Packet-Switching Networks

Traffic Management

Packet Level

Flow Level

Flow-Aggregate Level

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

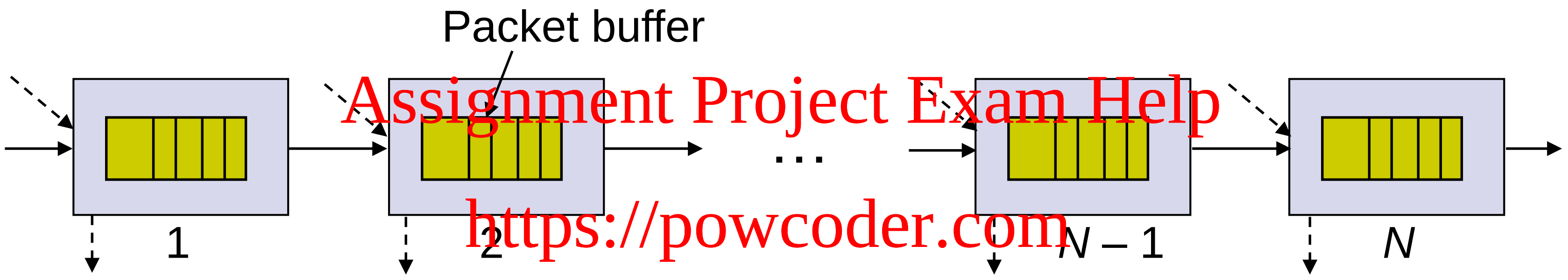
Time Scales & Granularities

- Packet Level
  - Queueing & scheduling at switches, routers
  - Determines relative performance offered to packets over a short time scale (microseconds)

- Flow Level
  - Management of individual traffic flows & resource allocation to ensure delivery of QoS (milliseconds to seconds)
  - Matching traffic flows to resources available; congestion control in case of too many packets for same resource

- Flow-Aggregate Level
  - Routing of aggregate traffic flows across the network for efficient utilization of resources and meeting of service levels
  - Called “Traffic Engineering”, at scale of minutes to days

End-to-End QoS

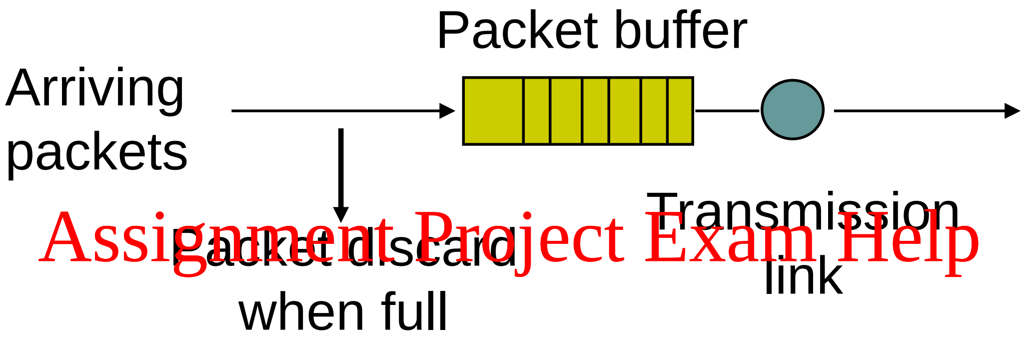


- A packet traversing network encounters delay and possible loss at various multiplexing points
- End-to-end performance is accumulation of per-hop performances

Scheduling & QoS

- End-to-End QoS & Resource Control
  - Buffer & bandwidth control → Performance
  - Admission control to regulate traffic level
  - End-to-end delay
- Scheduling Concepts
  - fairness/isolation
  - priority, aggregation,
- Packet Dropping
  - End-to-end probability pf packet loss
  - aggregation, drop priorities

