Observations in Darwin

Jon Schuringa

June 26, 2002

Contents

- Introduction
- Four Scenarios
  - Large Rings
  - Staggered Bottlenecks
  - On-Off Sources
  - High/low bandwidth flows
- Conclusions

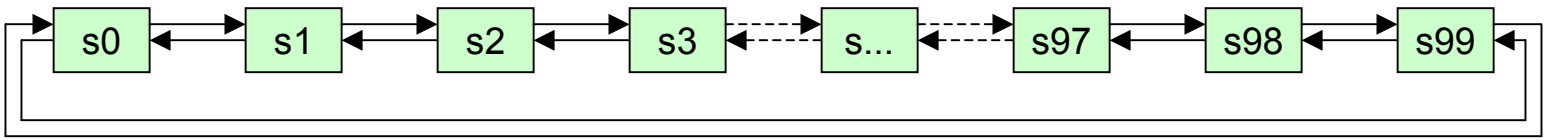
Other simulation results appended

Introduction

- Darwin
  - Simulation Model according to latest fairness description (cls09_fairness_701.pdf)
- Cyclic Queuing Multiple Access (CQMA)
  - New name for IKN proposal
  - Presented at the 7th European Conference on Networks & Optical Communications, June 18-21, 2002
  - CQMA-1: Optimal scheduler
  - CQMA-2: Simple, scalable, sub-optimal

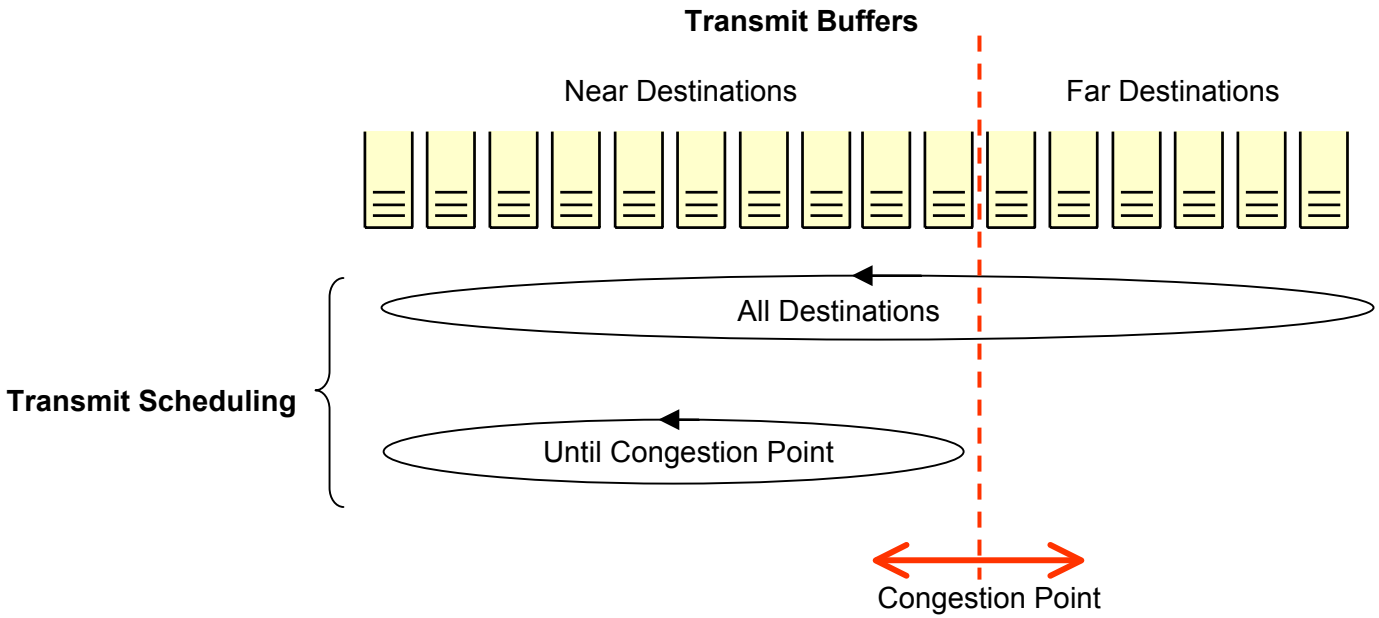
Large Rings (1)

- Scenario:
  - 100 Stations
  - Dual OC-12 Ring, traffic routed over ringlet 0
  - Low priority traffic
  - Uniform saturated, all to all traffic (100*99 flows)



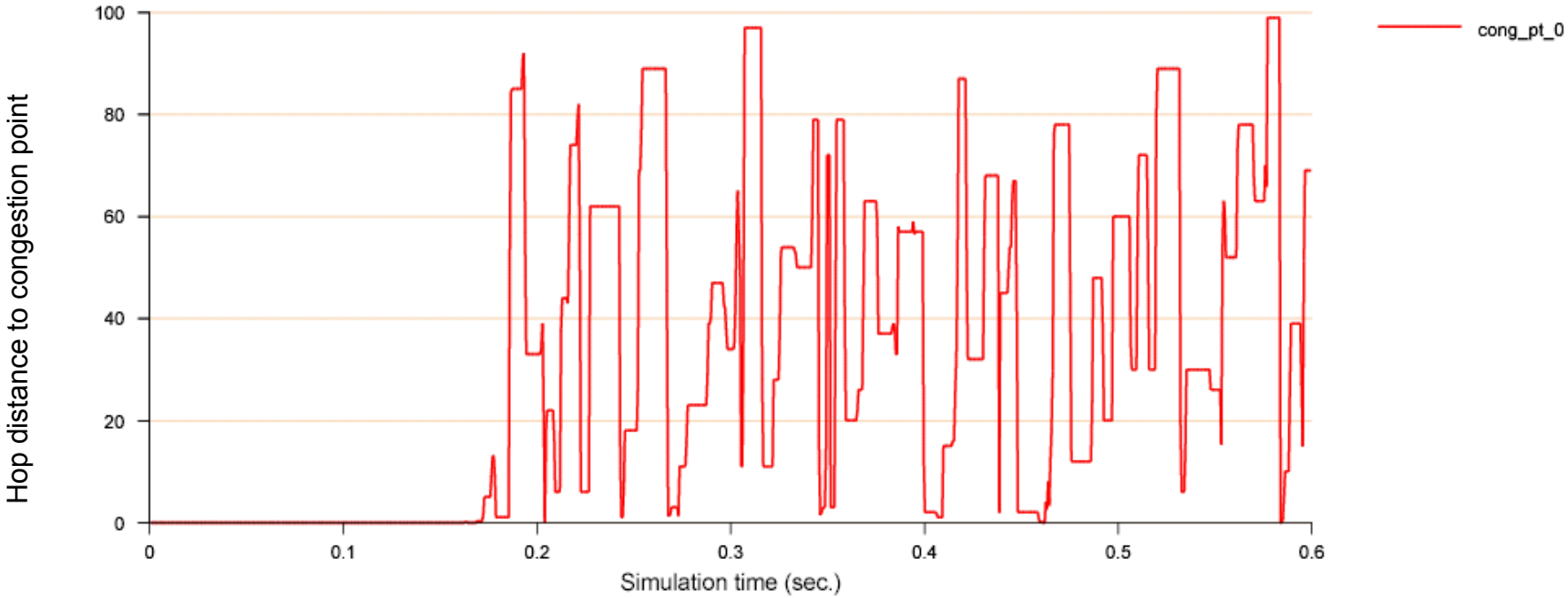
Large Rings (2)

