Implementation of Start-Time Fair Queuing Algorithm in OPNET

CMPT885/ENSC835 Final Project

by

Daryn Mitchell daryn@cs.sfu.ca

Jack Man Shun Yeung yeung@cs.sfu.ca

April 4, 2002.

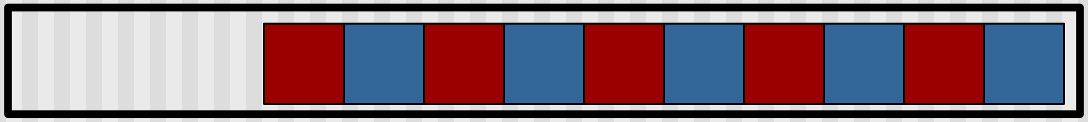
Road map

- Background and Introduction to *Start-Time Fair Queuing* (SFQ)
- Project Implementation
- Some Simulation Results
- Conclusion
- References

Queuing in Routers - FIFO

- Buffers in routers allow multiple sources share one outgoing link
- Router schedules order in which packets leave
- Almost all FIFO/FCFS queuing

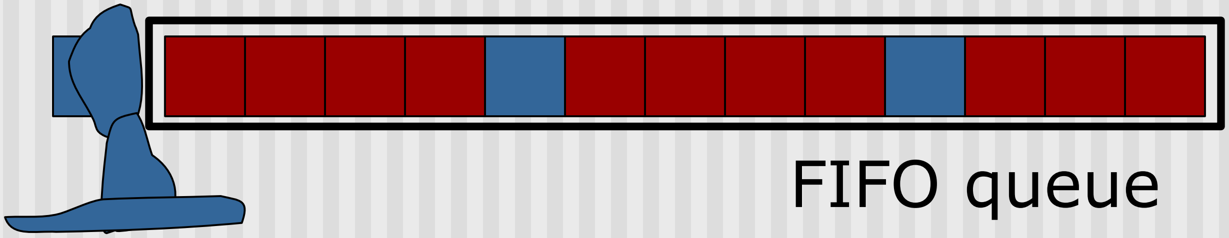
  (First-In, First-Out / First-Come First-Serve)
  - Simple, Cheap
  - Yet: Unfair?



FIFO queue

Queuing in Routers – FIFO unfair

- No bandwidth guarantee in the presence of congestion
- Misbehaving sources can steal bandwidth from behaving sources



FIFO queue

Fair Queuing (FQ)

- Concept: Everyone gets a fair share of router bandwidth [Nagle RFC970, 1985]
- Assign a 'weight' to each data flow or category

	(Cost)	
WFQ [Demers89]	Expensive	✘ Unfair for VBR
WF^2Q [Bennett96]	Expensive	
DRR (Deficit Round Robin) [Shreedhar95]	Cheap	✘ Not guaranteed
Virtual Clock[Zhang90]	Okay	✘ Punishes use of idle time [Parekh93]

Start-Time Fair Queuing (SFQ)

- Proposed in 1996 by Goyal,Vin and Cheng [Goyal96]
- Like most FQ, assign weights to flows to give minimum bandwidth and maximum delay guarantees.

✓ Simple algorithm (unlike WFQ)

✓ Fair for VBR (unlike most FQ algrothms)

✓ Fair use of idle bandwidth (unlike Virtual Clock)

