ECE 671 – Lecture 18

Quality of Service
Link Scheduling

Quality of Service (QoS)

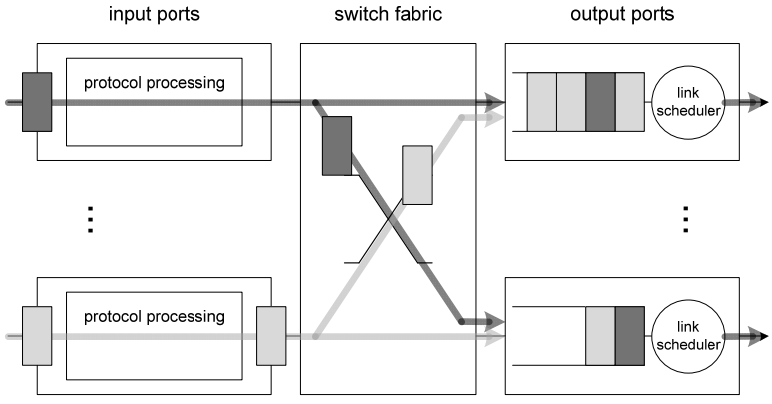
- Queuing theory provides basis network traffic
  - Simple model that treats all traffic equally
- Applications may require certain quality of service
  - Voice, video, control, gaming, etc.
- Quality metrics depend on application
  - Bandwidth, delay, jitter, loss
- How can the network provide quality of service?

QoS techniques in network

- Routing
  - Choose path with matching characteristics
- Queuing
  - Decide on outgoing order of packets
- QoS in network architecture
  - Overprovisioning
  - Integrated Services (IntServ): QoS routers, reservations
  - Differentiated Services (DiffServ): traffic classes, enforcement at edge

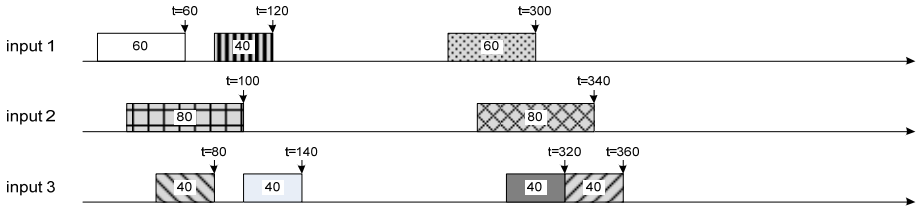
Link scheduling

- Order of outgoing packets determines quality
- Example: first-in-first-out queuing

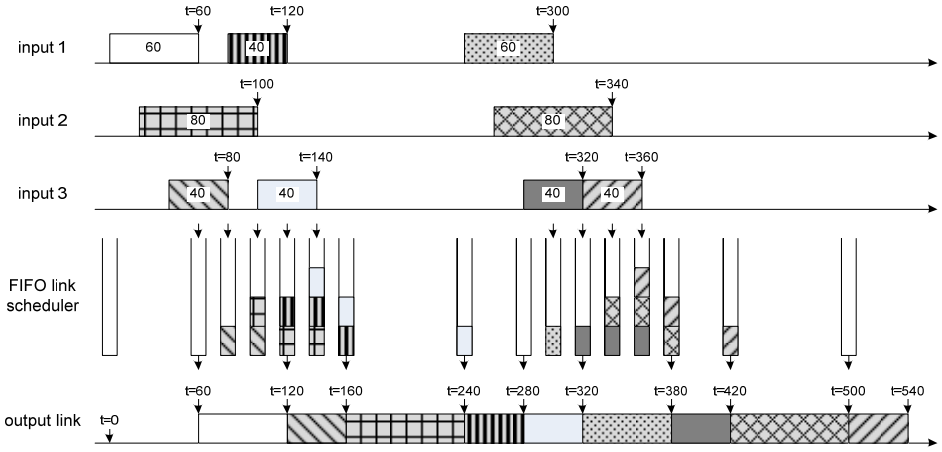


FIFO queuing example

- Traffic from 3 inputs to 1 FIFO output
  - What is the outgoing order and timing?