- Station can be in two states if allowed to send:
  - To all destinations
  - To destinations until congestion point (add_rate_congestion>allow_rate_congestion)
- Congestion point is moving



Large Rings (3)

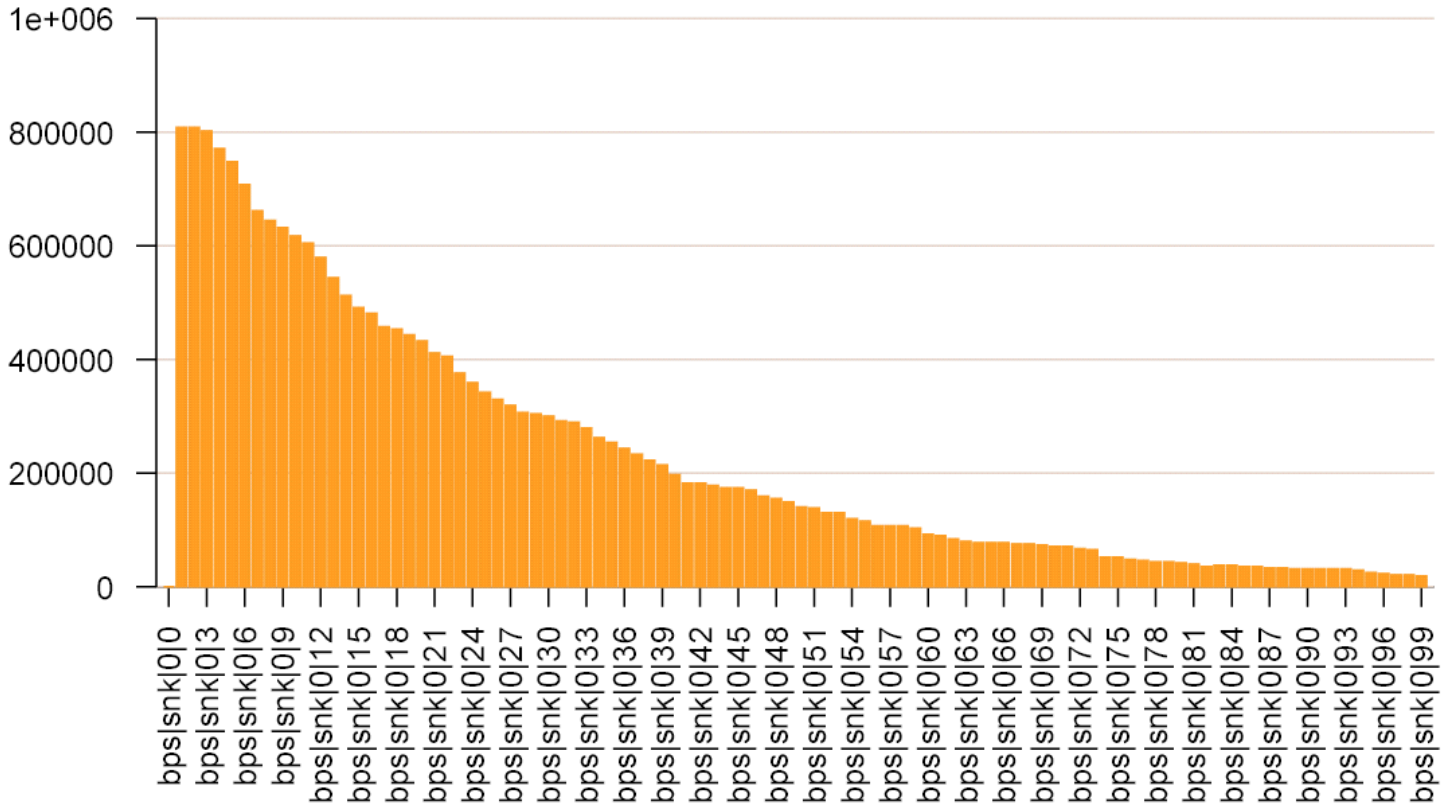
- Congestion point is not stable and can be anywhere



Hop distance to congestion point observed by node 0

Large Rings (4)

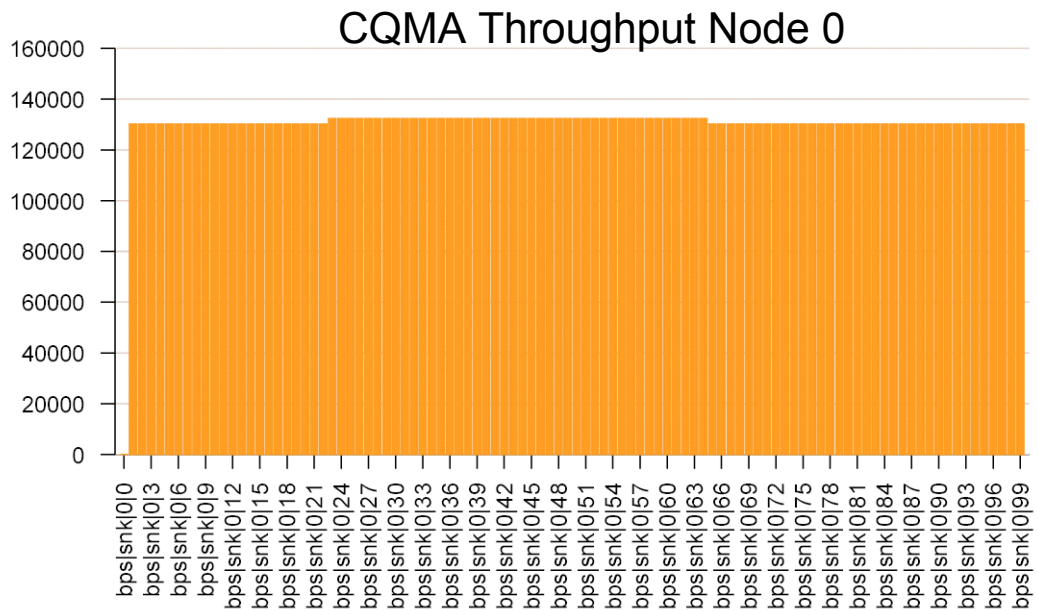
- All nodes get a fair share, but this is not equally divided amongst the destinations. The queuing discipline in the client is *not* responsible for this.



