Chapter 6
Queuing Disciplines

Networking
CS 3470, Section 1
Flow control vs Congestion control

- **Flow control** involves preventing senders from overrunning the capacity of the receivers.
- **Congestion control** involves preventing too much data from being injected into the network, thereby causing switches or links to become overloaded.
Congestion Control

- Queuing disciplines in the routers
  - Discussed today
- TCP congestion control
- Congestion avoidance mechanisms
Congestion Control and Resource Allocation

- Resources
  - Bandwidth of the links
  - Buffers at the routers and switches

- Packets contend at a router for the use of a link, with each contending packet placed in a queue waiting for its turn to be transmitted over the link
Congestion Control and Resource Allocation

- When too many packets are contending for the same link
  - The queue overflows
  - Packets get dropped
    - Network is congested!

- Network should provide a congestion control mechanism to deal with such a situation
Congestion Control and Resource Allocation

- Who’s fault is congestion?

- Who can fix it?
Congestion Control and Resource Allocation

- Who can fix it?
- In network elements
  - Various queuing disciplines can be used to control the order in which packets get transmitted and which packets get dropped
- At the hosts’ end
  - The congestion control mechanism paces how fast sources are allowed to send packets
Network Control Issues

- Why the congestion problem in the first place?
  - Resources are limited: buffer space, bandwidth allocation

- One could simply “route around” congested links. Put a large edge weight on a congested link to route around it. That doesn't solve the inherent problem though.
Routing in the Internet is a complex issue.

- What can possibly happen?
- [http://www.theregister.co.uk/2010/04/09/china_bgp_interweb_snafu/](http://www.theregister.co.uk/2010/04/09/china_bgp_interweb_snafu/)
Addressing Congestion at the Routers: Queuing Disciplines

- Routers must implement some queuing discipline that governs how packets are buffered or prioritized.

- One can think of queuing disciplines as rules for the allocating bandwidth or rules for the allocation of buffer space within the router.

- Two common disciplines:
  - FIFO
  - Fair Queuing
FIFO Queuing

- **FIFO queuing** is called first-come-first-served (FCFS) queuing
- First packet that arrives at a router is first packet to be transmitted
- Amount of buffer space at each router is finite
  - **Tail drop** - If a packet arrives and the queue (buffer space) is full, then the router discards that packet
FIFO Queuing

(a) FIFO queuing; (b) tail drop at a FIFO queue.
FIFO Queuing – Priority Queuing

- The routers then implement multiple FIFO queues, one for each priority class.
- Router always transmits packets out of the *highest-priority queue* if that queue is nonempty before moving on to the next priority queue.
- Within each priority, packets are still managed in a FIFO manner.
Fair Queuing

*Fair queuing (FQ)* maintains a separate queue for each flow currently being handled by the router.

- The router then services these queues in round-robin algorithm.
Fair Queuing

Round-robin service of four flows at a router
The main complication with Fair Queuing is that the packets being processed at a router are not necessarily the same length.

To truly allocate the bandwidth of the outgoing link in a fair manner, it is necessary to take packet length into consideration.
Fair Queuing

- Example: Packet length not taken into consideration
  - Two flows:
    - One with 1000-byte packets, another with 500-byte packets
    - Simple round-robin servicing of packets from each flow’s queue will give the first flow two thirds of the link’s bandwidth and the second flow only one-third of its bandwidth.
What we really want is **bit-by-bit** round-robin; that is, the router transmits a bit from flow 1, then a bit from flow 2, and so on.

- However, it is not feasible to interleave the bits from different packets.

Simulates this behavior instead

- Determine when a given packet would finish being transmitted if it were being sent using bit-by-bit round-robin
- Use this finishing time to sequence the packets for transmission.
Queuing Disciplines

- Fair Queuing
  - To understand the algorithm for approximating bit-by-bit round robin, consider the behavior of a single flow
  - For this flow, let
    - \( P_i \): denote the length of packet \( i \)
    - \( S_i \): time when the router starts to transmit packet \( i \)
    - \( F_i \): time when router finishes transmitting packet \( i \)
    - \( F_i = S_i + P_i \)
Queuing Disciplines

- Fair Queuing
  - When do we start transmitting packet $i$?
    - Depends on whether packet $i$ arrived before or after the router finishes transmitting packet $i-1$ for the flow
  - Let $A_i$ denote the time that packet $i$ arrives at the router
  - Then $S_i = \max(F_{i-1}, A_i)$
  - $F_i = \max(F_{i-1}, A_i) + P_i$
Queuing Disciplines

- **Fair Queuing**
  - Now for every flow, we calculate $F_i$ for each packet that arrives using our formula
  - We then treat all the $F_i$ as timestamps
  - Next packet to transmit is always the packet that has the lowest timestamp
    - The packet that should finish transmission before all others
**Queuing Disciplines**

- Fair Queuing

Example of fair queuing in action:
(a) packets with earlier finishing times are sent first;
(b) sending of a packet already in progress is completed