Traffic Management

Masoud Sabaei

Associate professor

Department of Computer Engineering,

Amirkabir University of Technology

Traffic Management

Vehicular traffic management

- Traffic lights & signals control flow of traffic in city street system
- Objective is to maximize flow with tolerable delays
- Priority Services
  - Police sirens
  - Cavalcade for dignitaries
  - Bus & High-usage lanes
  - Trucks allowed only at night

Packet traffic management

- Multiplexing & access mechanisms to control flow of packet traffic
- Objective is make efficient use of network resources & deliver QoS
- Priority
  - Fault-recovery packets
  - Real-time traffic
  - Enterprise (high-revenue) traffic
  - High bandwidth traffic

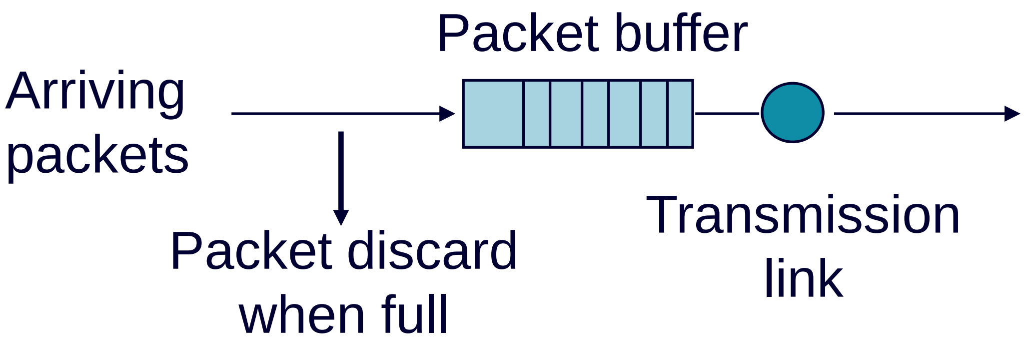
Time Scales & Granularities

- Packet Level
  - Queueing & scheduling at multiplexing points
  - Determines relative performance offered to packets over a short time scale (microseconds)
- Flow Level
  - Management of traffic flows & resource allocation to ensure delivery of QoS (milliseconds to seconds)
  - Matching traffic flows to resources available; congestion control
- Flow-Aggregate Level
  - Routing of aggregate traffic flows across the network for efficient utilization of resources and meeting of service levels
  - "Traffic Engineering", at scale of minutes to days

End-to-End QoS

- A packet traversing network encounters delay and possible loss at various multiplexing points
- End-to-end performance is accumulation of per-hop performances

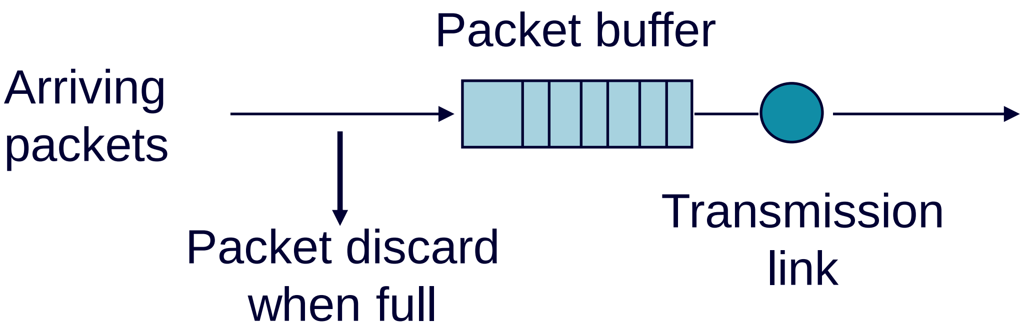
FIFO Queueing



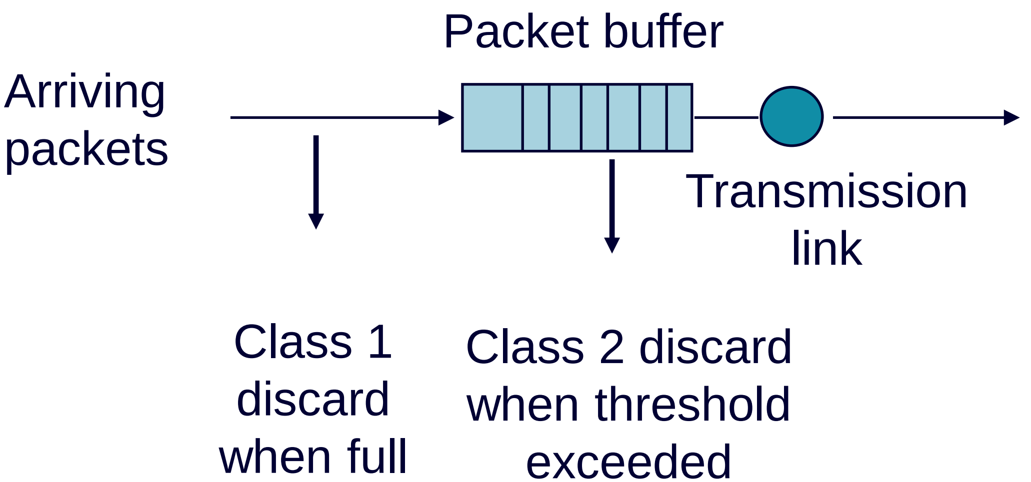
- All packet flows share the same buffer
- Transmission Discipline: First-In, First-Out
- Buffering Discipline: Discard arriving packets if buffer is full (Alternative: random discard; pushout head-of-line, i.e. oldest, packet)