FIFO Queueing

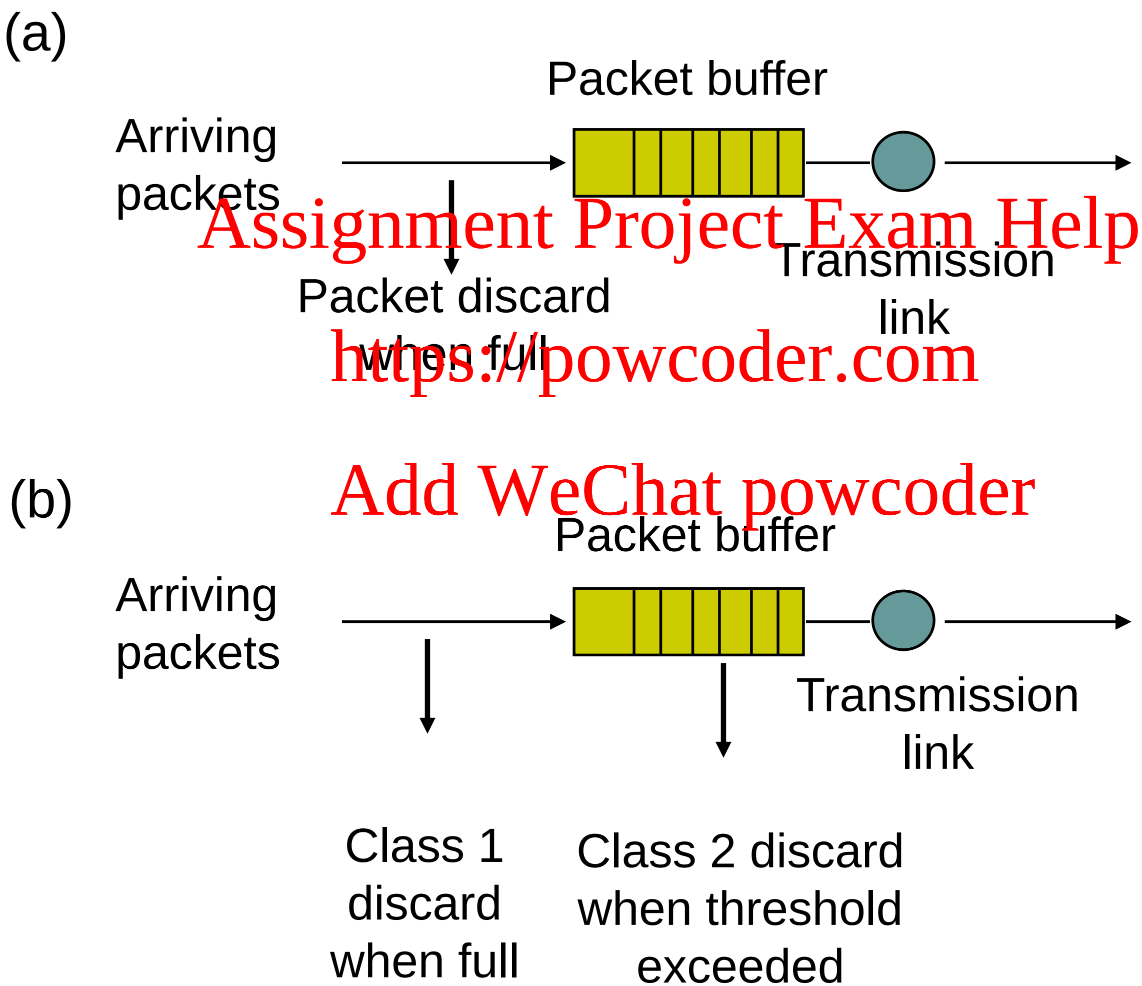


- All packet flows share the same buffer
- Queue Scheduling:
  - Transmission Discipline: First-In, First-Out
- Queue Management
  - Buffering Discipline: Discard arriving packets if buffer is full (Alternative: random discard; pushout head-of-line, i.e. oldest, packet)

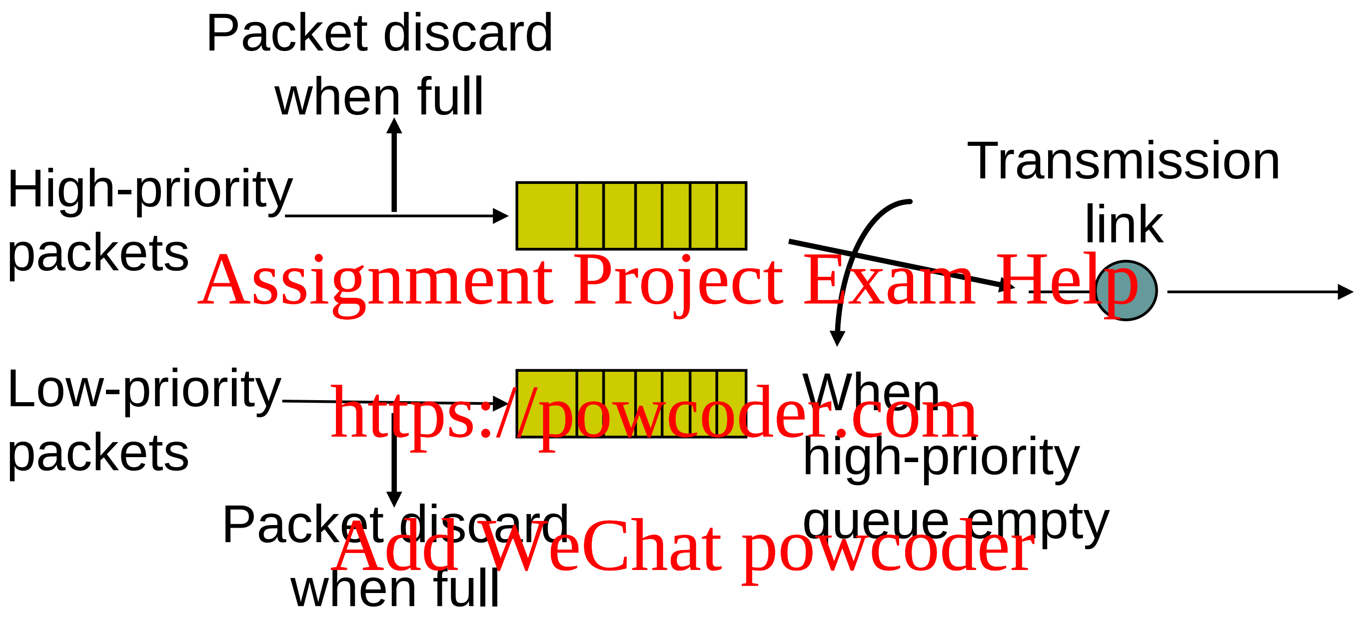
FIFO Queueing

- Cannot provide differential QoS to different packet flows
  - Different packet flows interact strongly
- Statistical delay guarantees via load control
  - Restrict number of flows allowed (connection admission control)
  - Difficult to determine performance delivered
- Finite buffer determines a maximum possible delay
- Buffer size determines loss probability
  - But depends on arrival & packet length statistics
- Variation: packet enqueueing based on queue thresholds
  - some packet flows encounter blocking before others
  - higher loss, lower delay