Darwin Throughput to all destinations from node 0

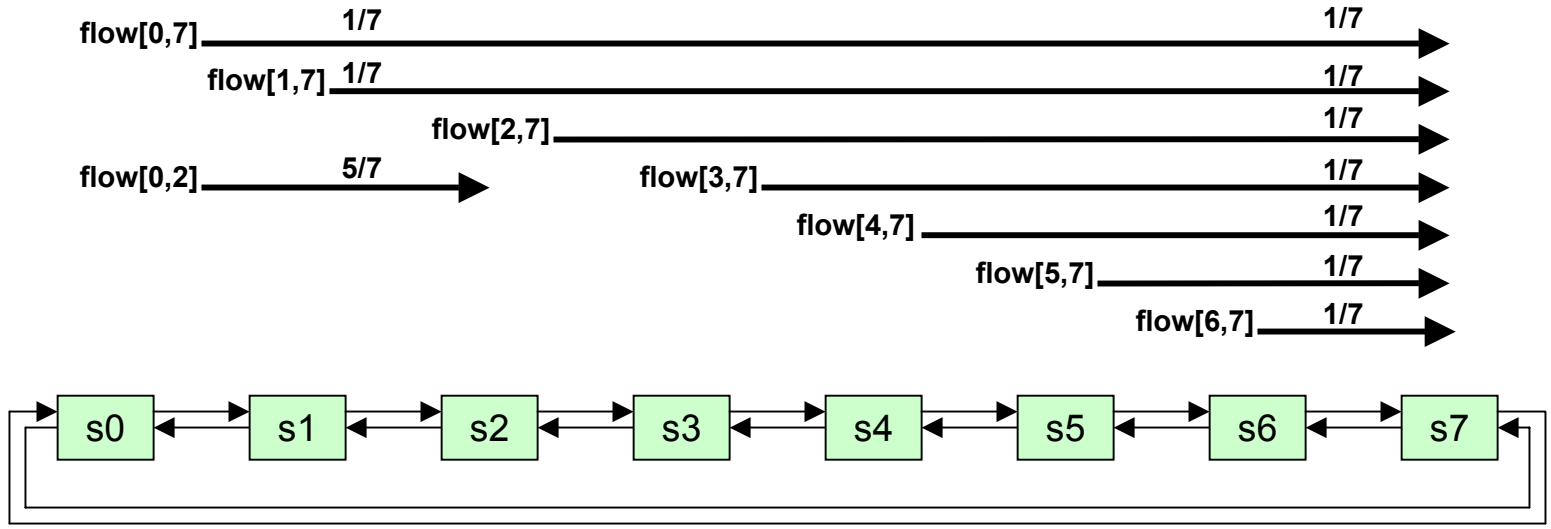
Large Rings (5)

- Nearby destinations get more bandwidth, far away nodes get almost nothing
  - Reason: “single choke” implementation
  - Queuing discipline (e.g., round robin, …) in MAC client is not the reason
- Solution
  - Multi choke
  - Other fairness algorithms



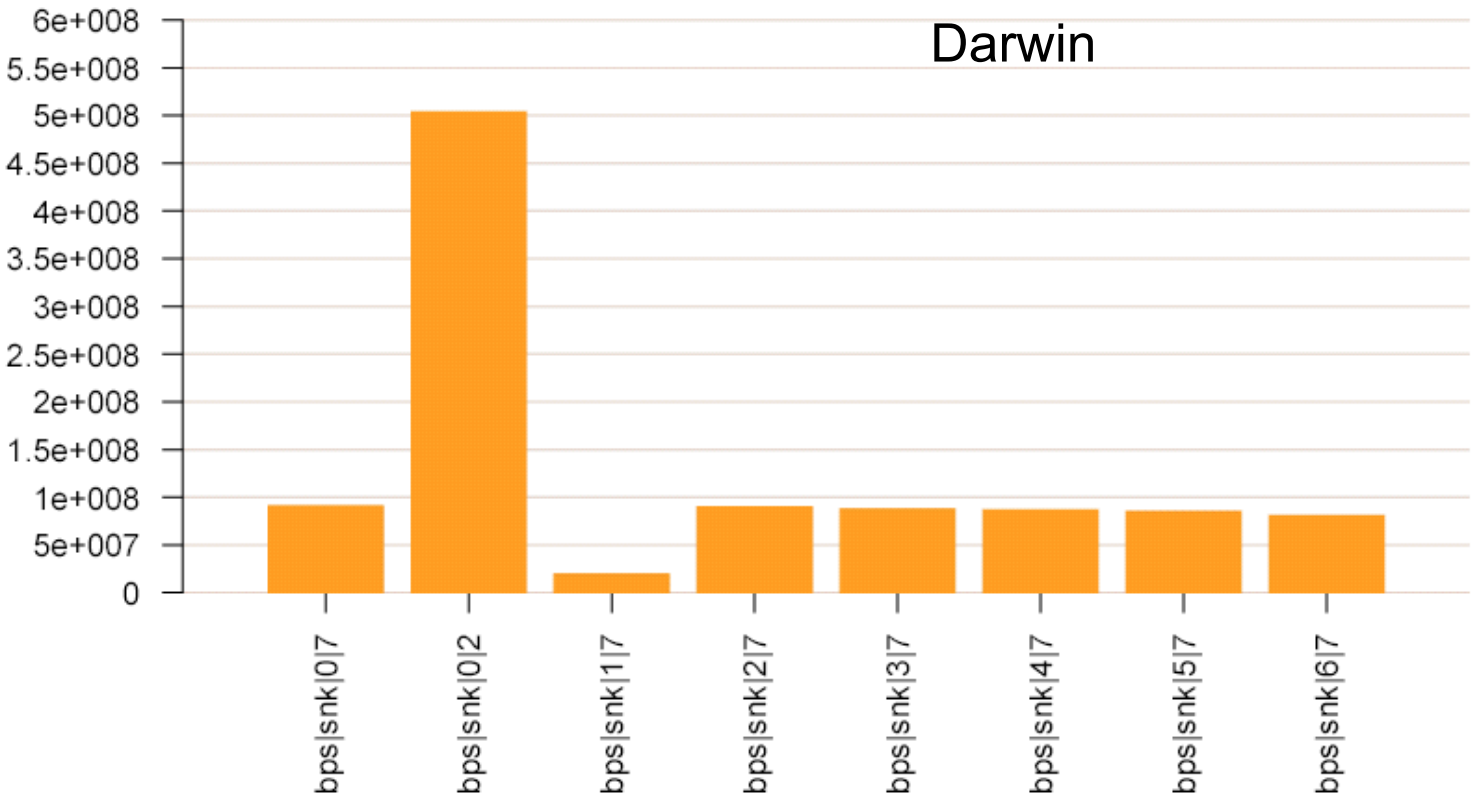
Staggered Bottlenecks (1)

- Two bottlenecks, link 6->7 and link 1->2
- Problem: Station 0 could see only the strongest bottleneck 6->7, thereby giving flow 0->2 too much bandwidth.