FIFO Queueing with Discard Priority

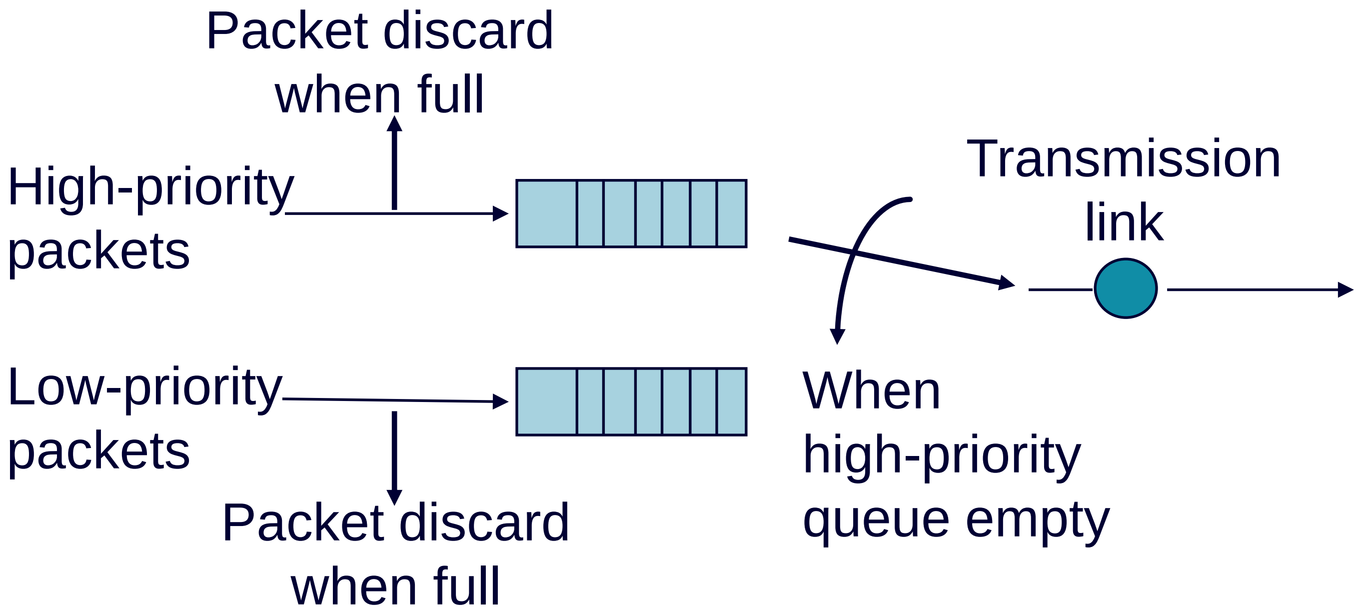
(a)



(b)

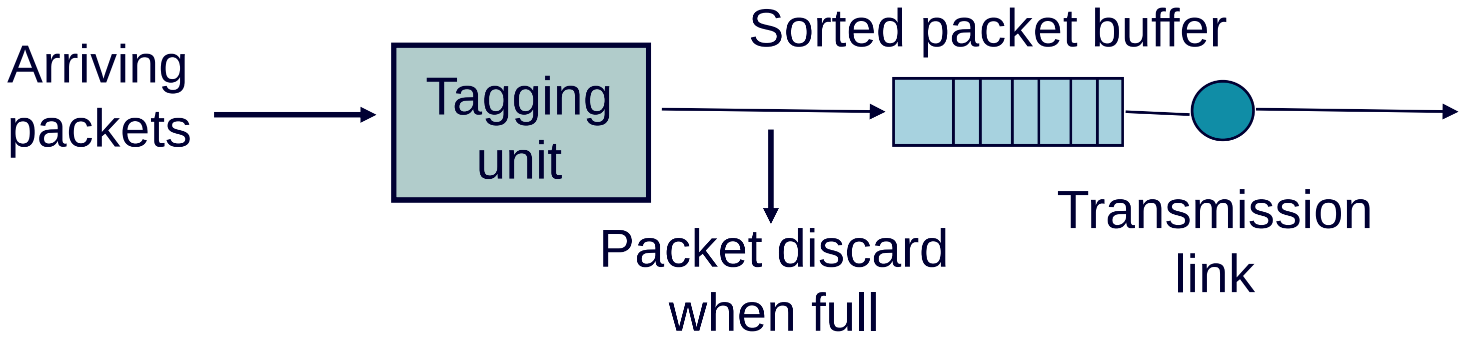


HOL Priority Queueing



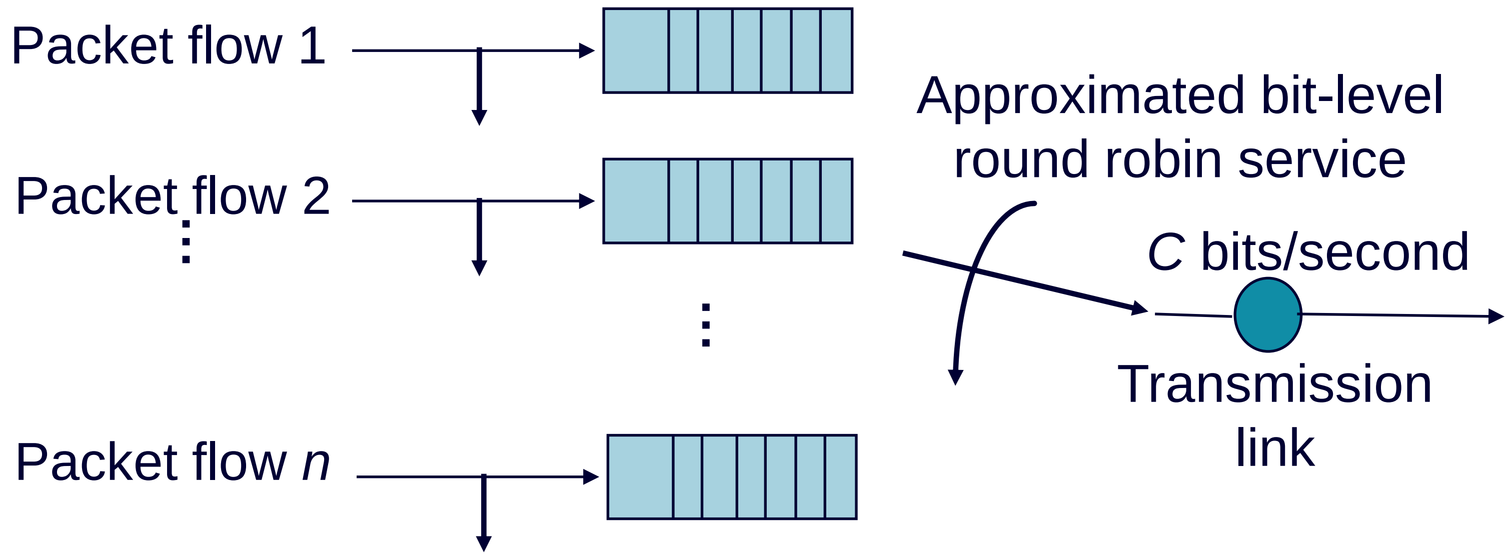
- High priority queue serviced until empty
- High priority queue has lower waiting time
- Buffers can be dimensioned for different loss probabilities
- Surge in high priority queue can cause low priority queue to saturate