FIFO Queueing with Discard Priority



HOL Priority Queueing

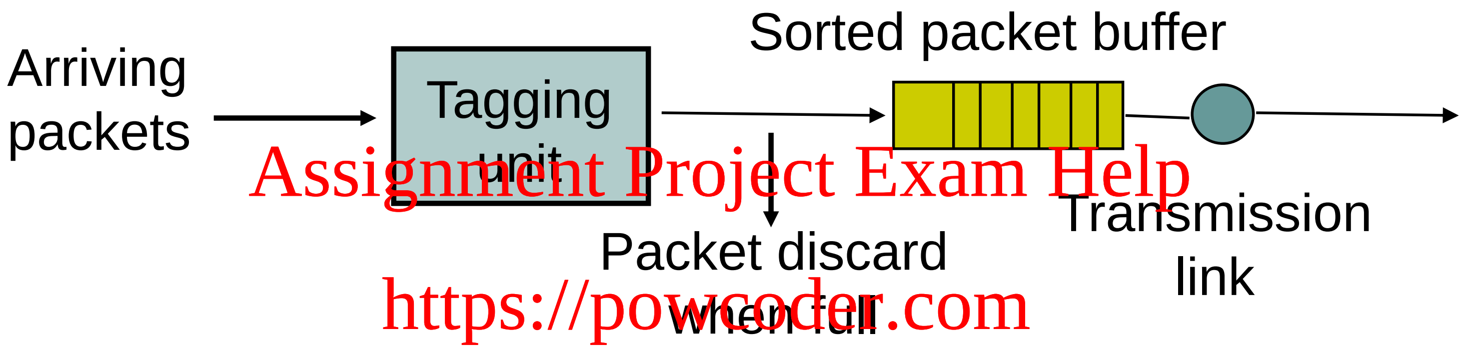


- High priority queue serviced until empty
- High priority queue has lower waiting time
- Buffers can be dimensioned for different loss probabilities
- Surge in high priority queue can cause low priority queue to saturate

HOL Priority Features

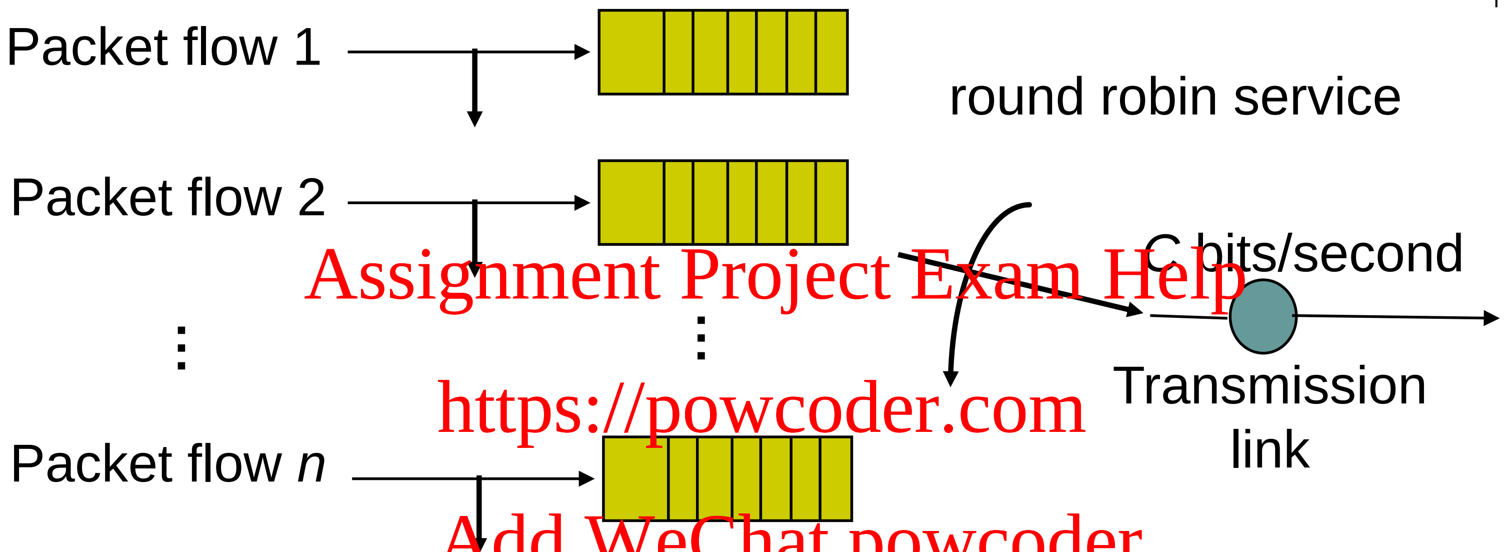
- Provides differential QoS
- Does not provide guaranteed access to bandwidth to lower priority classes
- Does not discriminate users of same priority
- High-priority classes can hog all of the bandwidth by sending excessive number of packets & starve lower priority classes
- Need to provide some isolation between classes

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



- Each flow has its own logical queue
- Fair queueing prevents hogging; allows equitable access to transmission bandwidth
- C bits/sec allocated equally among non-empty queues
  - transmission rate = C / n(t), where n(t)=# non-empty queues

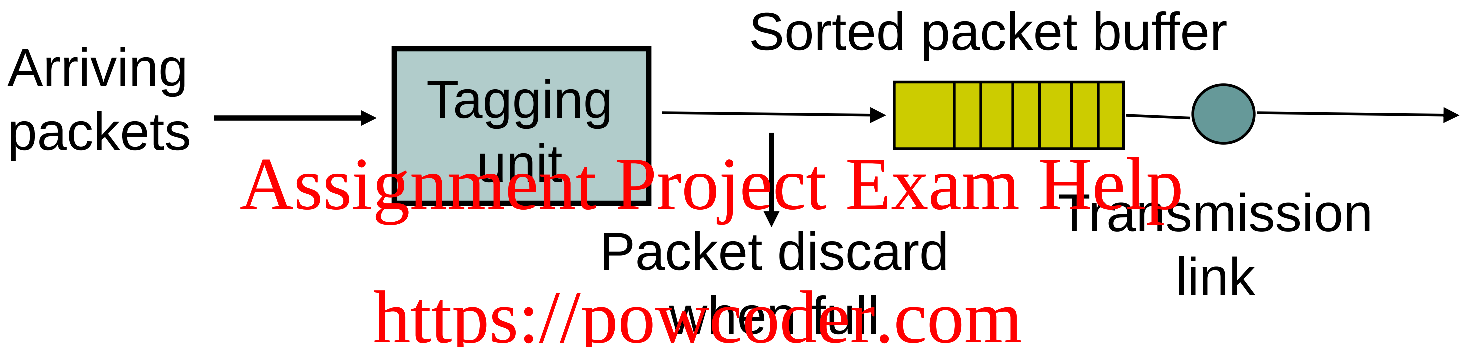
Bit-by-Bit Fair Queueing

- Service each nonempty buffer one bit at a time in round-robin fashion
- Decomposing the resulting bit stream into component packets require framing information and extra processing at output
- Easy to implement in ATM because all packets are of same length
  - Service nonempty buffers one ATM packet at a time in a round-robin fashion