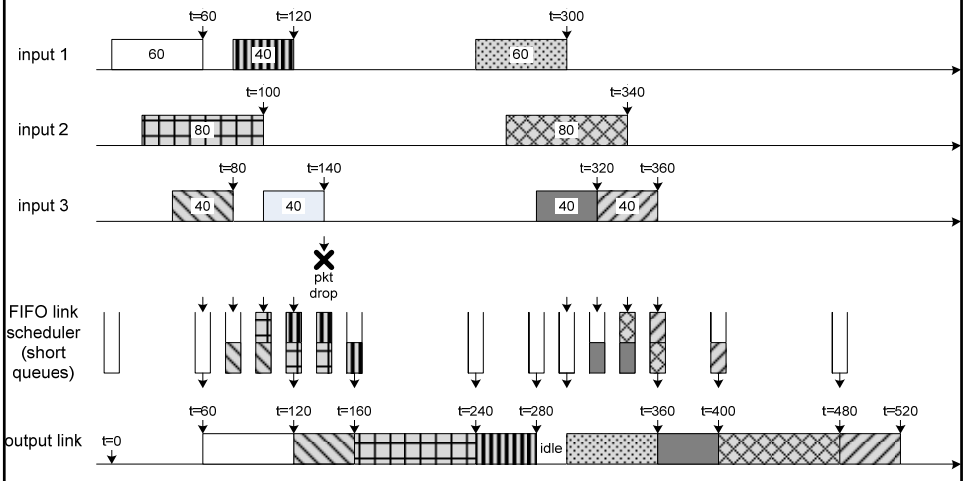


FIFO queuing example



- What if there is a short queue (e.g., 2 packets)?

FIFO queuing example (short queues)



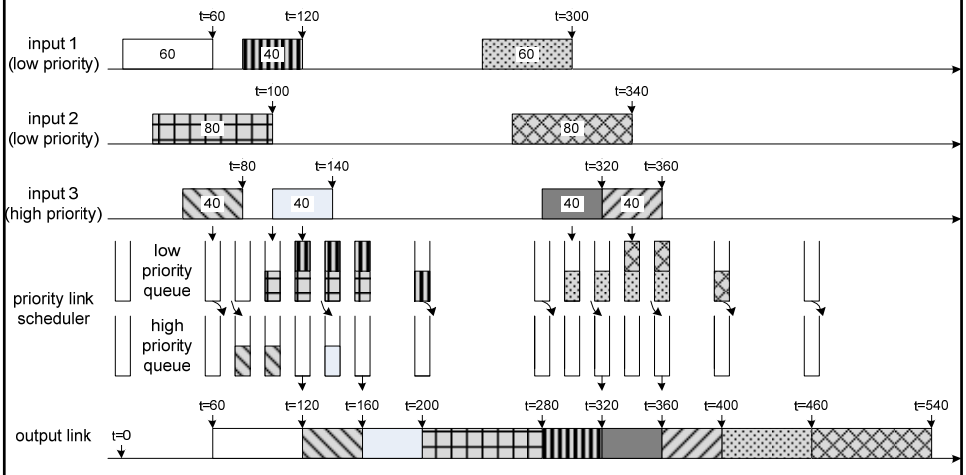
Link scheduling for QoS

- How can link scheduling affect quality of service?
- What are possible scheduling approaches?

