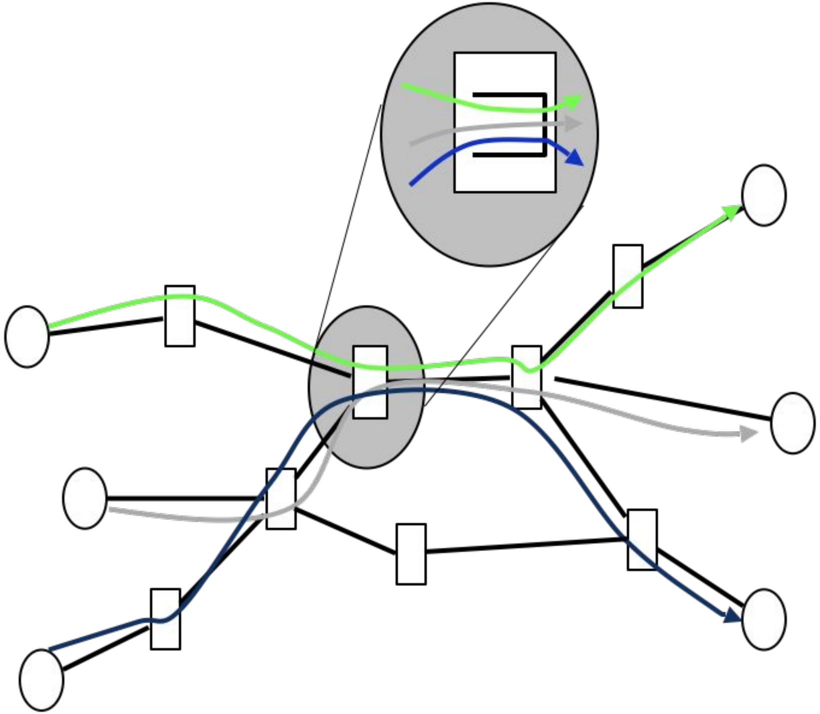
Queue Management / QoS

Traffic and resource management

- Resources are statistically shared in packet switched networks
- Overload causes congestion
  - packets delayed or dropped
  - application performance suffers
- Transient vs. persistent
- Challenge
  - high resource utilization
  - high application performance

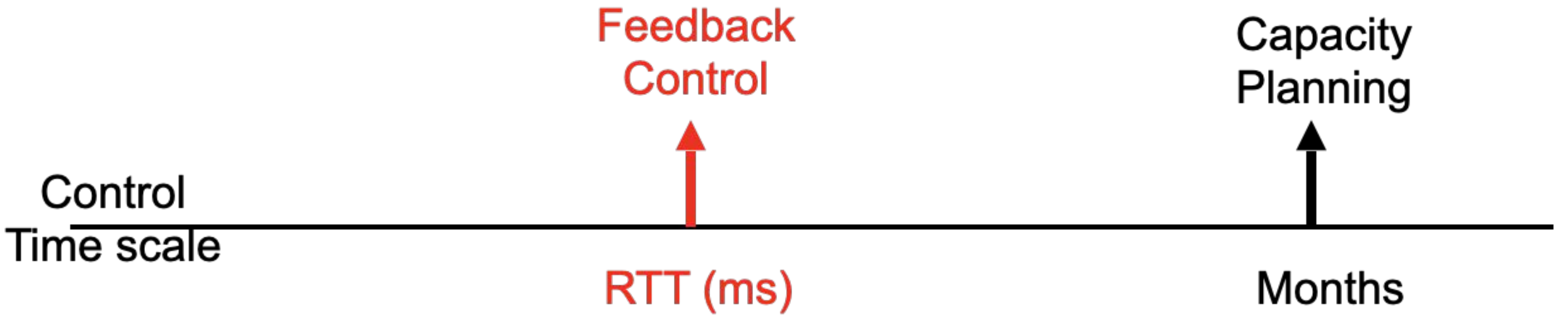


Resource management approaches

- Increase resources
  - install new links, faster routers
  - capacity planning, provisioning, traffic engineering
  - happen at longer timescale
- Reduce or delay demand
  - Reactive approach: encourage everyone to reduce or delay demand
  - Reservation approach: some requests will be rejected by the network

Recall TCP congestion control

- End-system-only solution (TCP)
  - dynamically estimates network state
  - packet loss signals congestion
  - reduces transmission rate in presence of congestion
  - routers play little role



Ideas to improve?