Packet-by-Packet Fair Queueing

- Service each nonempty buffer one packet at a time in round-robin fashion
- Equal size packets in flows guarantees equal access to transmission bandwidth
- When packets have variable lengths – does not provide a fair allocation of transmission bandwidth
  - If packets of one flow are twice the size of packets in another flow, then first flow packets obtain twice the bandwidth

Packet-by-Packet Fair Queueing



- Better Approach is to compute packet completion time in ideal system
  - add tag to packet
  - sort packet in queue according to tag
  - serve according to HOL

Weighted Fair Queueing

- WFQ addresses the situation in which different users have different requirements
- Each user flow has a *weight* that determines its relative share of the bandwidth

  - If buffer 1 has weight 1 and buffer 2 has weight 3, then buffer 1 will receive 1/(1+3) = ¼ of the bandwidth and buffer 2 will receive ¾ of the bandwidth

- Bit-by-bit queueing would allocate 1 bit/round for buffer 1 and 3 bits/round for buffer 2
- Packet-by-packet queueing would allocate 1 packet/round for buffer 1 and 3 packets/round for buffer 2

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 and Chapter 10)

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

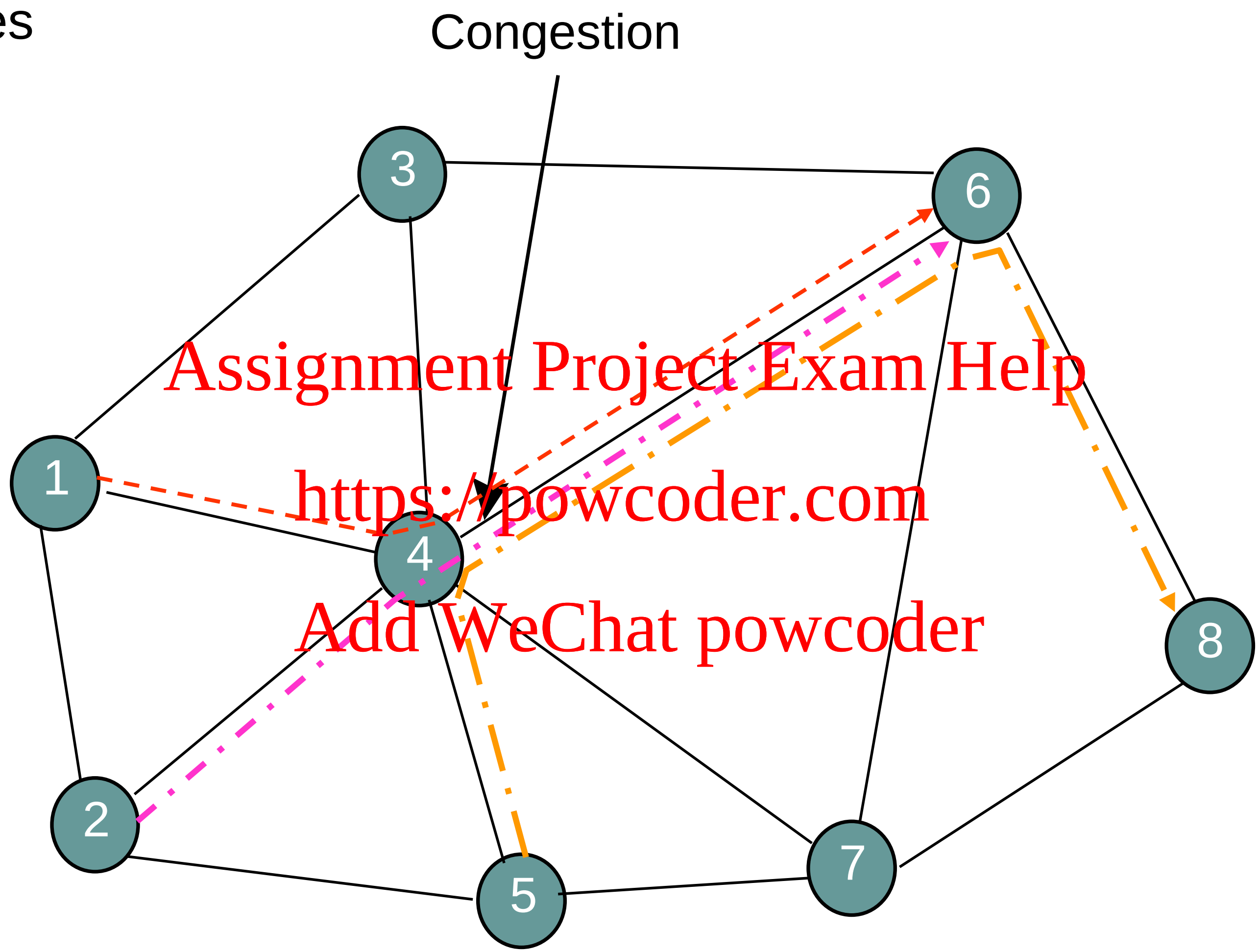
Algorithm:

- 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-Switching Networks

Traffic Management at the Flow Level

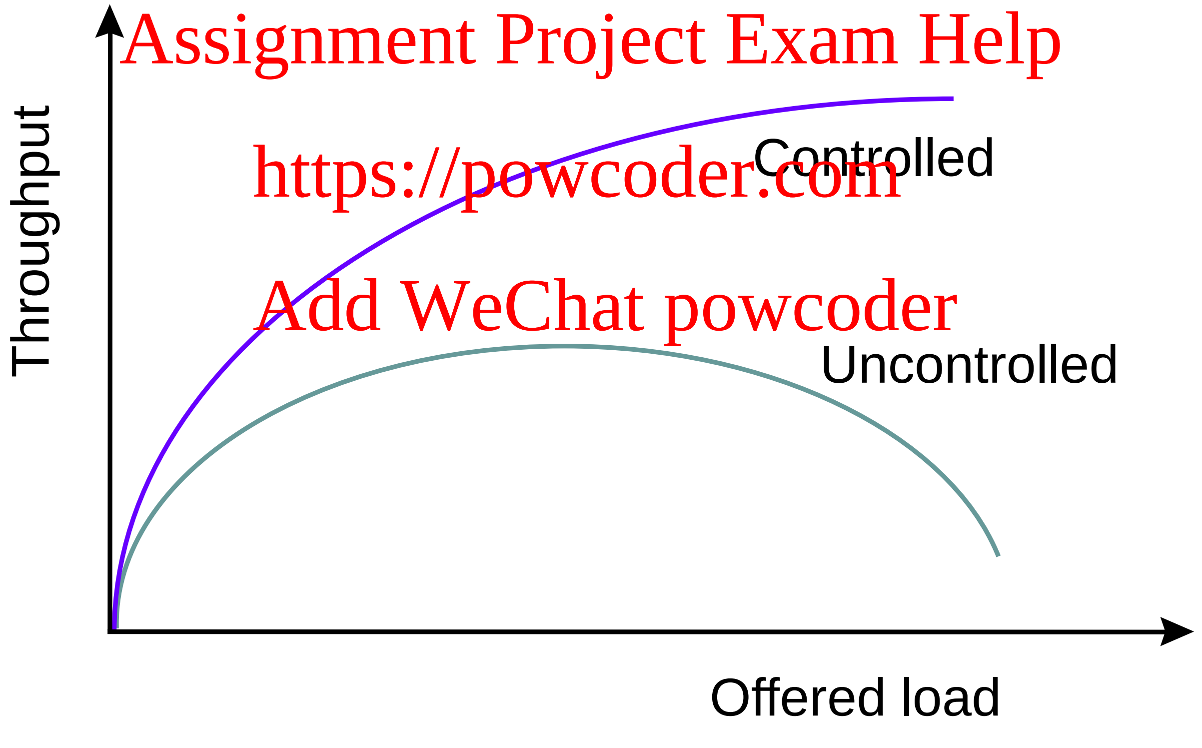
Congestion occurs when a surge of traffic overloads network resources



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

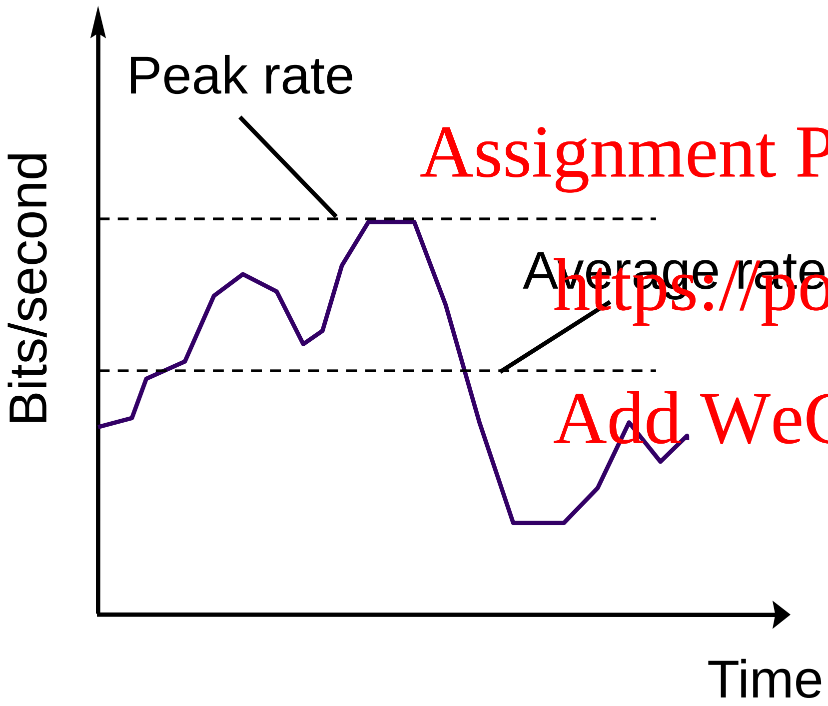
- Network performance is guaranteed to all traffic flows that have been admitted into the network

- Initially for connection-oriented networks

- Key Mechanisms

  - Admission Control
  - Policing
  - Traffic Shaping
  - Traffic Scheduling

Admission Control



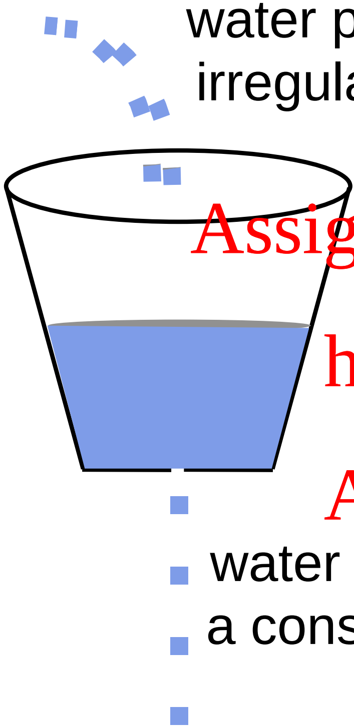
Typical bit rate demanded by a variable bit rate information source