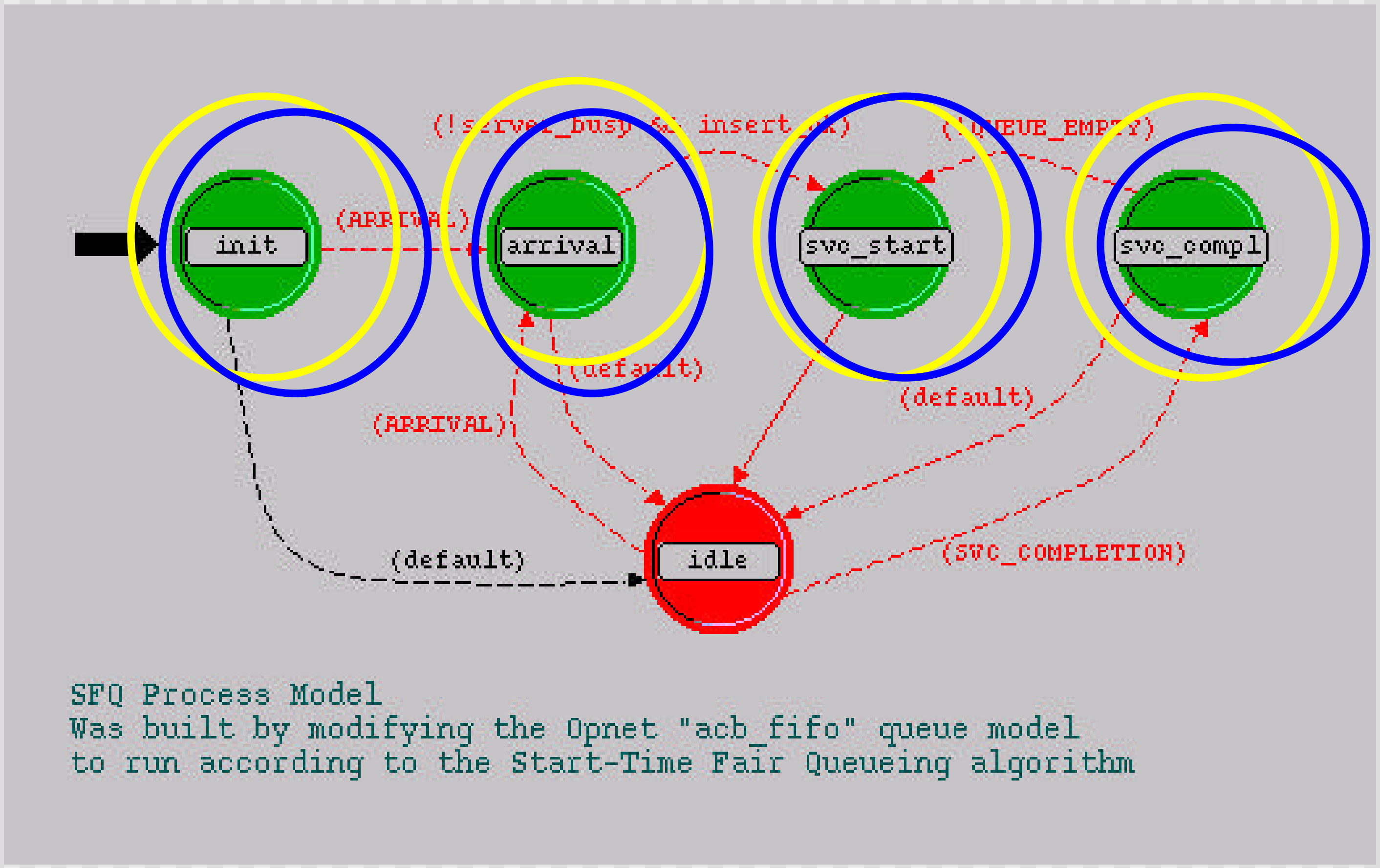
SFQ Overview

- Basic idea:
  - Choose packet to send by lowest "start tag" among all flows
  - Start Tags in a flow generally increment by: $\frac{\textit{Packet Size}}{\textit{Flow Service Rate}}$
    - If packets arrive as agreed, they are all served with guaranteed maximum and average delay
    - If packets arrive faster than agreed, they will be delayed so they don't affect other flows

Project Proposal

- Implement SFQ in OPNET as a queue module
- Create network incorporating SFQ in a router
- Run simulations to compare performance with other scheduling algorithms (e.g. WFQ and Virtual Clock)
- If possible, test with a real traffic trace

Implementation: Queue Process Model

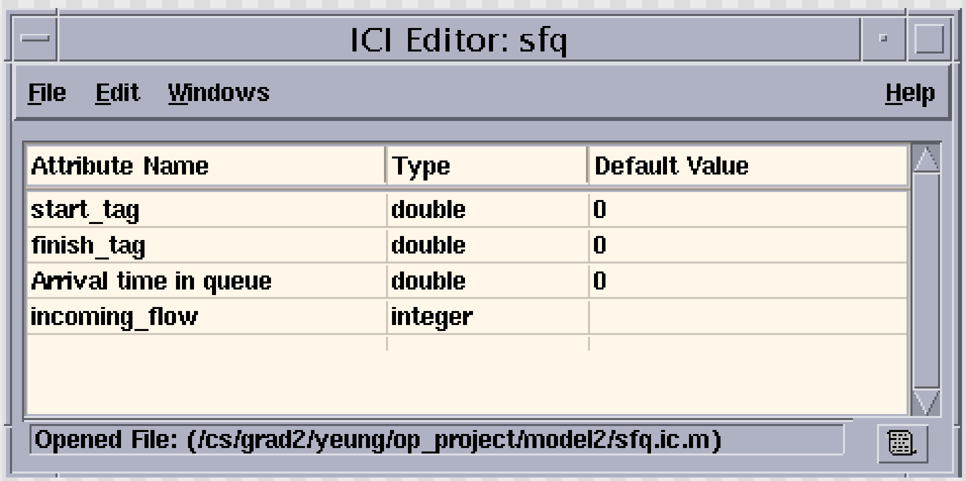


Implementation: ICI

- Opnet “Interface Control Information”
  - Use to associate data fields with packets