Earliest Due Date Scheduling



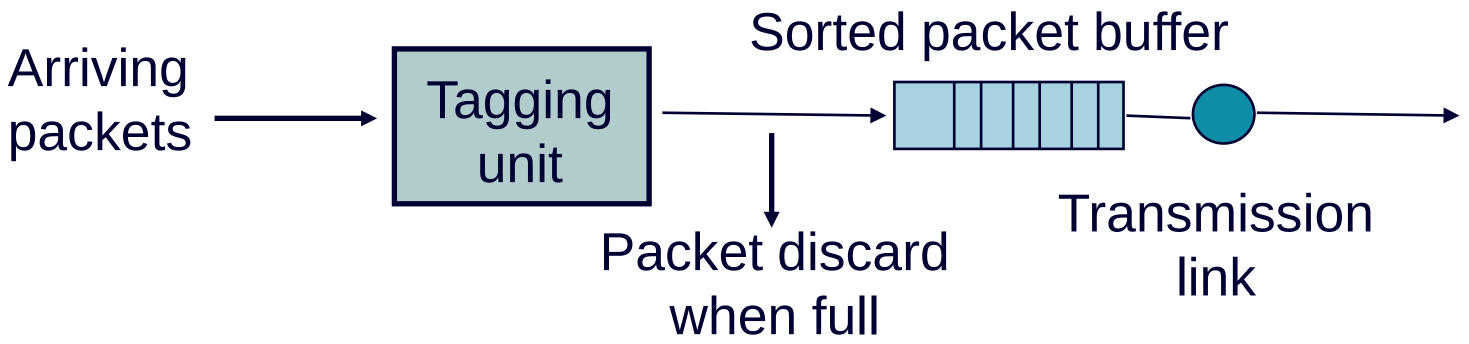
- Queue in order of “due date”
  - packets requiring low delay get earlier due date
  - packets without delay get indefinite or very long due dates

Fair Queueing / Generalized Processor Sharing



- Each flow has its own logical queue: prevents hogging; allows differential loss probabilities
- C bits/sec allocated equally among non-empty queues
  - transmission rate = C / n(t), where n(t)=# non-empty queues
- Idealized system assumes fluid flow from queues
- Implementation requires approximation: simulate fluid system; sort packets according to completion time in ideal system

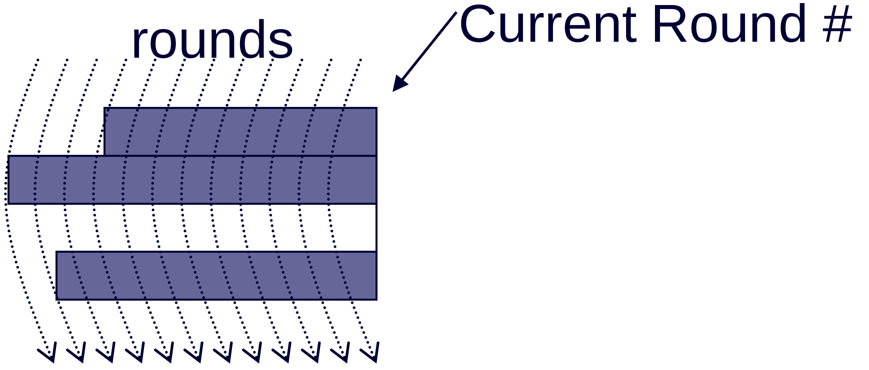
Packetized GPS/WFQ



- Compute packet completion time in ideal system
  - add tag to packet
  - sort packet in queue according to tag
  - serve according to HOL

Bit-by-Bit Fair Queueing

- Assume n flows, n queues
- 1 round = 1 cycle serving all n queues
- If each queue gets 1 bit per cycle, then 1 round = # active queues
- Round number = number of cycles of service that have been completed