- 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

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

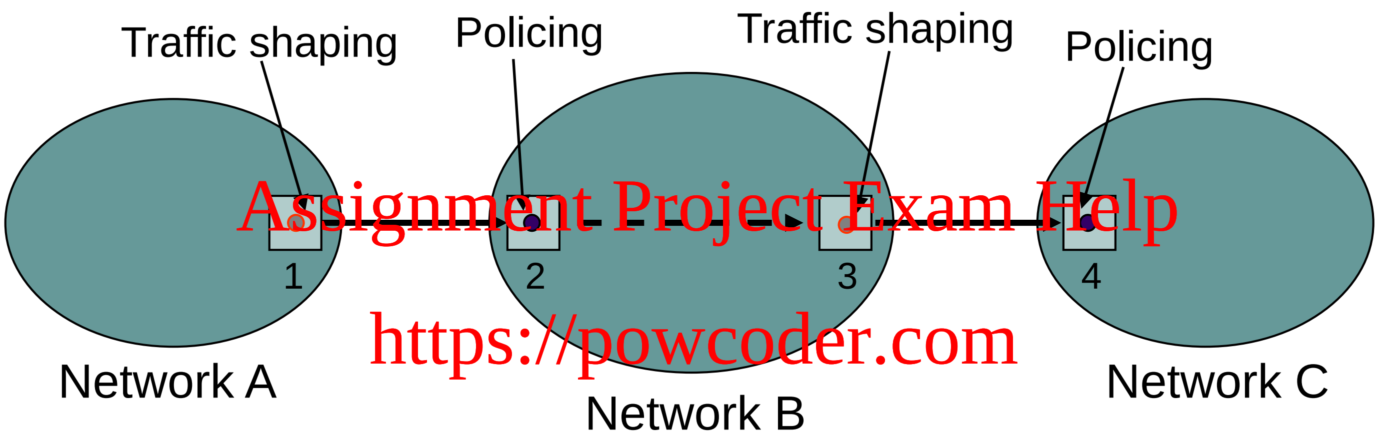
Leaky Bucket algorithm can be used to police arrival rate of a packet stream



Leak rate corresponds to long-term rate

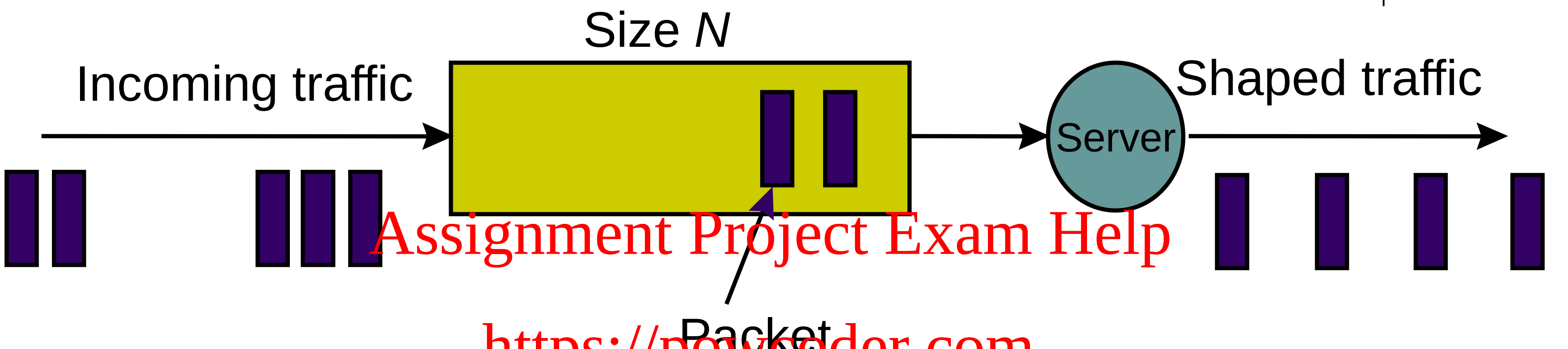
Bucket depth corresponds to maximum allowable burst arrival