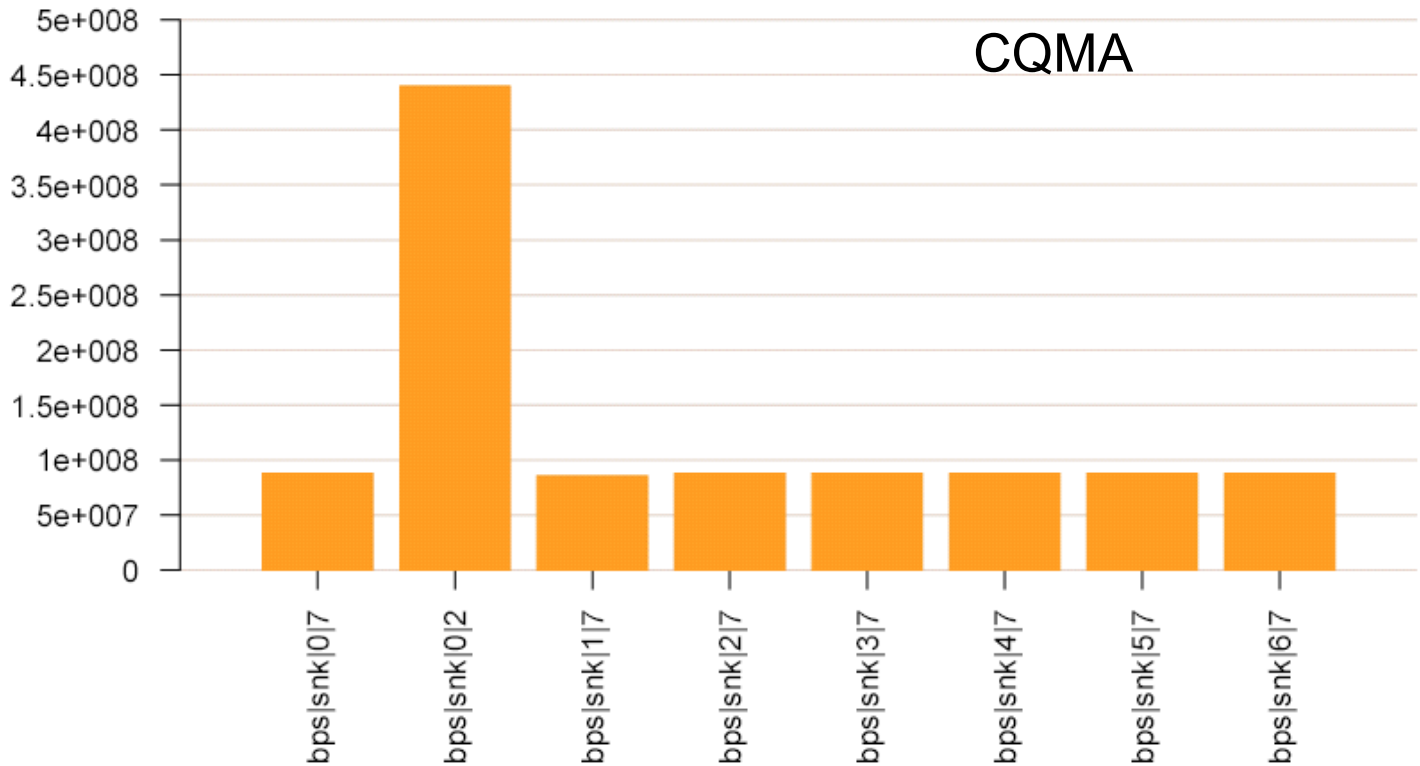
Staggered Bottlenecks (2)

- Bottleneck is moving between link 6->7 and 1->2, causing a very small throughput between stations 1 and 7



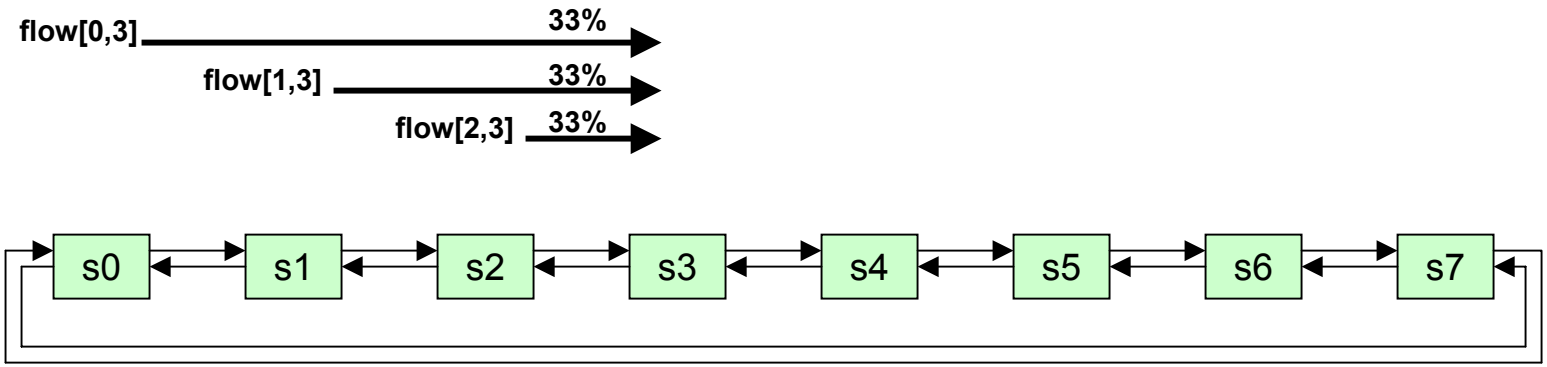
Staggered Bottlenecks (3)

- Multi choke, or other fairness algorithms could solve the problem
- CQMA shows correct behavior