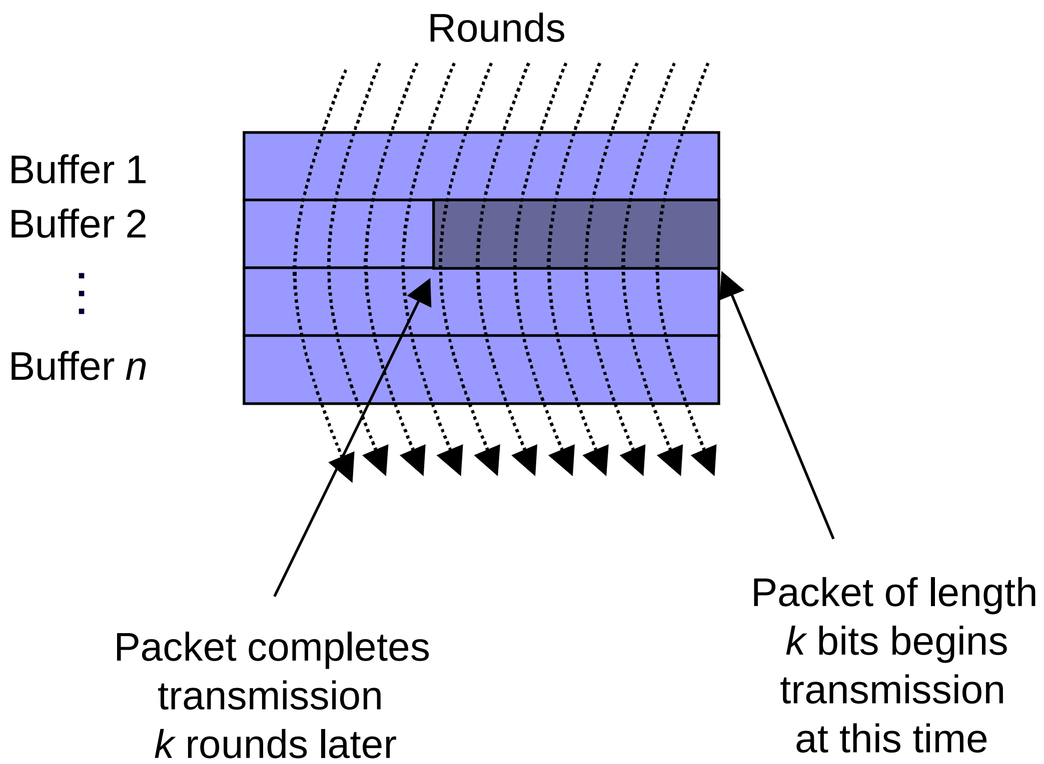


- If packet arrives to idle queue:

  Finishing time = round number + packet size in bits
- If packet arrives to active queue:

Finishing time = finishing time of last packet in queue + packet size

Number of rounds = Number of bit transmission opportunities



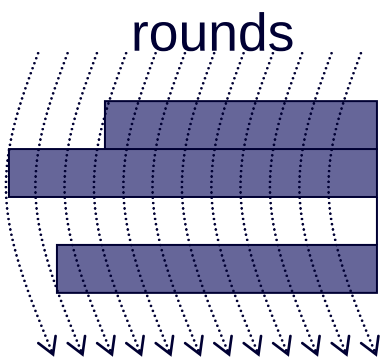
Differential Service:

If a traffic flow is to receive twice as much bandwidth as a regular flow, then its packet completion time would be half

Computing the Finishing Time

- $F(i,k,t)$ = finish time of kth packet that arrives at time t to flow i
- $P(i,k,t)$ = size of kth packet that arrives at time t to flow i
- $R(t)$ = round number at time t

Generalize so $R(t)$ continuous, not discrete



$R(t)$ grows at rate inversely proportional to $n(t)$

- Fair Queueing:

$$F(i,k,t) = \max\{F(i,k-1,t), R(t)\} + P(i,k,t)$$

- Weighted Fair Queueing:

$$F(i,k,t) = \max\{F(i,k-1,t), R(t)\} + P(i,k,t)/w_i$$

WFQ and Packet QoS

- WFQ and its many variations form the basis for providing QoS in packet networks
- Very high-speed implementations available, up to 10 Gbps and possibly higher
- WFQ must be combined with other mechanisms to provide end-to-end QoS (next section)

Buffer Management

- Packet drop strategy: Which packet to drop when buffers full
- Fairness: protect behaving sources from misbehaving sources
- Aggregation:
  - Per-flow buffers protect flows from misbehaving flows
  - Full aggregation provides no protection
  - Aggregation into classes provided intermediate protection
- Drop priorities:
  - Drop packets from buffer according to priorities
  - Maximizes network utilization & application QoS
  - Examples: layered video, policing at network edge
- Controlling sources at the edge

Early or Overloaded Drop

Random early detection:

- drop pkts if short-term avg of queue exceeds threshold
- pkt drop probability increases linearly with queue length
- mark offending pkts
- improves performance of cooperating TCP sources
- increases loss probability of misbehaving sources

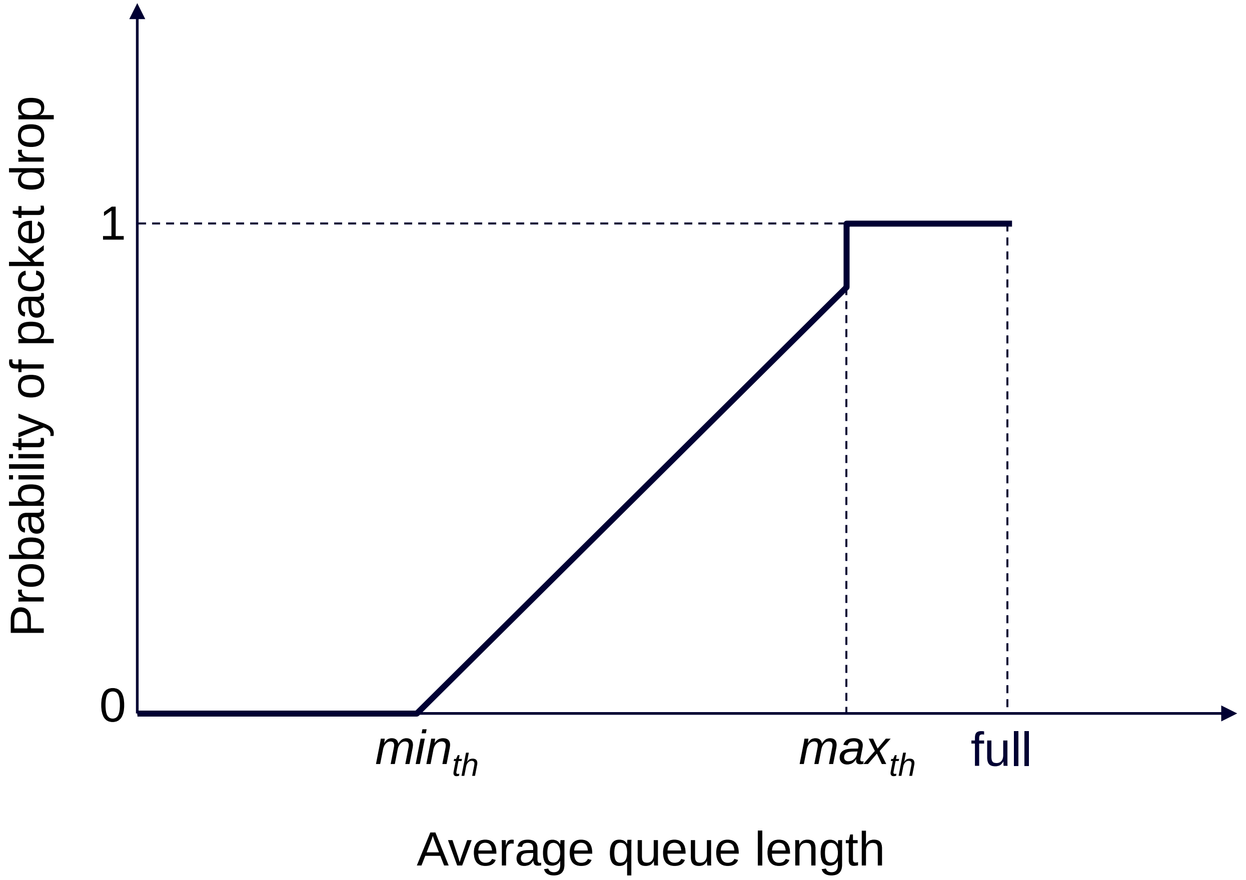
Random Early Detection (RED)