Traffic Shaping



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

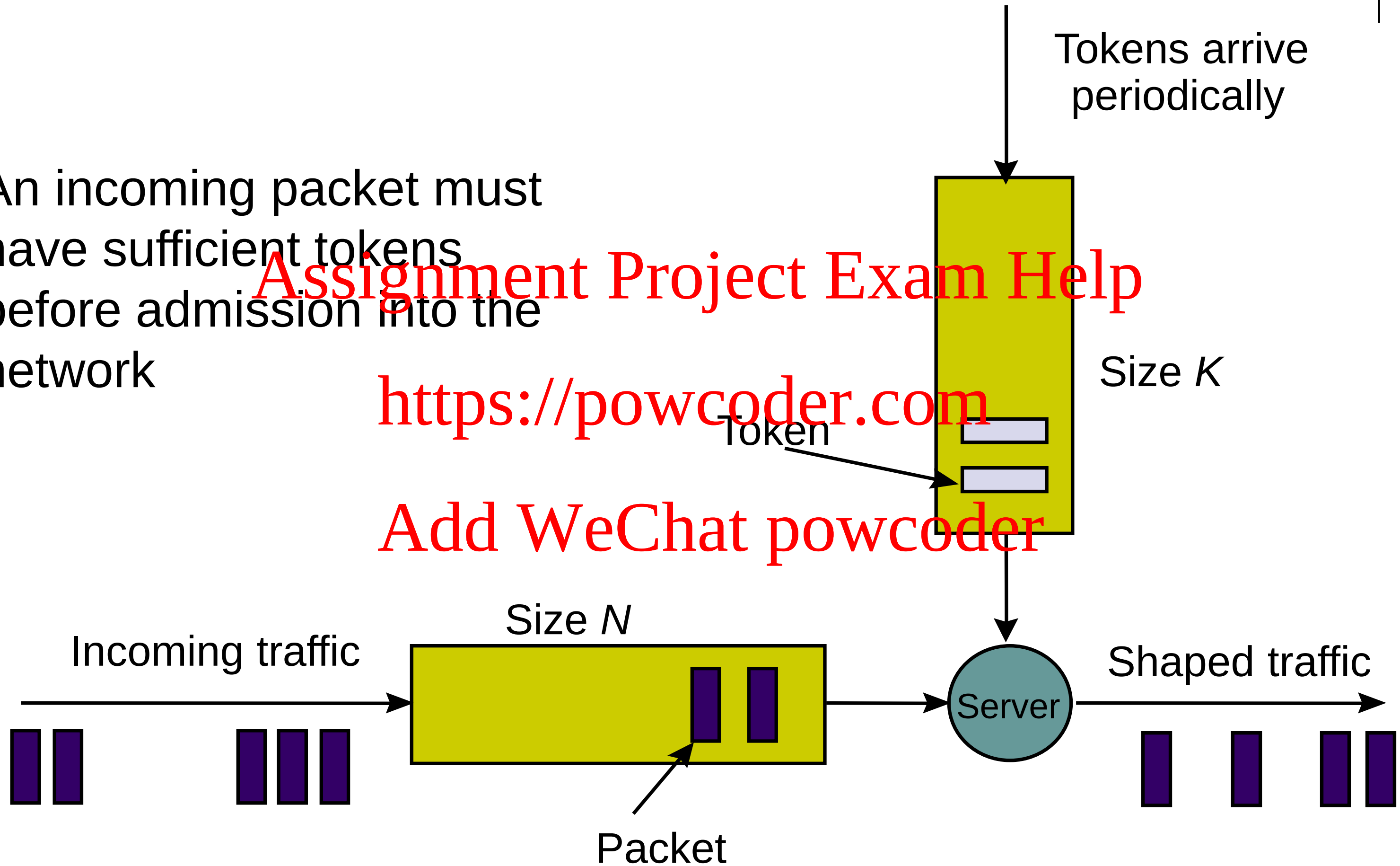
Leaky Bucket Traffic Shaper



- Buffer incoming packets
- Play out periodically to conform to parameters
- Surges in arrivals are buffered & smoothed out
- Possible packet loss due to buffer overflow
- Too restrictive, since conforming traffic for policing device does not need to be completely smooth
  - Policing device allows for burstiness if under certain limit

Token Bucket Traffic Shaper

An incoming packet must have sufficient tokens before admission into the network

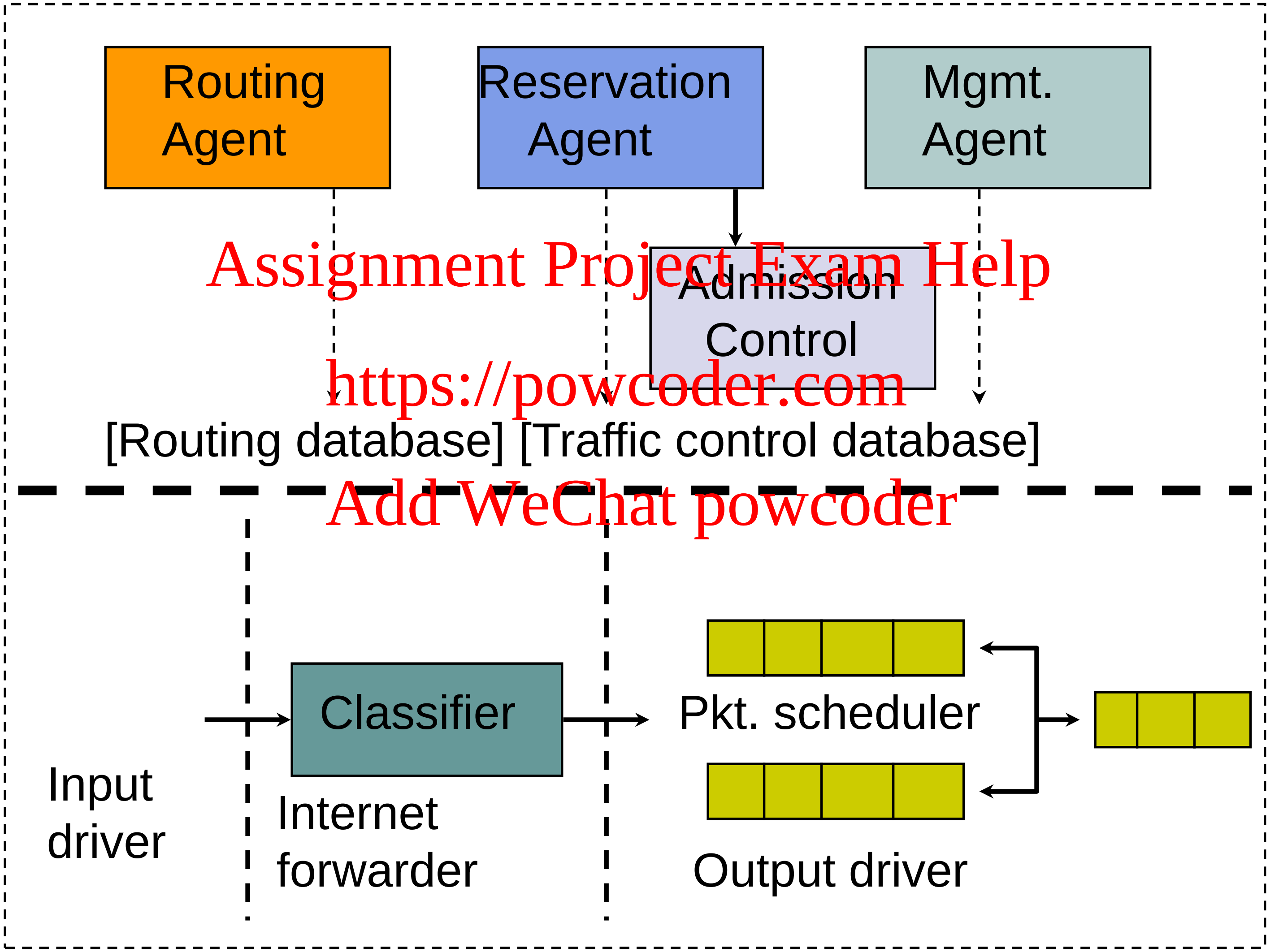


- Token rate regulates transfer of packets
- If sufficient tokens available, packets enter network without delay
- K determines how much burstiness allowed into the network

Scheduling for Guaranteed Service

- Suppose guaranteed bounds on end-to-end delay across the network are to be provided
- A call admission control procedure is required to calculate and allocate resources & set schedulers
- Call setup procedure must identify a route in which links can provide necessary guaranteed bandwidth so that the bound is met.
  - Involves obtaining information from potential hops about their available bandwidth, selecting a path, and allocating appropriate bandwidth in the path
- Traffic flows from sources must be shaped/regulated so that they do not exceed their allocated resources
- Strict delay bounds can be met