- Enhance routers to control traffic
  - Rate limiting
  - Fair Queueing
- Provide QoS by limiting congestion
- Enhance routers to help TCP
  - Random Early Discard
  - ECN

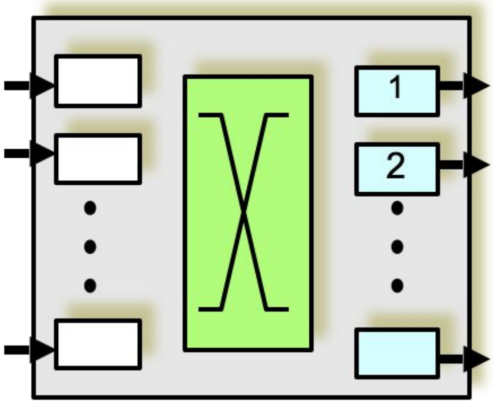
What can routers control to help?

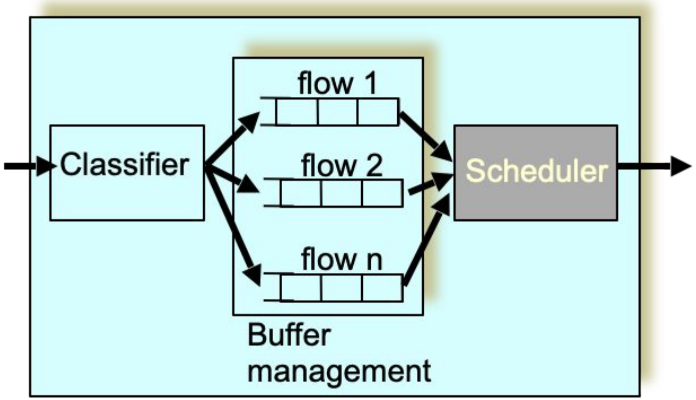
What resource can routers control to help?

Buffers

Router mechanisms

- Buffer management: when and which packet(s) to drop?
- Scheduling: which packet to transmit next?





Queuing disciplines

- Each router **must** implement some queuing discipline
- Queuing allocates both bandwidth and buffer space:
  - Bandwidth: which packet to serve (transmit) next
  - Buffer space: which packet to drop next (when required)
- Queuing also affects latency

Queuing disciplines

- **FIFO + drop-tail**
  - Simplest choice
  - Used widely in the Internet
- FIFO (first-in-first-out)
  - Implies single class of traffic
- Drop-tail
  - Arriving packets get dropped when queue is full regardless of flow or importance
- Important distinction:
  - FIFO: scheduling discipline
  - Drop-tail: drop policy

FIFO

- What if scheduler uses one first-in first-out queue?
  - Simple to implement
  - But, restrictive in providing guarantees
- Example: two kinds of traffic
  - Video conferencing needs high bandwidth and low delay
  - E.g., 1 Mbps and 100 msec delay
  - E-mail transfers not very sensitive to delay
- Cannot admit much e-mail traffic
  - Since it will interfere with the video conference traffic

FIFO + drop-tail problems

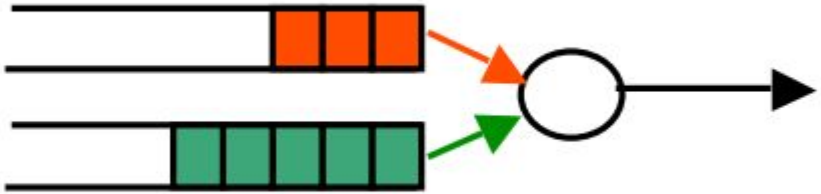
- Leaves responsibility of congestion control completely to the edges (e.g., TCP)
- Does not separate between different flows
- No policing: send more packets -> get more service
- Synchronization: end hosts react to same events

FIFO + drop-tail problems

- Full queues
  - Routers are forced to have have large queues to maintain high utilizations
    - Leads to bufferbloat
  - TCP detects congestion from loss
    - Forces network to have long standing queues in steady-state
- Lock-out problem
  - Drop-tail routers treat bursty traffic poorly
  - Traffic gets synchronized easily -> allows a few flows to monopolize the queue space

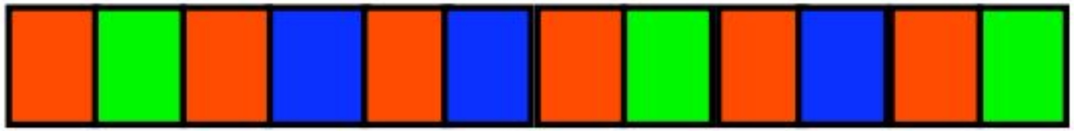
Priority queues

- Strict priority
  - Multiple levels of priority
  - Always transmit high-priority traffic, when present
  - .. and force the lower priority traffic to wait
- Isolation for the high-priority traffic
  - Almost like it has a dedicated link
  - Except for the (small) delay for packet transmission
    - High-priority packet arrives during transmission of low-priority
      - Router completes sending the low-priority traffic first
- Starvation possible