- Packets produced by TCP will reduce input rate in response to network congestion
- Early drop: discard packets before buffers are full
- Random drop causes some sources to reduce rate before others, causing gradual reduction in aggregate input rate

Algorithm:

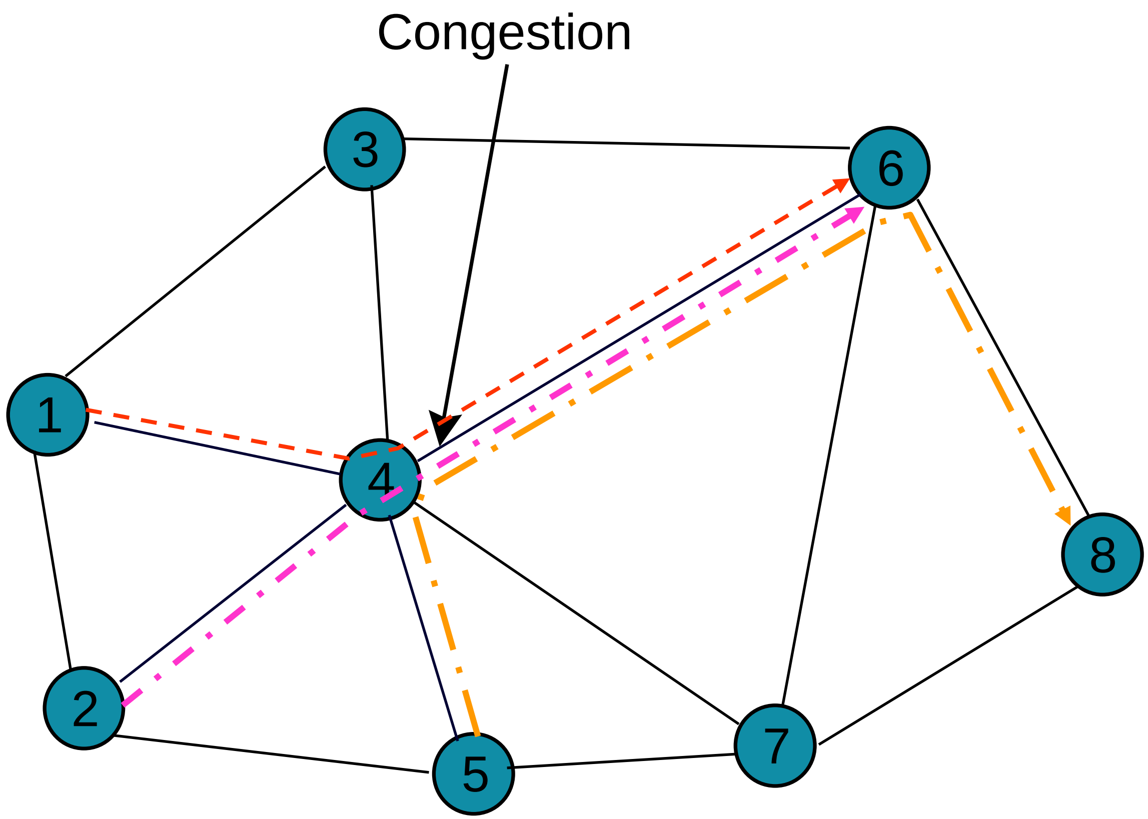
- Maintain running average of queue length
- If Q_{avg} < minthreshold, do nothing
- If Q_{avg} > maxthreshold, drop packet
- If in between, drop packet according to probability
- Flows that send more packets are more likely to have packets dropped

