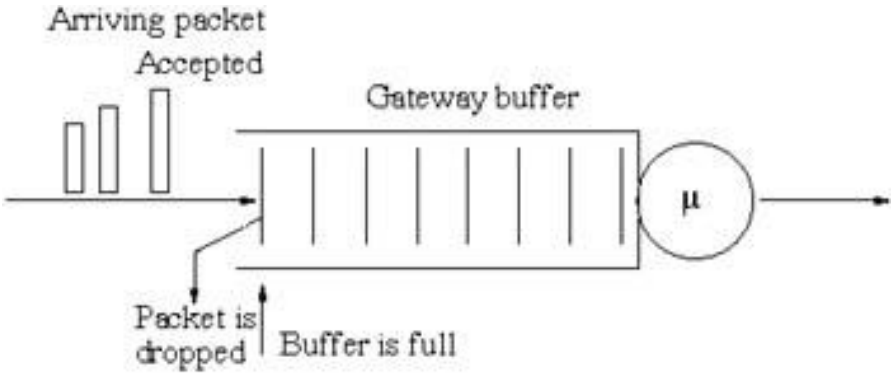


Passive Queue Management (2/2)

- Drop-Front
  - Drop the packet in the buffer with the oldest age.

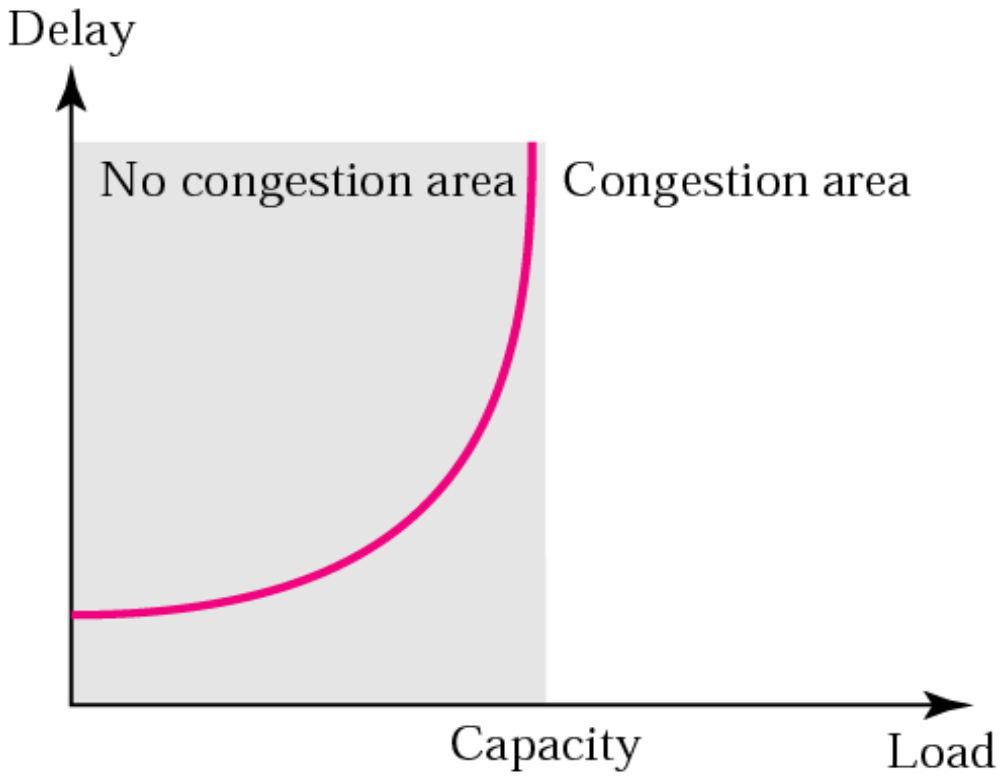


- Push-Out
  - The latest buffered packet is pushed out from the queue.



