Link Scheduling & Queuing

COS 461: Computer Networks

http://www.cs.princeton.edu/courses/archive/spr14/cos461/
Outline

• Link scheduling (not covered on Monday)
• Queuing practice questions
• Miscellanea
First-In First-Out Scheduling

• First-in first-out scheduling
  – Simple, but restrictive

• Example: two kinds of traffic
  – Voice over IP needs low delay
  – E-mail is not that sensitive about delay

• Voice traffic waits behind e-mail
Strict Priority

• Multiple levels of priority
  – Always transmit high-priority traffic when present

• Isolation for the high-priority traffic
  – Almost like it has a dedicated link
    • Except for (small) delay for transmission

• Possible starvation
Weighted Fair Queuing

• Each queue gets a fraction of the link bandwidth
• Rotate across queues on a small time scale
• What if a queue is empty during its time slice?
  – Move on to next queue if work conserving

50% red, 25% blue, 25% green
Weighted Fair Queuing

• If non-work conserving
  – Flows get at most their allocated weight

• If work-conserving
  – Send extra traffic from one queue if others are idle
  – Bytes, not packets
  – Higher or lower utilization than non-work conserving?

• Results in max-min fairness
  – Maximize the minimum rate of each flow
Implementation Trade-Offs

• FIFO
  – One queue, trivial scheduler

• Strict priority
  – One queue per priority level, simple scheduler

• Weighted fair scheduling
  – One queue per class, and more complex scheduler
For the following questions, assume that these are always coupled with a FIFO scheduling policy.

The **full queue** problem occurs if routers’ queues are often full.

The **lockout** problem refers to a situation where a small number of flows monopolize the available queue space on a router.

• Drop-tail solves the full queue problem  T/F
For the following questions, assume that these are always coupled with a FIFO scheduling policy.

The **full queue** problem occurs if routers’ queues are often full.

The **lockout** problem refers to a situation where a small number of flows monopolize the available queue space on a router.

• Drop-tail solves the full queue problem  T/F
For the following questions, assume that these are always coupled with a FIFO scheduling policy.

The **full queue** problem occurs if routers’ queues are often full.

The **lockout** problem refers to a situation where a small number of flows monopolize the available queue space on a router.

• Drop-tail solves the full queue problem  T/F
• Random Early Detection (RED) solves the full queue problem  T/F
For the following questions, assume that these are always coupled with a FIFO scheduling policy.

The **full queue** problem occurs if routers’ queues are often full.

The **lockout** problem refers to a situation where a small number of flows monopolize the available queue space on a router.

- Drop-tail solves the full queue problem  \( T/F \)
- Random Early Detection (RED) solves the full queue problem  \( T/F \)
For the following questions, assume that these are always coupled with a FIFO scheduling policy.

The **full queue** problem occurs if routers’ queues are often full.

The **lockout** problem refers to a situation where a small number of flows monopolize the available queue space on a router.

- Drop-tail solves the full queue problem  T/F
- Random Early Detection (RED) solves the full queue problem  T/F
- RED solves the lockout problem for TCP flows  T/F
For the following questions, assume that these are always coupled with a FIFO scheduling policy.

The **full queue** problem occurs if routers’ queues are often full.

The **lockout** problem refers to a situation where a small number of flows monopolize the available queue space on a router.

- Drop-tail solves the full queue problem  \( \text{T/F} \)
- Random Early Detection (RED) solves the full queue problem  \( \text{T/F} \)
- RED solves the lockout problem for TCP flows  \( \text{T/F} \)
1. Drop-tail has fewer burst losses than RED
   T/F
1 Drop-tail has fewer burst losses than RED
T/F
1. Drop-tail has fewer burst losses than RED  
   T/F 

2. Both drop-tail and RED can be used with Explicit Congestion Notification (ECN), so the router can signal congestion to the sender without dropping a packet  
   T/F
1. Drop-tail has fewer burst losses than RED
   T/F

2. Both drop-tail and RED can be used with Explicit Congestion Notification (ECN), so the router can signal congestion to the sender without dropping a packet
   T/F
1. Drop-tail has fewer burst losses than RED
   T/F

2. Both drop-tail and RED can be used with Explicit Congestion Notification (ECN), so the router can signal congestion to the sender without dropping a packet
   T/F

3. RED drops every incoming packet with some probability $P > 0$
   T/F
1. Drop-tail has fewer burst losses than RED
   \[ T/F \]

2. Both drop-tail and RED can be used with Explicit Congestion Notification (ECN), so the router can signal congestion to the sender without dropping a packet
   \[ T/F \]

3. RED drops every incoming packet with some probability \( P > 0 \)
   \[ T/F \]
TCP Windows

• Receive *window*
  – flow control

• Congestion *window*
  – Additive increase / multiplicative decrease
  – Triple duplicate ACKs
  – Slow-start, fast retransmit, fast recovery, etc.