Packet Drop Profile in RED



Traffic Management At the Flow Level

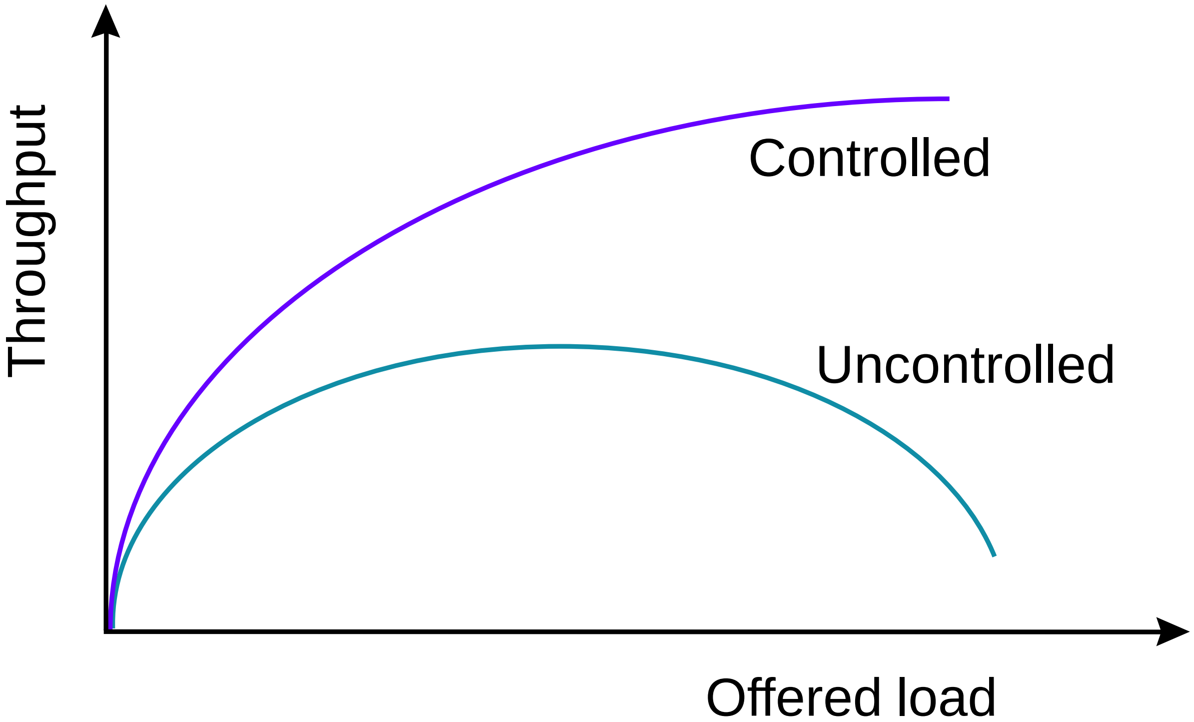
Congestion occurs when a surge of traffic overloads network resources



Approaches to Congestion Control:

- Preventive Approaches: Scheduling & Reservations
- Reactive Approaches: Detect & Throttle/Discard

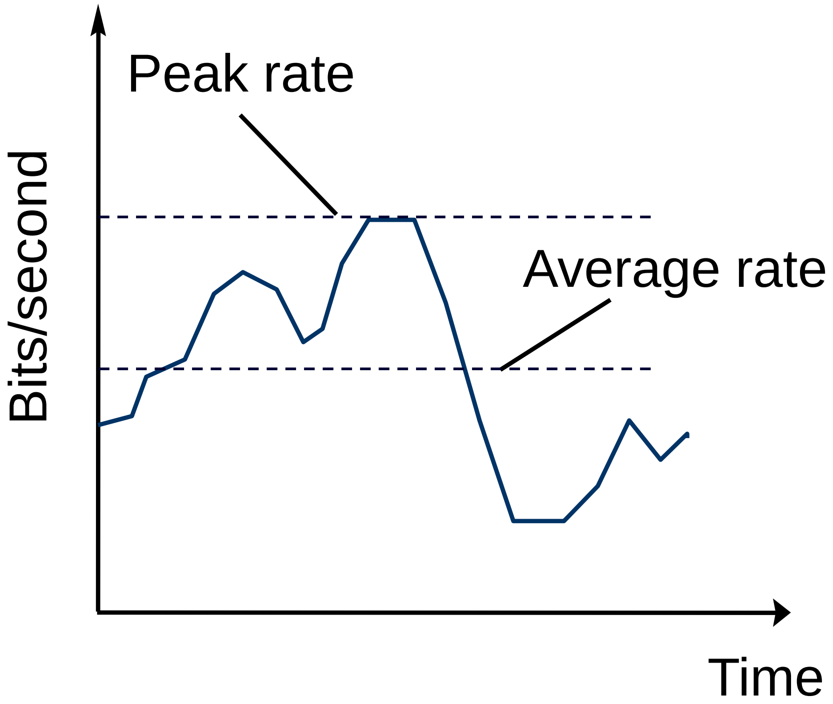
Ideal effect of congestion control:
Resources used efficiently up to capacity available



Open-Loop Control

- Network performance is guaranteed to all traffic flows that have been admitted into the network
- Initially for connection-oriented networks
- Key Mechanisms
  - Admission Control
  - Policing
  - Traffic Shaping

Admission Control



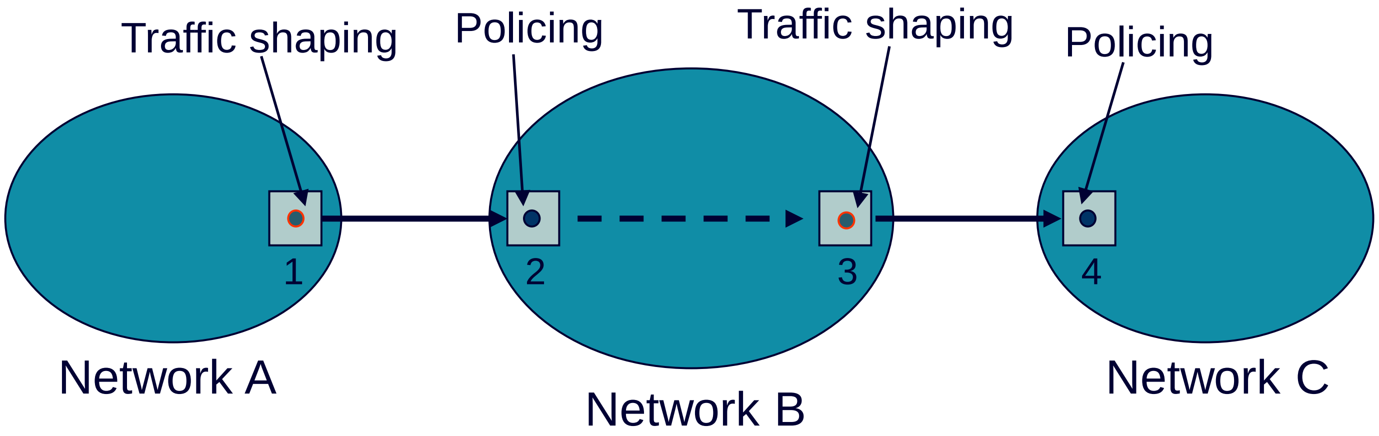
Typical bit rate demanded by a variable bit rate information source