Problem with PQM

- Lock out
- Global synchronization
- Full queue

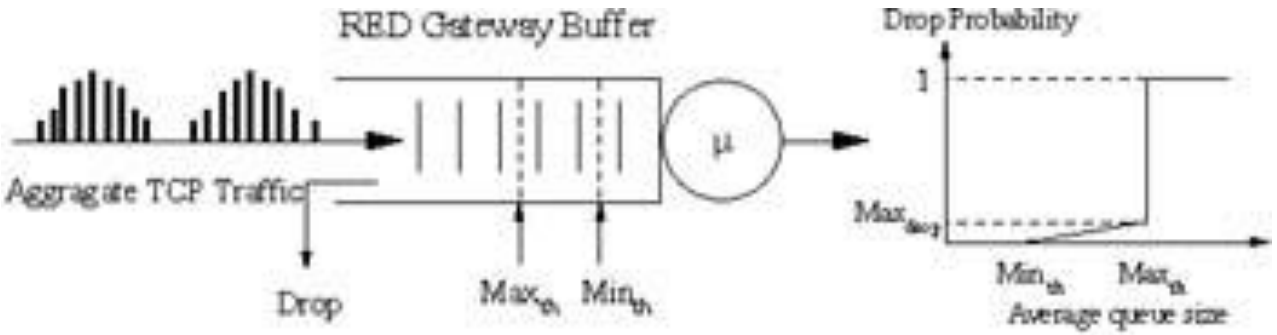


Active Queue Management

- Provide preventive measures to manage a buffer to eliminate problems associated with PQM.
- Characteristics:
  - Preventive random packet drop is performed before the buffer is full
  - The probability of preventive packet drop increases with the increasing level of congestion.
- Goals:
  - Reduce dropped packets
  - Support low-delay interactive services
  - Improve the fairness

Random Early Detection (RED)

- A router maintains two thresholds:
  - Min_{th}:
    - Accept all packets until the queue reaches Min_{th}
    - Drop packets with a linear drop probability when the queue is greater than Min_{th}
  - Max_{th}:
    - All packets are dropped with probability of Max_{drop} when the queue exceeds this threshold



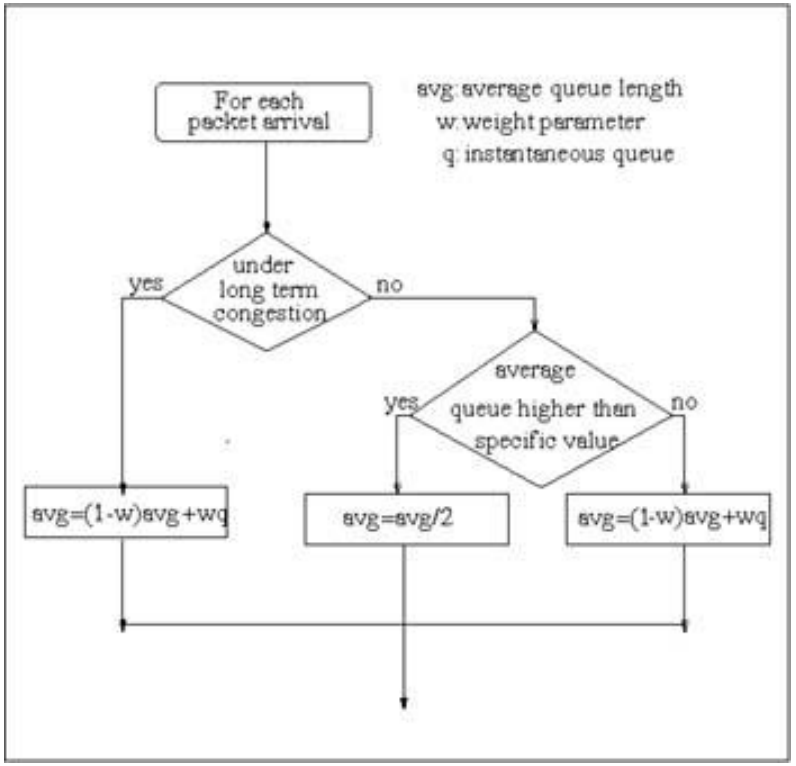
Selection of Max_{drop} for RED

- Selection of Max_{drop} significantly affects the performance of RED
  - Too small: Active packet drops not enough to prevent global synchronisation
  - Too large: Decreases the throughput
  - Optimal value depends on number of connections, round trip time, etc.
- Selection of an optimal value for Max_{drop} remains an open issue

Calculation of the Average Queue Length

- The average queue length controls the active packet drop.
  - Accumulate short term congestion and trace long term congestion.
- Average queue length works as a low pass filter (LPF).

LPF/ODA



RED Variants

- Aggregate control
  - Modifying the calculation of the control variable and/or drop function.
  - Determines packet drop probability.
  - SRED, DSRED, REM, BLUE, …
- Per-flow control
  - Configuring and setting RED's parameters.
  - Addresses fairness problem.
  - FRED, FB-RED, XRED, …

