Queueing

CS640, 2015-03-24

FIFO Queueing

- Ensures that of all the packets that are sent out the one arriving early is sent out first.
- Tail-Drop - the packet arriving last is dropped.
    - FIFO does not imply tail drop - Drop policy is independent of queuing policy. For e.g.:, we can drop the packet which takes the largest space in the buffer.
- FIFO is not good enough - why ?
    - Rouge sources can flood the router buffer; thus depriving other sources of their bandwidth.

Other Queueing Models

- Priority queues - can be used to distinguish classes of traffic
    - Higher class gets better treatment
    - Can starve low priority traffic
    - The rouge source problem still exists
- We need a round robin mechanism:
    - Naive round robin : Cycle over packets in a flow; if there are 2 flows send packets alternately from each flow. Great, works if packet sizes are the same.
    - But packet sizes differ; because of type of service, fragmentation in upstream routers etc.
    - So what do we do ? We need to send bit by bit; however, routers do not forward bit wise, they can process only packets.
    - Thus we need to schedule packet processing order such that it simulates bit-by-bit round robin. Note that it cannot exactly be bit-by-bit, but we want it to be as close as possible.

Fair Queueing

- Calculating finishing times: $F_i = \max(A_i, F_{i-1}) + P_i$
- For high values of P_i, the value of F_i tends to increase and thus will be scheduled later.
- Some properties of the algorithm:
    - Work conserving - If a packet is there to be sent it is sent, there is no reservation
    - Share automatically increases / decreases as number of flows decreases /increases
- Weighted fair queuing
- Interaction with Congestion Control