- Flows negotiate contract with network
- Specify requirements:
  - Peak, Avg., Min Bit rate
  - Maximum burst size
  - Delay, Loss requirement
- Network computes resources needed
  - “Effective” bandwidth
- If flow accepted, network allocates resources to ensure QoS delivered as long as source conforms to contract

Policing

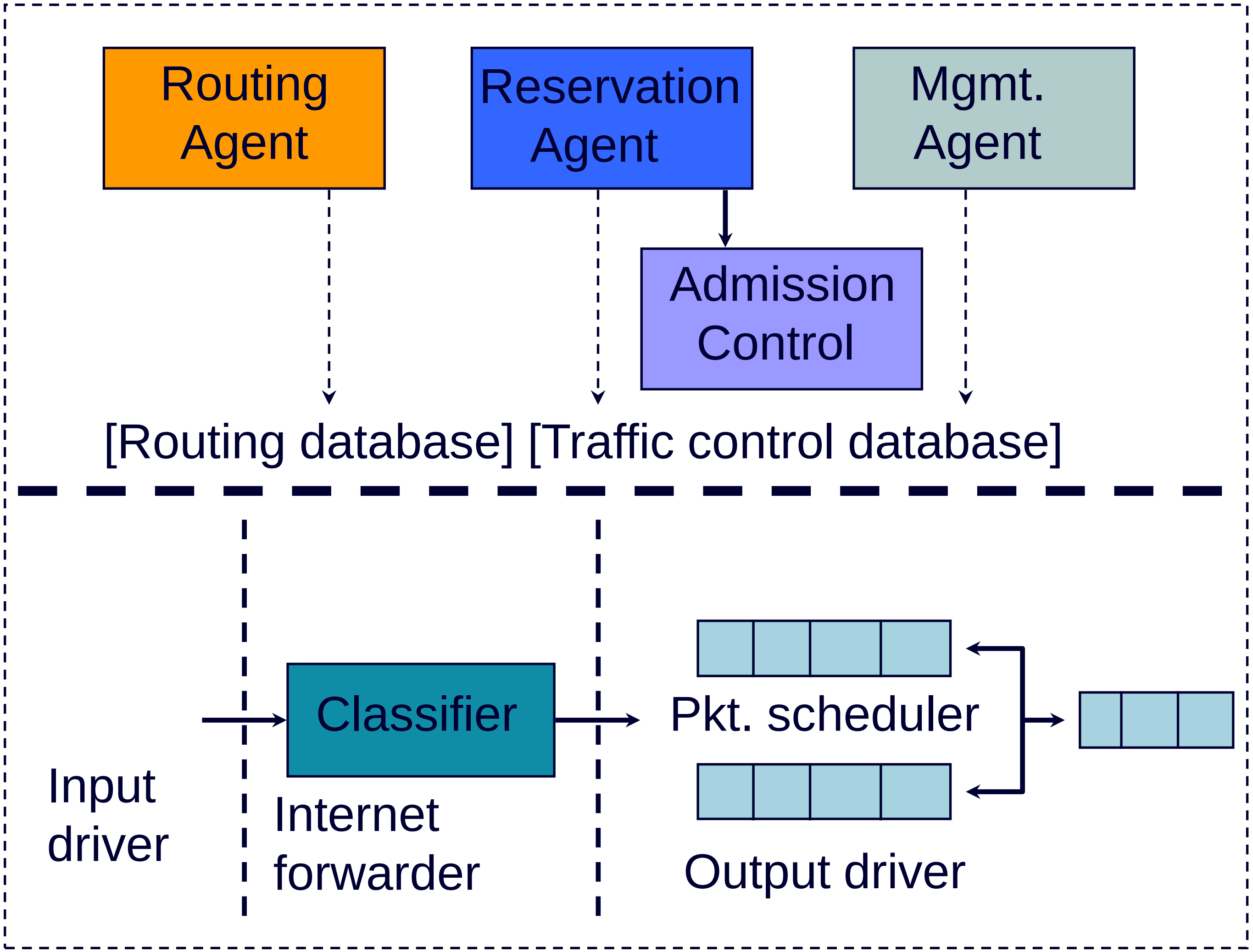
- Network monitors traffic flows continuously to ensure they meet their traffic contract
- When a packet violates the contract, network can discard or tag the packet giving it lower priority
- If congestion occurs, tagged packets are discarded first
- *Leaky Bucket Algorithm* is the most commonly used policing mechanism
  - Bucket has specified leak rate for average contracted rate
  - Bucket has specified depth to accommodate variations in arrival rate
  - Arriving packet is *conforming* if it does not result in overflow

Traffic Shaping



- Networks police the incoming traffic flow
- *Traffic shaping* is used to ensure that a packet stream conforms to specific parameters
- Networks can shape their traffic prior to passing it to another network