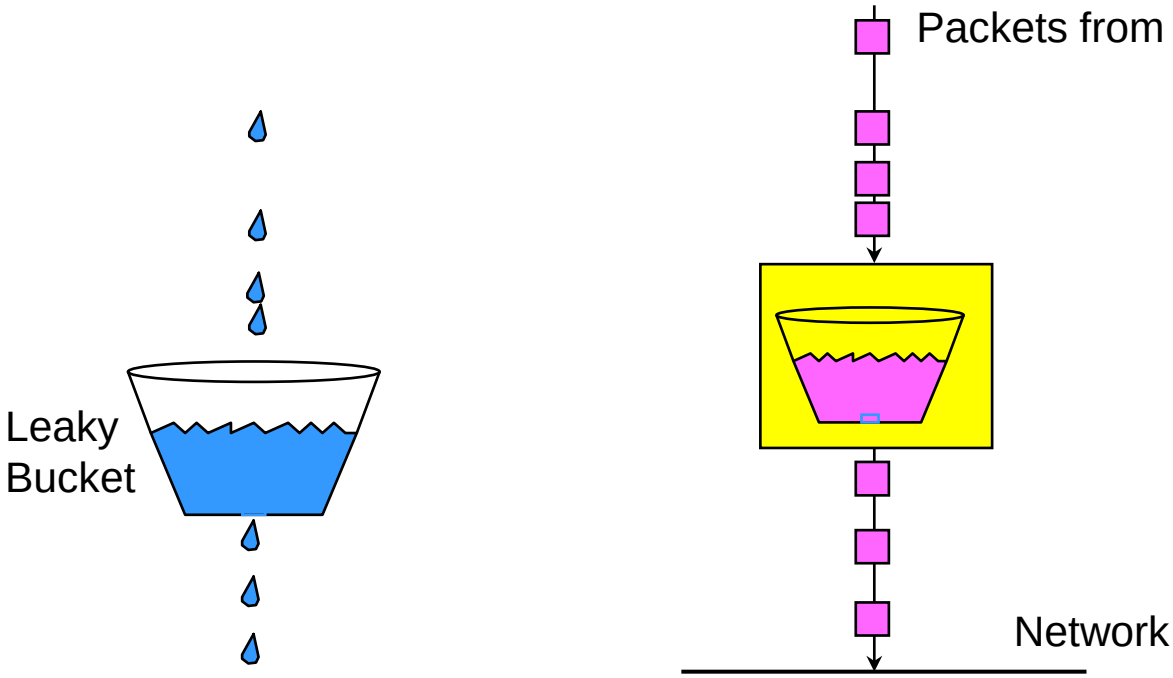
Admission Control

- Why admission control?
- Expressing QoS can be done in flow specification.

Characteristics of the Input	**Service Required**
■ maximum data unit size (bytes) ■ Token bucket rate (bytes/sec) ■ Toke bucket size (bytes) ■ Maximum transmission rate (bytes/sec)	■ Loss sensitivity (bytes) ■ Loss interval (μsec) ■ Burst loss sensitivity (data units) ■ Minimum delay noticed (μsec) ■ Maximum delay variation (μsec) ■ Quality of guarantee

Leaky Bucket (1/2)

- At the host-network interface, allow packets into the network at a constant rate.
- Packets may be generated in a bursty manner, but after they pass through the leaky bucket, they enter the network evenly spaced.



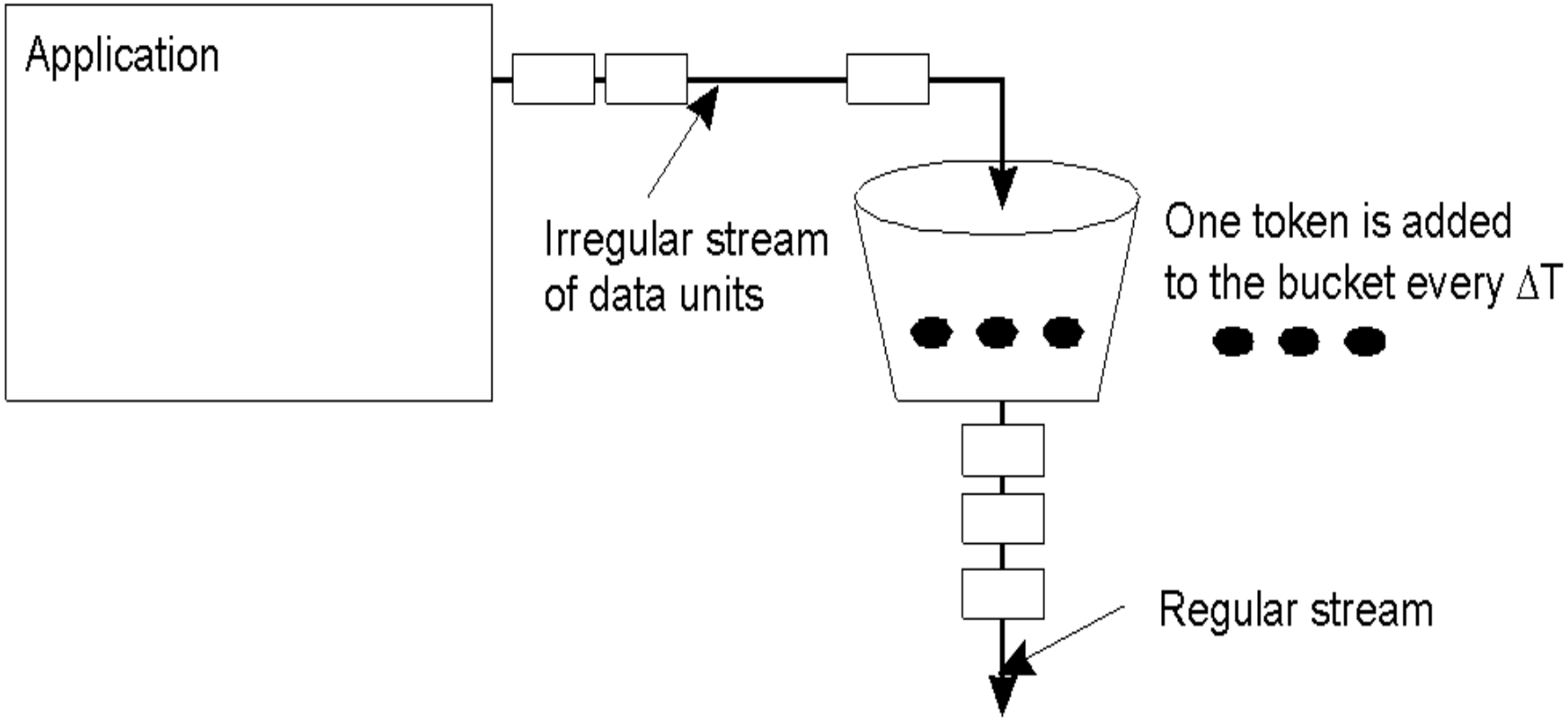
Leaky Bucket (2/2)

