Beyond Best Effort

- Long queues delay all packets
- Internet is used by different users / applications
  - Different performance requirements
  - Different classes of service
- Service provided by choosing which packet to send
  - Scheduler determines which packet is next
- Scheduling considerations
  - Level of service / priority
  - Fairness
  - System utilization
- Ultimate goal: traffic isolation
- Requires classification
  - Identification of different classes of traffic
FIFO Queuing

- Simplest case
  - Single queue
  - FIFO queuing

Priority Queuing

- Different priority classes
  - Requires separate queues
Fair Sharing

- Within priority class

Round-Robin Scheduling

- Each queue gets equal opportunity to transmit
  - If queue has packet, send packet
  - Move to next queue
  - Etc.
- Benefits?
  - Traffic isolation
- What is the problem with this approach?
  - Packet size can cause unfairness
Fairness

- What is a fair allocation of resources?
- Max-min fairness
  1. No user receives more than its request.
  2. No other allocation that scheme satisfying condition 1 has a higher minimum allocation.
  3. Condition 2 remains recursively true as we remove the minimal user and reduce the total resource accordingly.

- Alternative formulation
  - A feasible allocation of rates is “max-min fair” if and only if an increase of any rate within the domain of feasible allocations must be at the cost of a decrease of some already smaller rate.

Weights

- Different traffic class can have different weights
  - Different levels of service
  - Different bandwidth needs
  - Aggregate flows
  - Etc.
- Weights $w_i$ indicate proportion of link bandwidth
  - Each flow receives $w_i/(\sum w_j)$ of the link capacity
- Most scheduling algorithms can be extended to consider weights
Bit-wise round robin

- Packet sizes cause unfairness in round robin
- Bit-wise round robin
  - Idealized scheduling
  - Smallest entity is bit
    » Unrealistic for real networks
- For link of capacity C, each of N flows receives C/N
  - Variation between flows at most one bit

Deficit Round Robin

- Round robin scheduler with O(1) complexity
- Each queue has a “deficit counter”
  • “Credit” for how much can be sent
- Steps:
  • Deficit counter incremented by “quantum size”
  • While next packet size in queue is less than deficit
    » Send packet
    » Decrement deficit by packet size
  • Move to next queue
- Packets need to wait until credit has accumulated
  • Fairness
  • No delay guarantees
**Fair Queuing**

- Need to discretize bit-wise round robin
  - Whole packets
- **Solution:**
  - Emulate bitwise round-robin
  - Determine order of completed packet transmissions
  - Send packets in same order
- **Notation**
  - $\alpha$ – flow id
  - $S_i^\alpha$ – start time of packet $i$
  - $F_i^\alpha$ – finish time of packet $i$
  - $P_i^\alpha$ – size of packet $i$
  - $t_i$ – arrival time of

**Start time:** $S_i^\alpha = \max(F_{i-1}^\alpha, t_i^\alpha)$

**Transmission time** $T_i^\alpha$: P rounds
- One round takes $N$ bit times, $T_i^\alpha = P_i^\alpha \cdot N$

**Finish time:** $F_i^\alpha = S_i^\alpha + T_i^\alpha$

- Packets are sent in order of finish time
- **What is the complexity of fair queuing?**
  - $O(\log N)$ for each packet
  - Expensive for large number of flows / classes
Queuing Delay

- Delay for different flows
  - Three FTP sources
  - One Telnet source (variable)
- Queuing delay depends on load of source
  - Isolated from other traffic

![Graph showing queuing delay vs throughput](image)

Homework

- Read
- SPARK
  - Assessment quiz