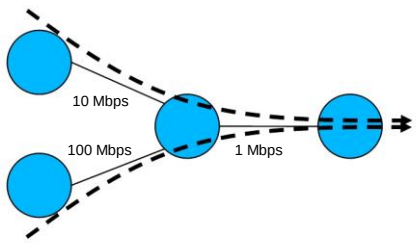
Network-Based Congestion Control

Motivation

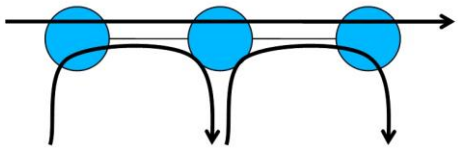
- Hosts can't always be "trusted" to implement proper congestion control
- Network needs to place guarantees on traffic flows to ensure required service level
  - Support real-time traffic, multimedia
- Before
  - Network treats each packet equally
- After
  - Network treats packets as a part of a network flow and ensures fair allocation between flows

Flow View



Goal: Fairness

- Goal is to provide a fair allocation between flows
- What's fair?

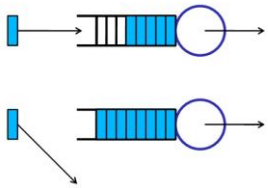


Queuing Disciplines

- Routers are basically sets of queues
- The order packets are serviced from queues determines which flows get how much of the networks resources

FIFO

- First-In First-Out
- If queue is full, packets are dropped



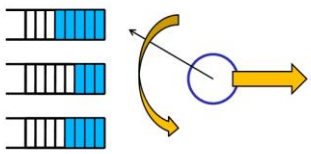
Priority FIFO

- Packets are marked with “service class”
- Uses the TOS (type of service) field in the IP header

- Problem: Starvation
  - Low priority packets may NEVER get through if the queue is constantly full with higher priority packets
  - Only works if very few packets have a higher priority

Fair Queuing

- Each flow receives its own virtual queue
- Virtual queues are serviced in round-robin fashion



- Problem: per-flow state
  - Router must maintain queue for every active flow
  - Not scalable

- *Weighted Fair Queuing"* scheduler (WFQ).
- To put a packet into a queue:
- Class one code Best-effort" services (queues)
- Class second code to select the "*premium queue"* (higher weighted than Best-effort"

- The premium class can be kept low, since WFQ will try to transmit premium packets as soon as possible.
- If premium load ~10% . behaves as if premium traffic is running on an under-loaded network.
- If premium load ~30% . behaves like a highly loaded network.
- Just as in WRED, we can generalize this WFQ-based approach to allow more than two classes represented by different code points.
- We can also combine the idea of a queue selector with a drop preference