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 – timeout on missing acknowledgements
  - Routers/switches generate messages alerting to congestion – as a separate packet or piggybacked on data

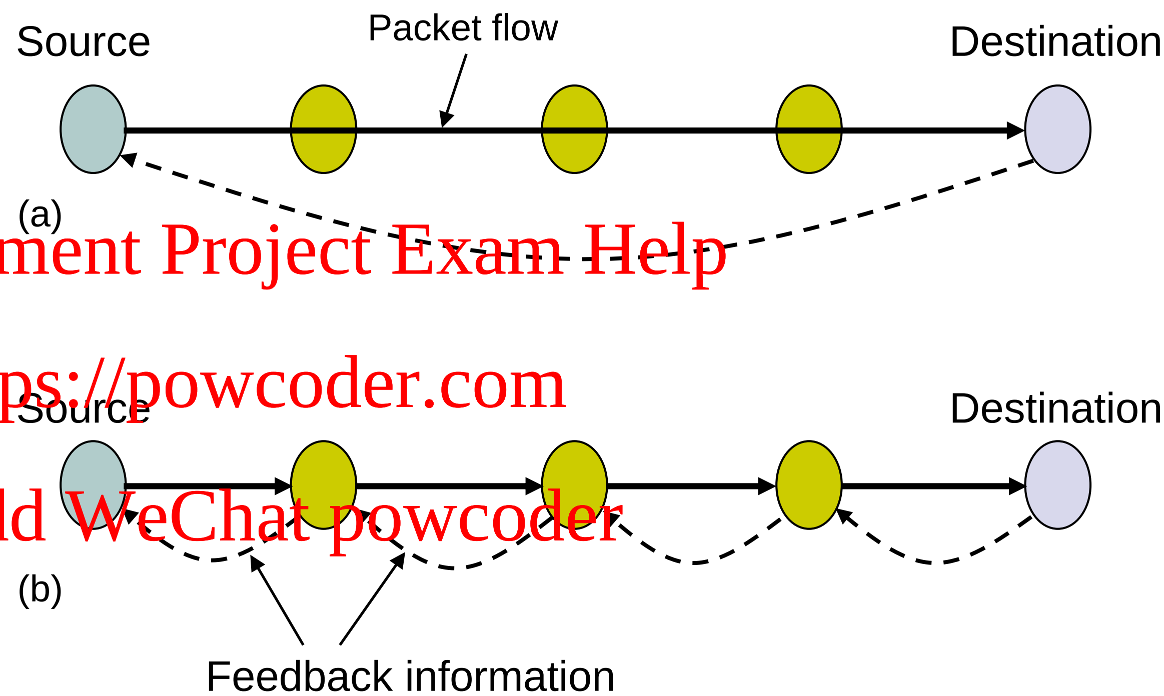
End-to-End vs. Hop-by-Hop Congestion Control

a) End-to-End

- because of delay information may not be accurate when source receives it

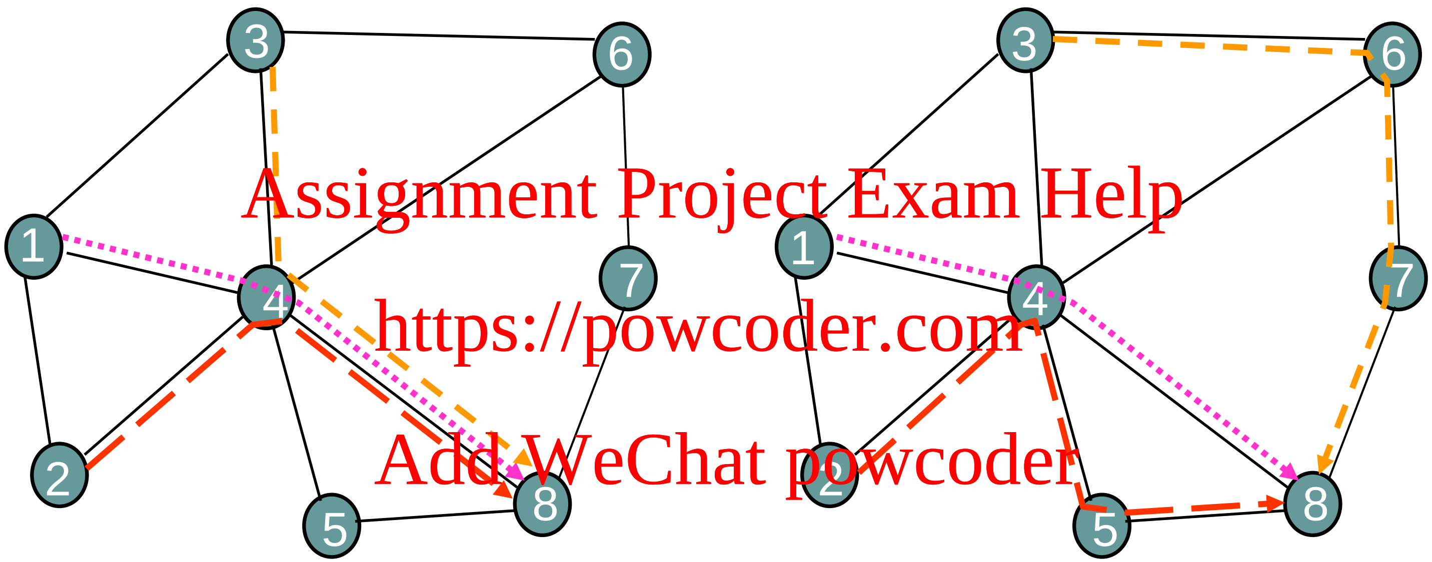
b) Hop-by-Hop

- reacts much faster



Traffic Engineering

- Management exerted at flow aggregate level – multiplicity of flows
- 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
  - Constrained Shortest Path Routing – pruning links having available bandwidth less than required bandwidth
- Must take into account aggregate demand from all flows



(a) (b)

Shortest path routing congests link 4 to 8

Better flow allocation distributes flows more uniformly