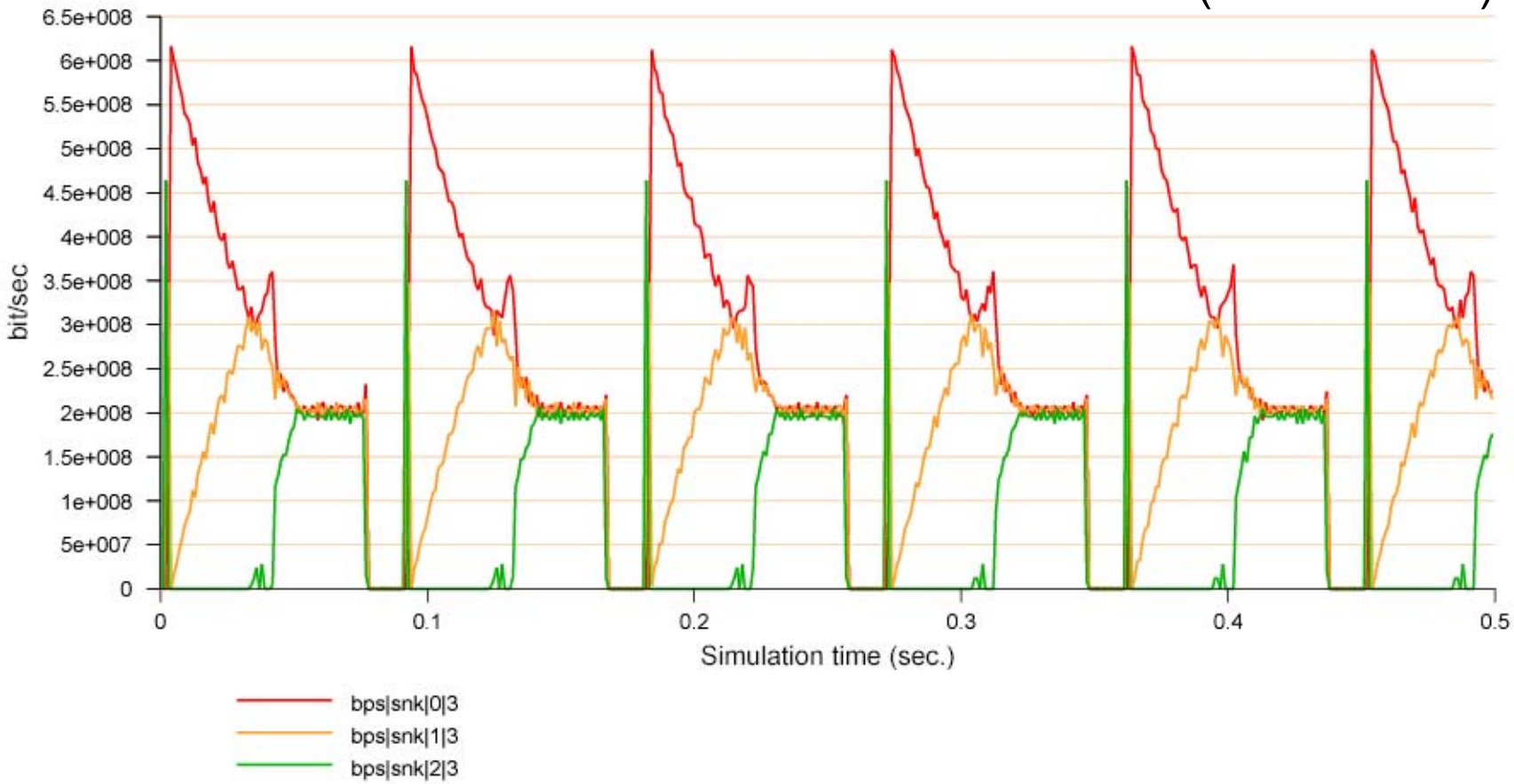
On-Off Sources (1)

- Three identical on-off flows
- During the “on phase”, all flows should get 33% of the available bandwidth



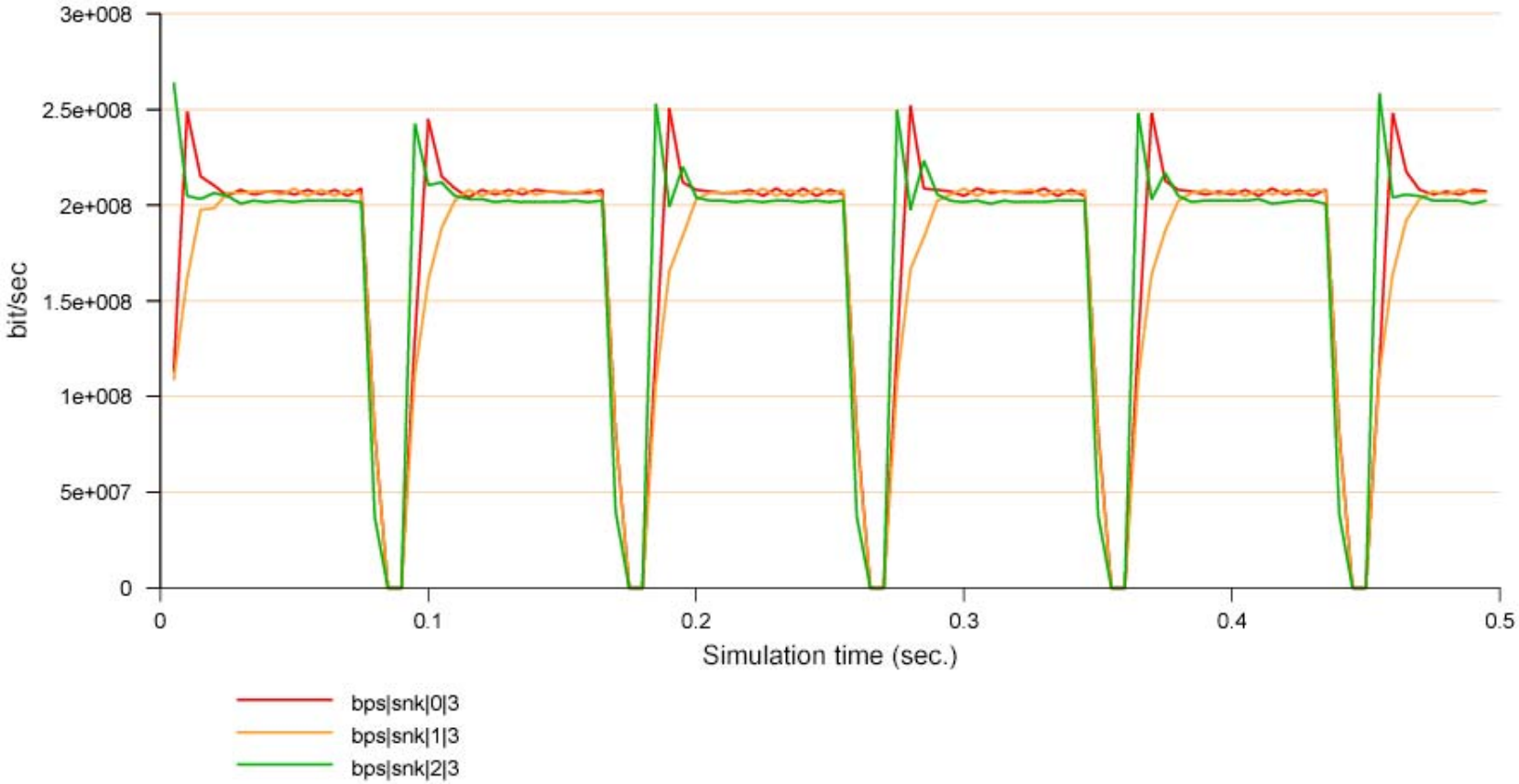
On-Off Sources (2)