- The leaky bucket is a “traffic shaper”: It changes the characteristics of packet stream.
- Traffic shaping makes traffic more manageable and more predictable.
- In some cases, we may want to allow short bursts of packets to enter the network without smoothing them out.
- For this purpose we use a token bucket, which is a modified leaky bucket.

Token Bucket (1/2)

- The bucket holds tokens instead of packets.
- Tokens are generated at a constant rate.
- When a packet arrives at the token bucket, it is transmitted if there is a token available. Otherwise it is buffered until a token becomes available.
- Tokens are buffered in a bucket with limited capacity.
  - Bucket is full, tokens are dropped.
- Data are passed to network in a relatively constant rate.
- Allows for some bursty traffic from the application.

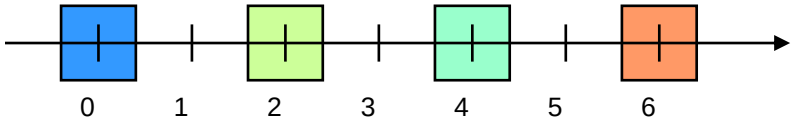
Token Bucket (2/2)



Token Bucket vs. Leaky Bucket (1/2)

Case 1: Short burst arrivals

Arrival time at bucket

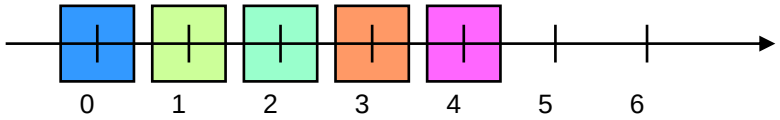


Departure time from a leaky bucket
Leaky bucket rate = 1 packet / 2 time units
Leaky bucket size = 2 packets

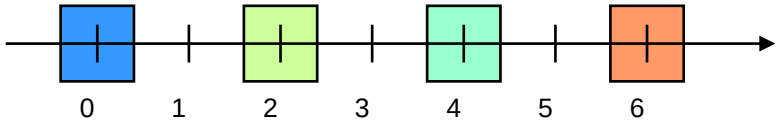
Departure time from a token bucket
Token bucket rate = 1 tokens / 2 time units
Token bucket size = 2 tokens
(There are two tokens in the bucket at time 0.)

Token Bucket vs. Leaky Bucket (2/2)

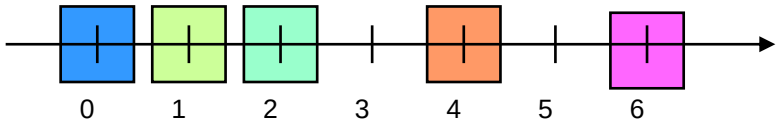
Case 2: Large burst arrivals



Arrival time at bucket



Departure time from a leaky bucket
Leaky bucket rate = 1 packet / 2 time units
Leaky bucket size = 2 packets



Departure time from a token bucket
Token bucket rate = 1 token / 2 time units
Token bucket size = 2 tokens
(There are two tokens in the bucket at time 0.)

Conclusions

- How congestion control in the Internet?
- Challenges
  - Increasing traffic volume
  - Varied QoS requirements
  - Complicated networking environment