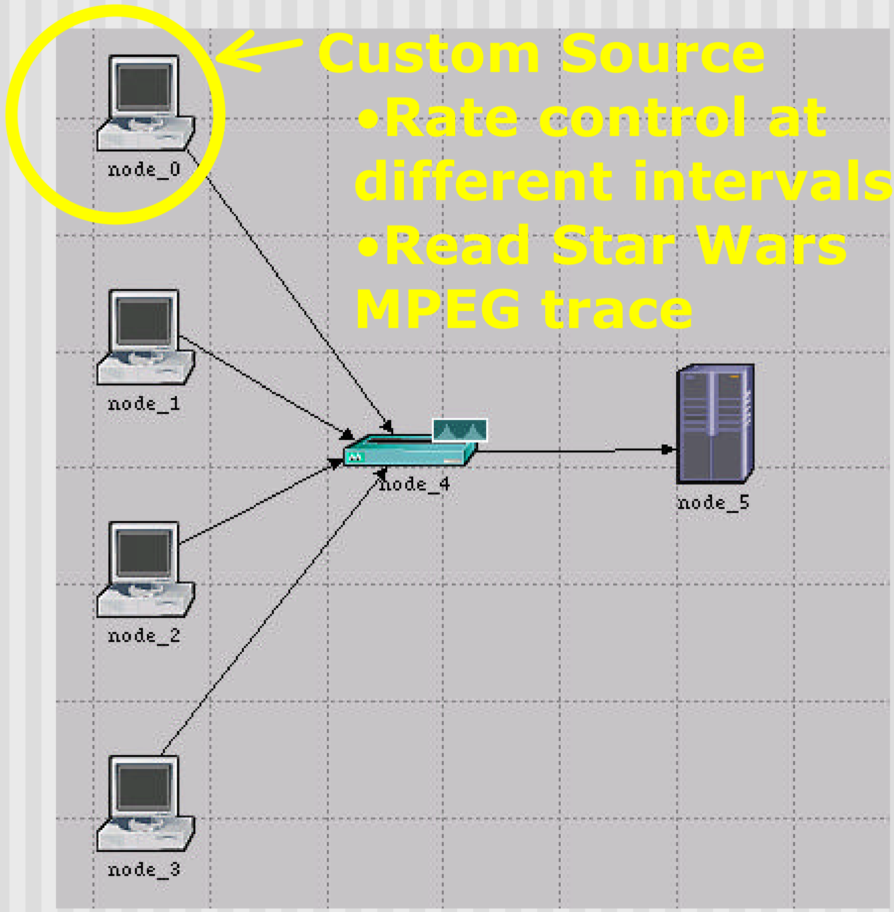
- SFQ fields:
  - Start Tag
  - Finish Tag

- VC fields
  - Virtual clock
  - Auxiliary Virtual Clock



Attribute Name	Type	Default Value
start_tag	double	0
finish_tag	double	0
Arrival time in queue	double	0
incoming_flow	integer	