Darwin (conservative)



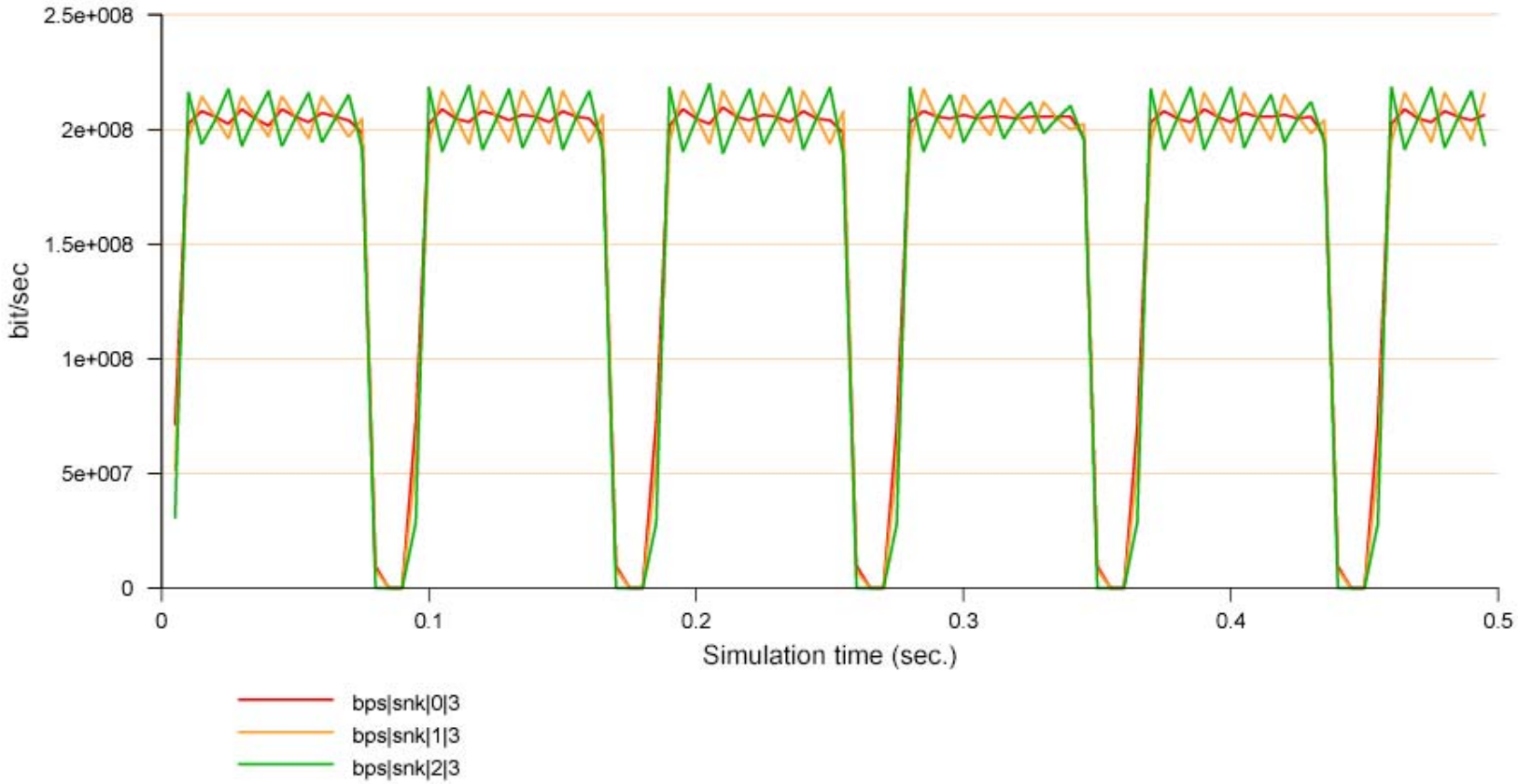
On-Off Sources (3)

Darwin (aggressive)



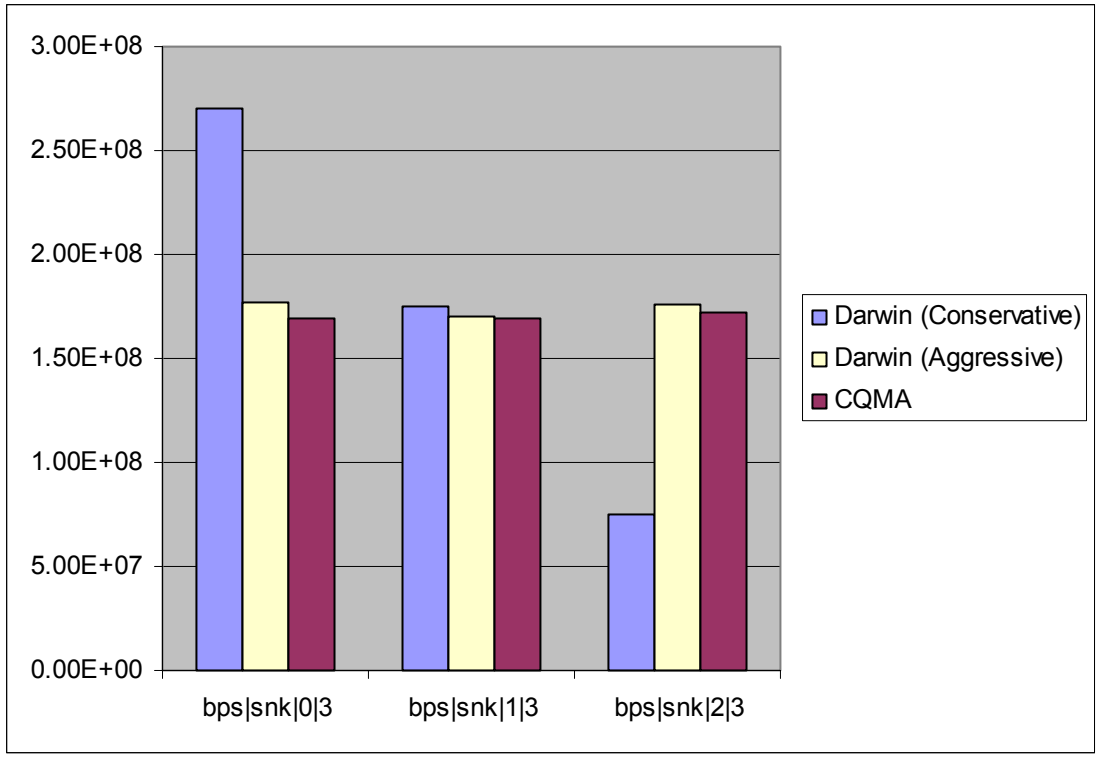
On-Off Sources (3)

CQMA



On-Off Sources (4)

