Analysis of Packet Scheduling using Network Simulation

Ethan Ruchotzke

CprE 558
Iowa State University

December 11, 2024

Problem Statement

- Modern internetworking is built for "best effort" scheduling
- Real time networks demand packet scheduling with different properties
  - Fairness (typical, max-min)
- Demand to investigate packet schedulers and identify their impact on flows

Solution

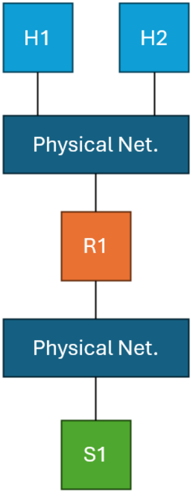
- Simulation!
- Work through theoretical flows of packets
    - Fast, easy to implement many test cases
    - Easy to implement routing schemes
- Easy to pull results from simulated network without noise/interference

Network Simulation Choices

- Many options
  - NS-3 (hard to tweak for experiments on scheduling)
  - Mininet (focused on SDNs/higher-level changes)
  - GNS-3 (too intensive/large)
- I decided to roll my own, since I had the experience to do so
  - SimPy - Discrete Event Simulator

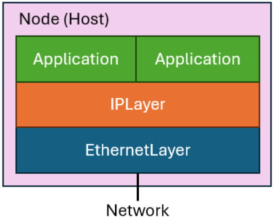
Network Simulation

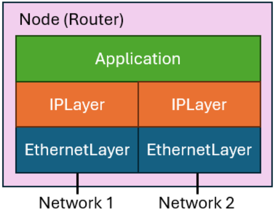
- Two major components
  - Nodes (contain a network stack)
  - Networks (equivalent to CSMA network)
- Networks have a bandwidth which limits transmission time
- Nodes must wait for the shared resource (network) before transmitting
- Nodes run applications and various network layers



Network Stack

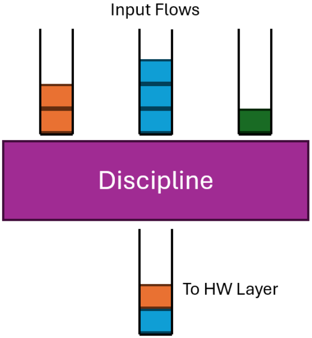
- Replicate the layers of an actual network
  - Ethernet: Physical network access, filtering
  - IP: Routing
  - App: Generation/consumption
- Multiple layers of a type allowed (routers have two interfaces)
- IP Layer handles packet queueing (of interest to project)
- Everything implemented to account for queueing delays/latency





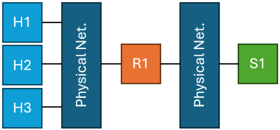
Discipline

- Handle prioritization/ordering of traffic
- Traffic incoming from HW layer is queued by flow
- Discipline pulls from flows and stores into output queue
- Three disciplines of interest
  - FIFO: Single input queue
  - Round Robin: Equally service multiple flows
  - Fair Queue: Fairly service multiple flows



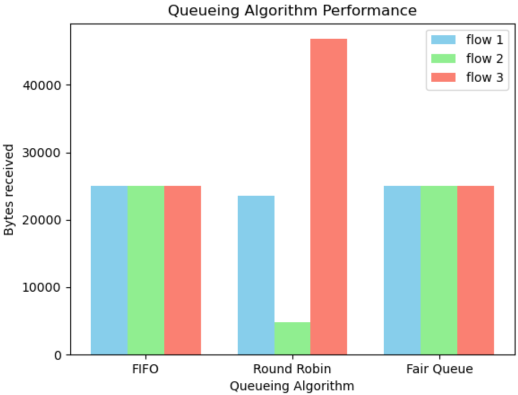
Test Setup

- Two networks
  - Input network high bandwidth
  - Output network variable (low, med, high)
- Three flows
  - Equivalent bandwidth (100 bs)
  - Different levels of packets (5/s, 1/s, 0.5/s)
- One router with tested discipline

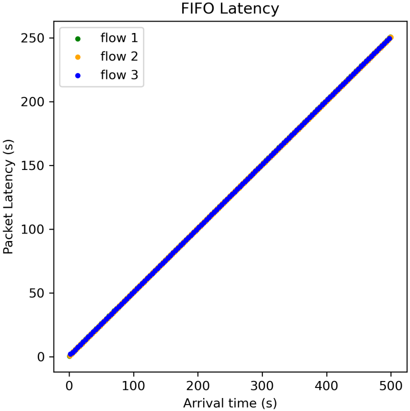


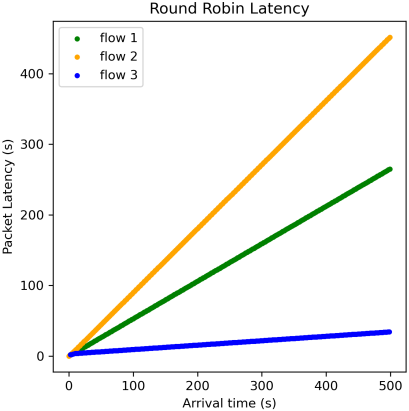
Results - Fairness

Flow	P. Size	Period
1	100	1s
2	20	0.2s
3	200	2s

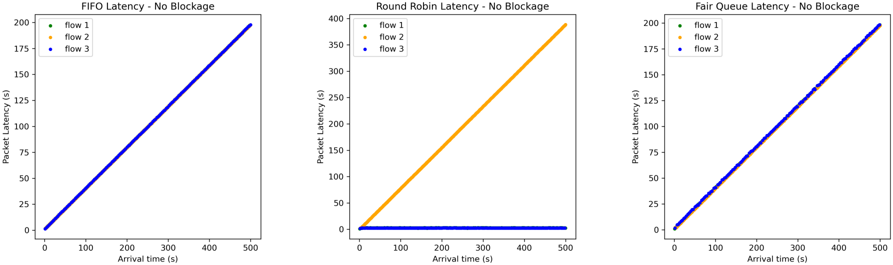


Results - Latency (congested)

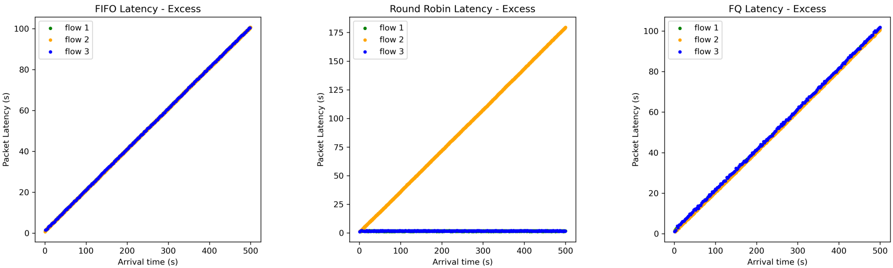




Results - Latency (equal)



Results - Latency (excess)



Conclusions

- Interesting Result: A single queue has the same result as a fair queue (for this test case)
- Round robin is a terribly unfair strategy for variable packets
  - Even when the queues are moving
- Modifying existing network simulation tools is dense work
  - Would be great to have better documentation/compilation (thinking about NS-3)

Key Takeaways

1. More advanced strategies aren't always better: sometimes a single queue can do the same work.
2. Real time networks face a lot of challenges; unlike OS, networks are inherently less predictable.
3. Real time networks have properties which aren't as easy to manipulate in existing simulators, like NS-3