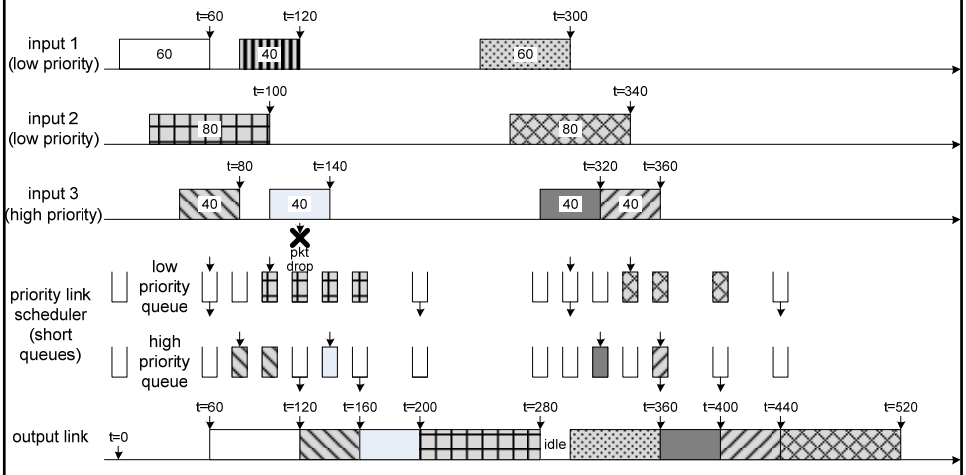
Priority queuing

- Traffic classified by priority classes
  - Two or more classes
- Strict ordering of link access priority by class
  - High priority: always sent when link is available
  - Low priority: only sent when no high priority traffic
- What are pros and cons?

Priority queuing example



Priority queuing example (short queues)

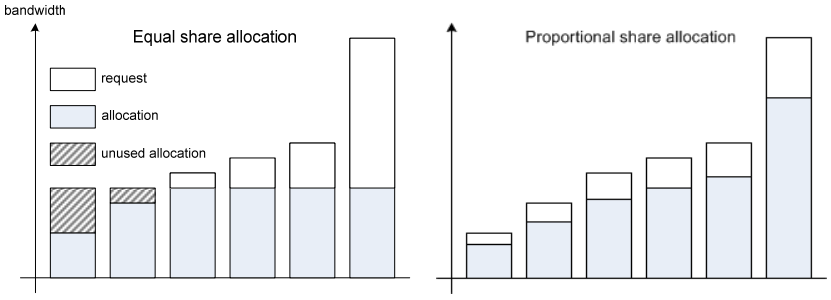


Fairness

- Priority queuing can starve low priority class
- What would be a fair way of sharing link bandwidth?

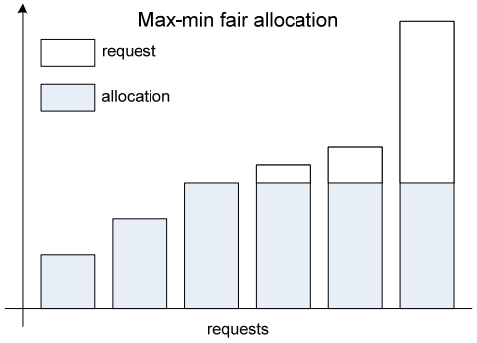
Resource allocation schemes

- What are the drawbacks of equal and proportional allocation?



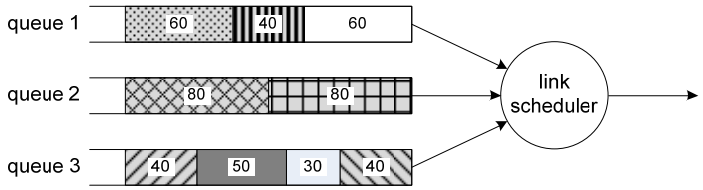
Max-min fair allocation

- Iterative allocation from smallest to largest
  - Allocate request or fair share (whatever is less)
  - Update remaining bandwidth and fair share



Fair link scheduling

- Scheduler picks from different queues
  - Fair sharing between queues



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

Round-robin

- 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

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
- Fluid version is called “generalized processor sharing” (GPS)

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 with packet
- 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
  - α – flow id
  - S_i^{α} – start time of packet i
  - F_i^{α} – finish time of packet i
  - P_i^{α} – size of packet i
  - t_i – arrival time of

Fair Queuing

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

Fair queuing example