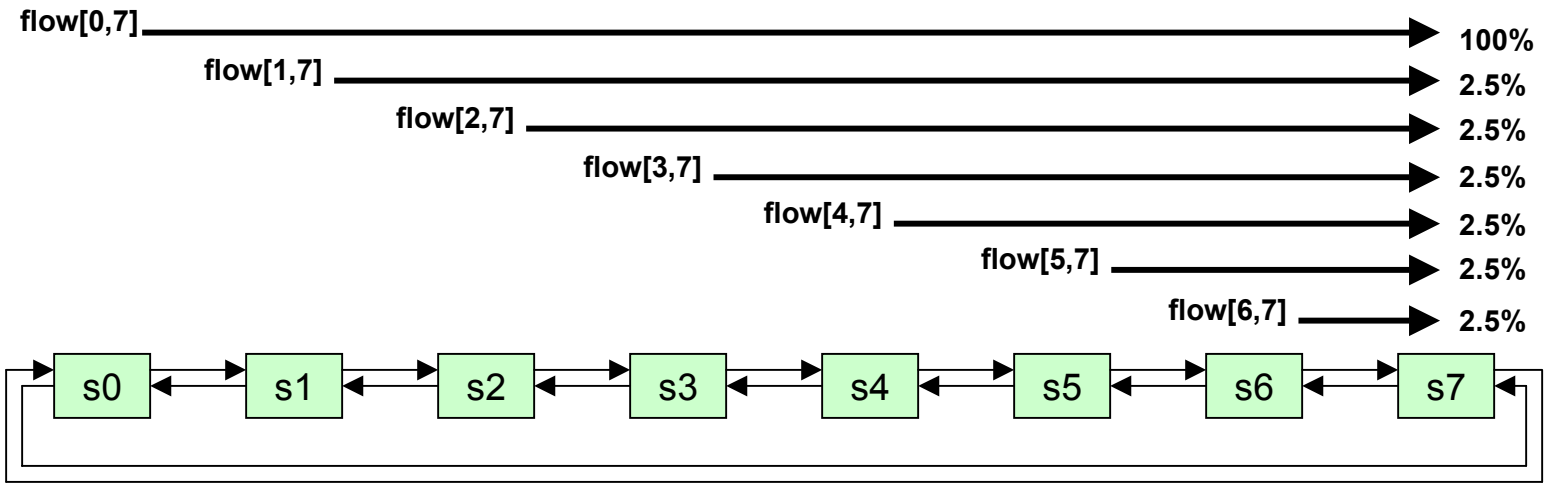
- Darwin in conservative mode responds relative slow to traffic changes, this may cause unfairness
- Aggressive mode is fair in this scenario, but may lead to instability in other scenarios



High- and low-bandwidth sources (1)

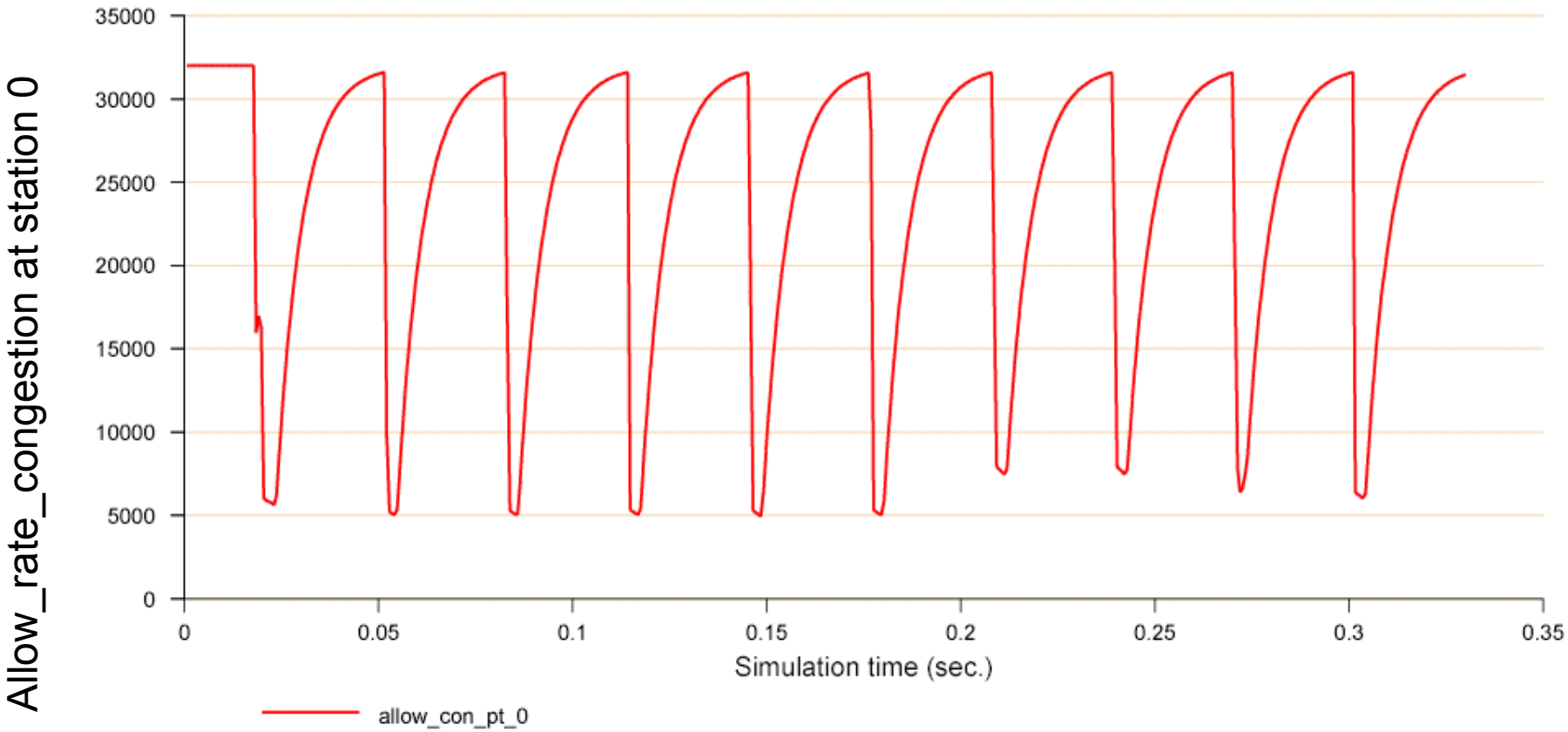
- Flow 0->7 wants full bandwidth, all other flows only 2.5%
- In theory, flow 0->7 should get 85%