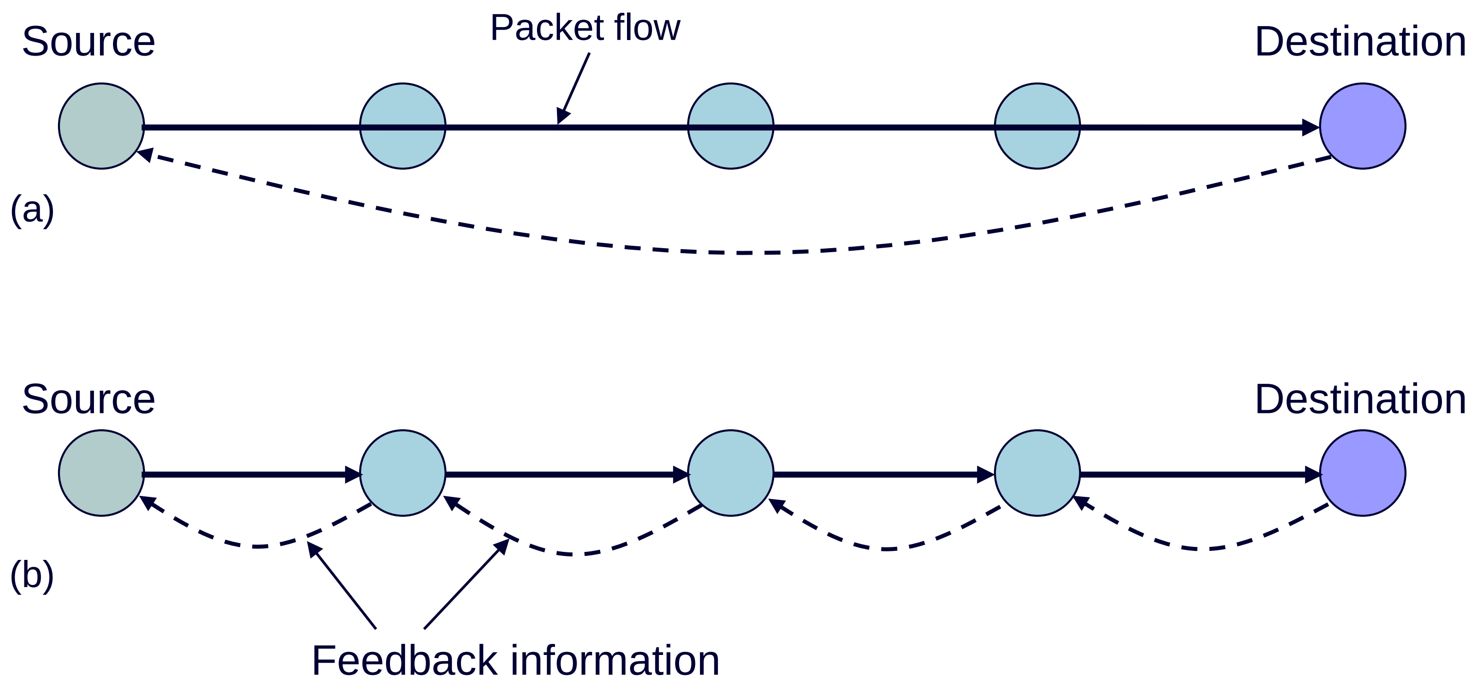
Current View of Router Function



Closed-Loop Flow Control

- Congestion control
  - feedback information to regulate flow from sources into network
  - Based on buffer content, link utilization, etc.
  - Examples: TCP at transport layer; congestion control at ATM level
- End-to-end vs. Hop-by-hop
  - Delay in effecting control
- Implicit vs. Explicit Feedback
  - Source deduces congestion from observed behavior
  - Routers/switches generate messages alerting to congestion

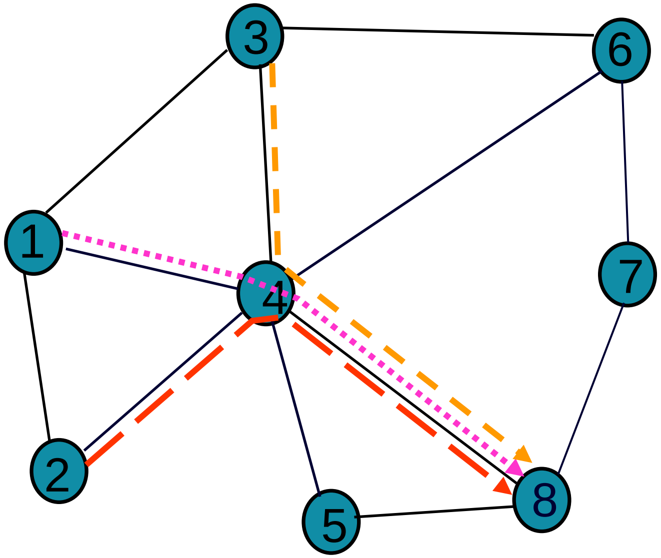
End-to-End vs. Hop-by-Hop Congestion Control



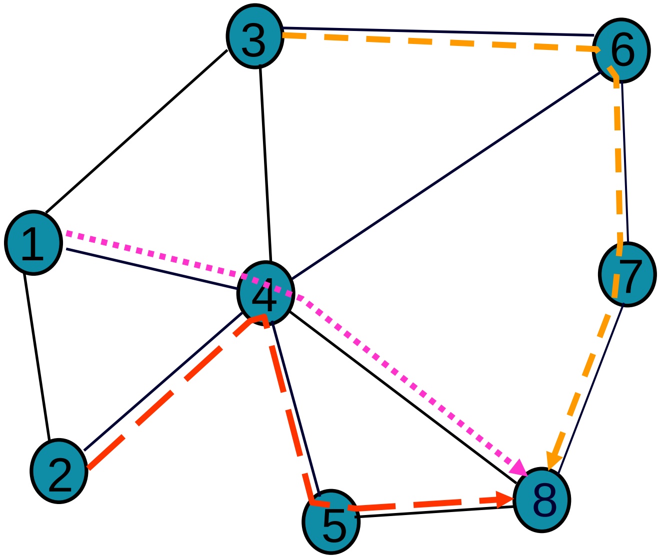
Traffic Engineering

- Management exerted at flow aggregate level
- Distribution of flows in network to achieve efficient utilization of resources (bandwidth)
- Shortest path algorithm to route a given flow not enough
  - Does not take into account requirements of a flow, e.g. bandwidth requirement
  - Does not take account interplay between different flows
- Must take into account aggregate demand from all flows

Traffic Engineering



(a)



(b)

Shortest path routing congests link 4 to 8

Better flow allocation distributes flows more uniformly