Implementation: Network Model



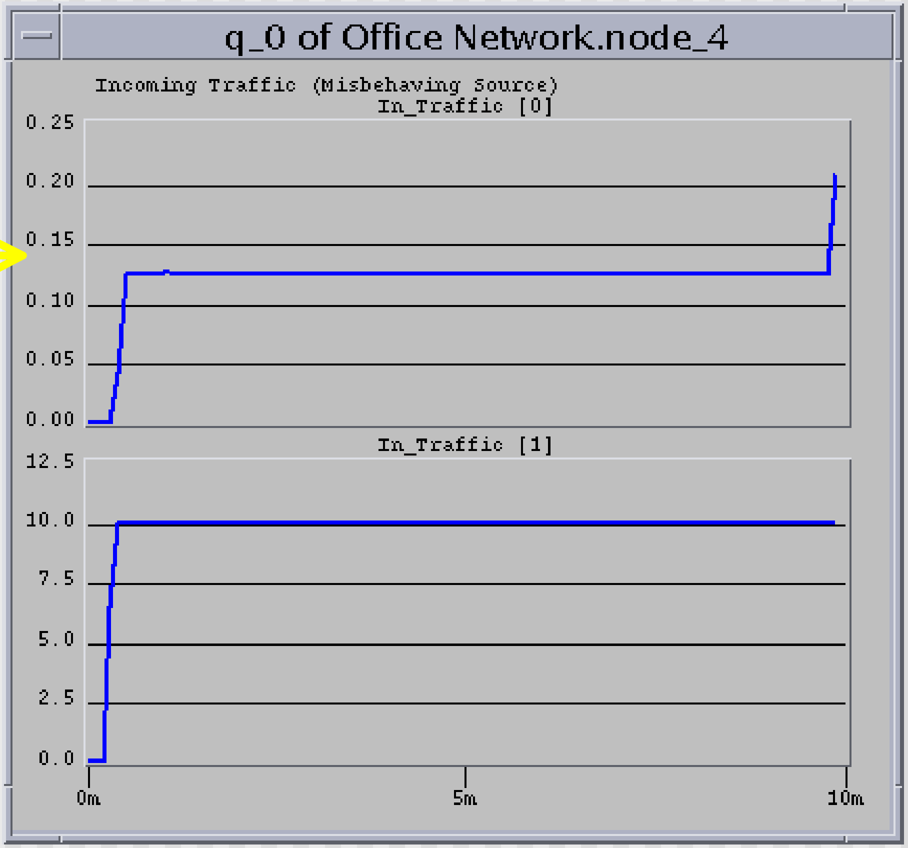
Result 1: FIFO unfair

Incoming traffic per flow

1 Pk/ 8 sec

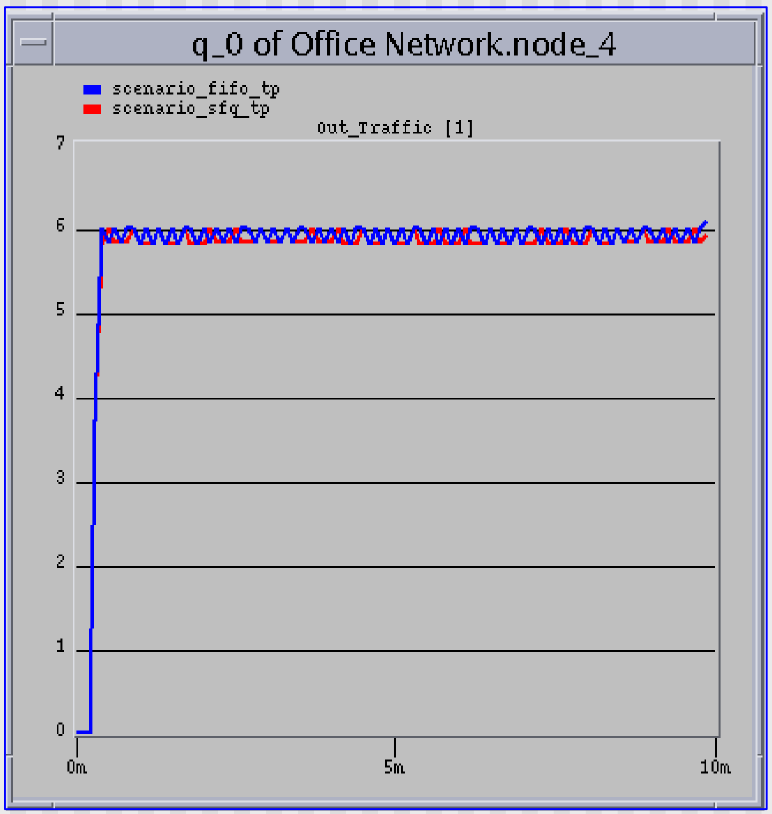