- Problem:
  - Flow 0->7 is oscillating, causing throughput loss

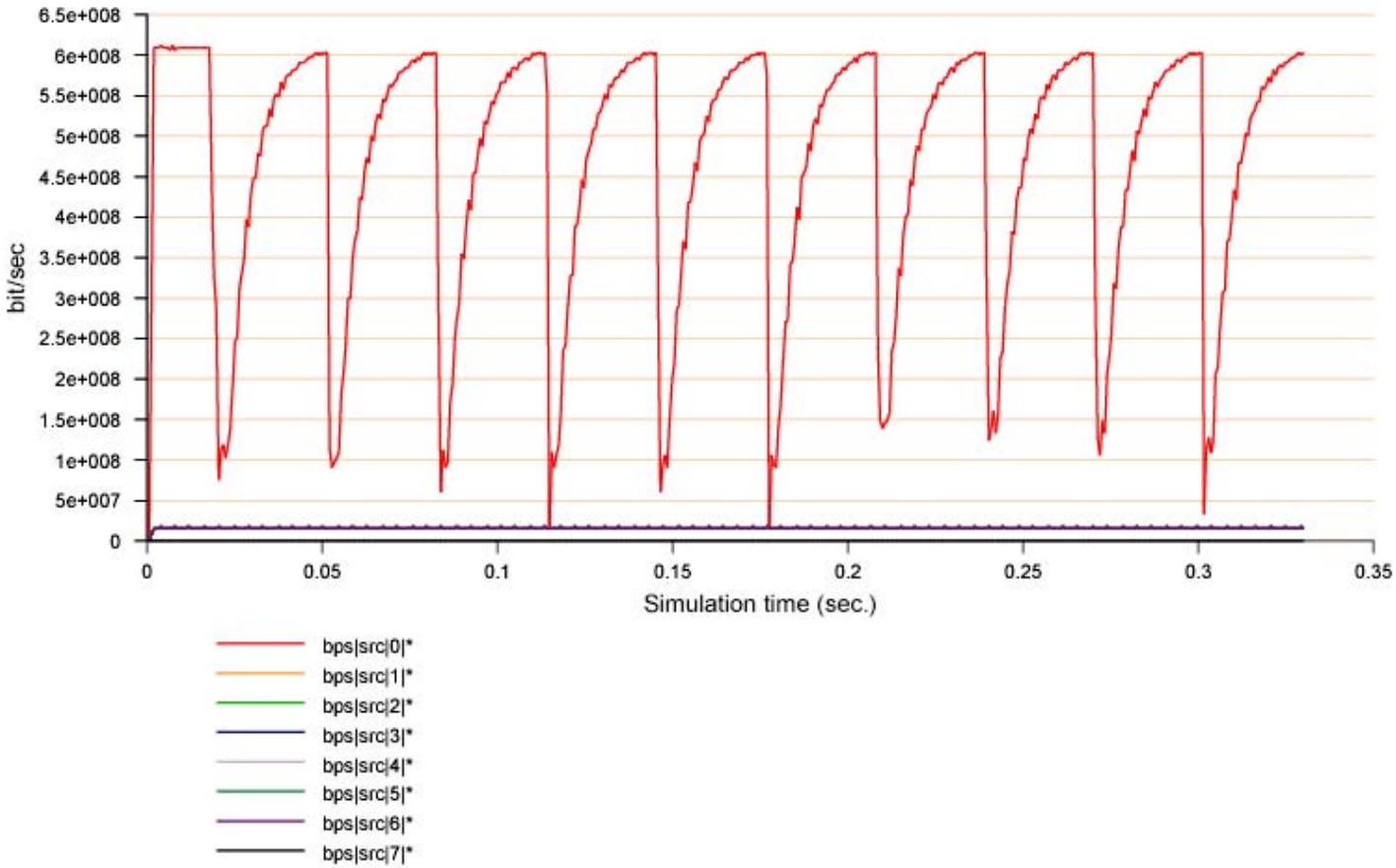


High- and low-bandwidth sources (2)

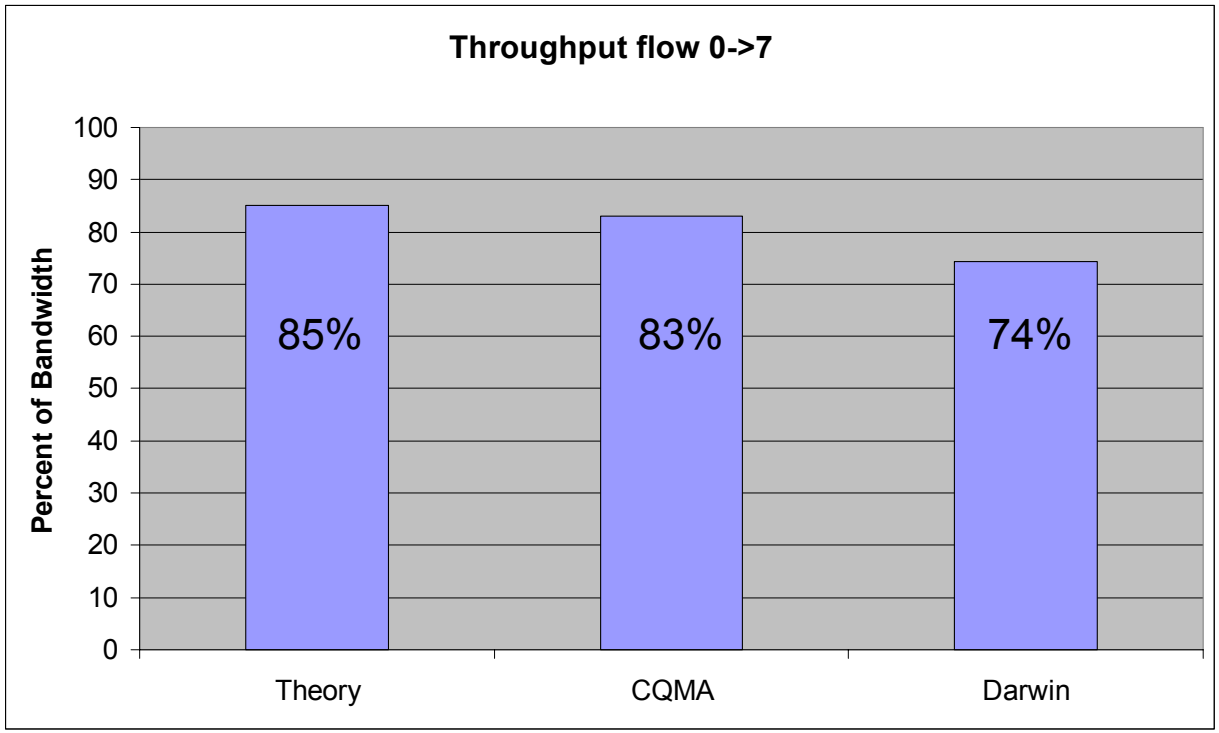
- Initial advertised rate is based on number of active stations



High- and low-bandwidth sources (3)



High- and low-bandwidth sources (4)



Conclusions

- Single choke concept may lead to
  - Unfairness
  - Less throughput
- Conservative mode may be too slow to adapt to dynamic traffic changes, resulting in unfairness