Weighted fair queues

- Limitations of strict priority
  - Lower priority queues may starve for long periods
  - … even if the high-priority traffic can afford to wait
  - Traffic still competes inside each priority queue
- Weighted fair scheduling
  - Assign each queue a fraction of the link bandwidth
  - Rotate across the queues on a small time scale
  - Send extra traffic from one queue if others are idle



50% red, 25% blue, 25% green

Queuing tradeoffs

- FIFO
  - One queue, trivial scheduler
- Priority queues
  - One queue per priority level, simple scheduler
- Weighted fair queues
  - One queue per class, and more complex scheduler

Active queue management

- Design active router queue management to aid congestion control
- Why?
  - Router has unified view of queuing behavior
  - Routers see actual queue occupancy (can distinguish between queue delay and propagation delay)
  - Routers can decide on transient congestion, based on workload

AQM design goals

- Keep throughput high and delay low
- Accommodate bursts
- Queue size should reflect ability to accept bursts rather than steady-state queuing
    - Must be careful to avoid bufferbloat
- Improve TCP performance with minimal hardware changes

Lock-out problem

- How can we solve this?

Lock-out problem

- How can we solve this?
  - Random drop
    - Packet arriving when queue is full causes some random packet to be dropped
  - Drop front
    - On full queue, drop packet at head of queue
  - Random drop and drop front solve the lock-out problem but not the full-queues problem

Full queues problem

- We need to avoid queues being full, how?

Full queues problem

- We need to avoid queues being full, how?
- Drop packets before queue becomes full (early drop)
- Intuition: notify senders of incipient congestion
  - Example: early random drop (ERD):
    - If qlen > drop level, drop each new packet with fixed probability p
    - Does not control misbehaving users

Random Early Detection (RED)

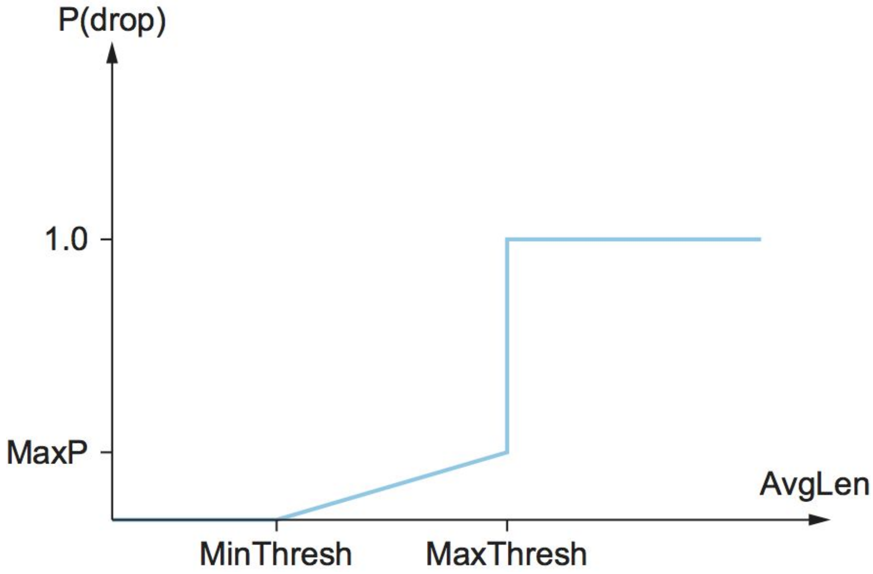
- Detect incipient congestion
- Assume hosts respond to lost packets
- Avoid window synchronization
  - Randomly mark packets
- Avoid bias against bursty traffic

Random Early Detection (RED) algorithm

- Maintain running average of queue length

```
if AvgLen <= MinThreshold
    queue the packet
if MinThreshold < AvgLen < MaxThreshold
    calculate probability P
    drop the arriving packet with probability P
if MaxThreshold <= AvgLen
    drop the arriving packet
```

Random Early Detection (RED) algorithm



Explicit Congestion Notification (ECN)

- Traditional mechanism
  - packet drop as implicit congestion signal to end systems
  - TCP will slow down
- Works well for bulk data transfer
- Does not work well for delay sensitive applications
  - audio, WEB, telnet
- Explicit Congestion Notification (ECN)
  - use two bits in IP header
    - ECN-Capable Transport (ECT) bit set by sender
    - Congestion Experienced (CE) bit set by router