10 Pks/sec

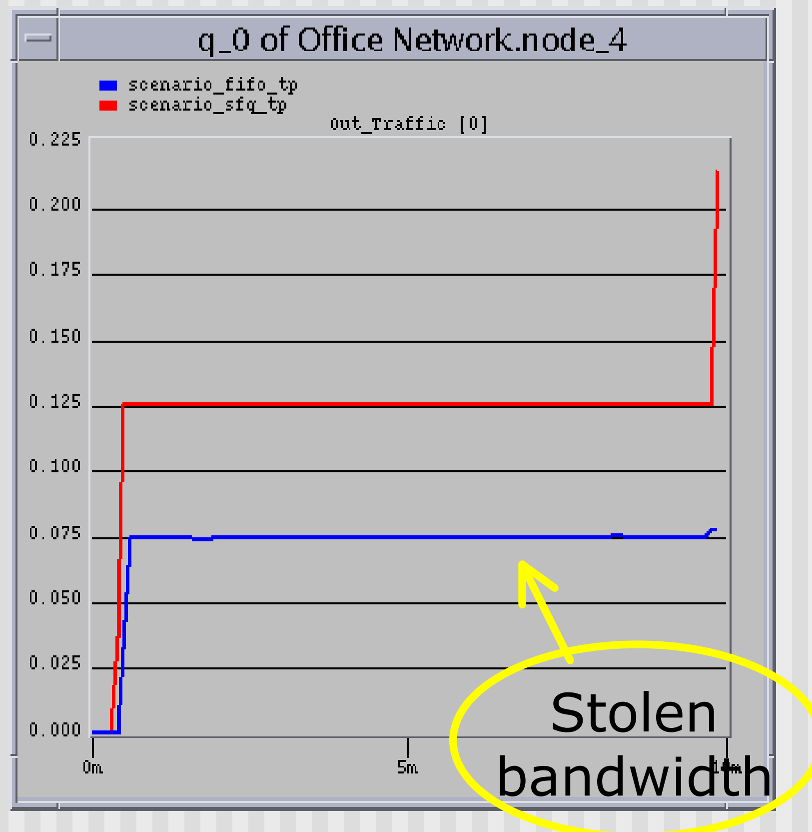
Misbehaving!



Result 1:
FIFO unfair

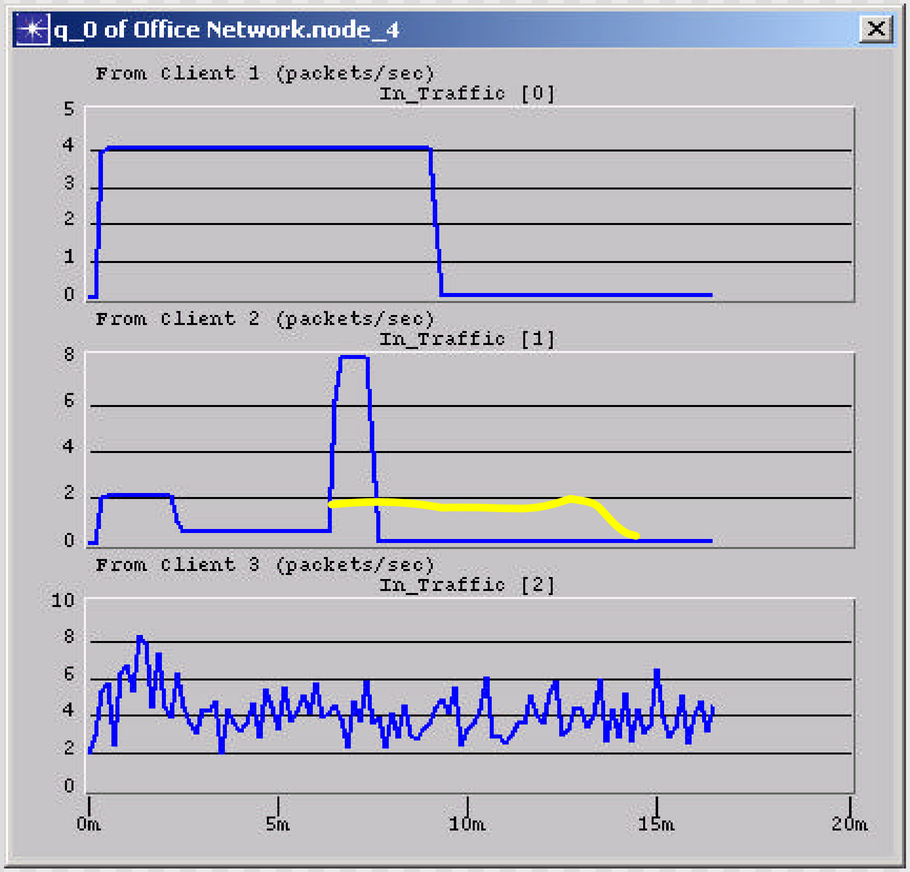
Throughput per flow





FIFO = SFQ =

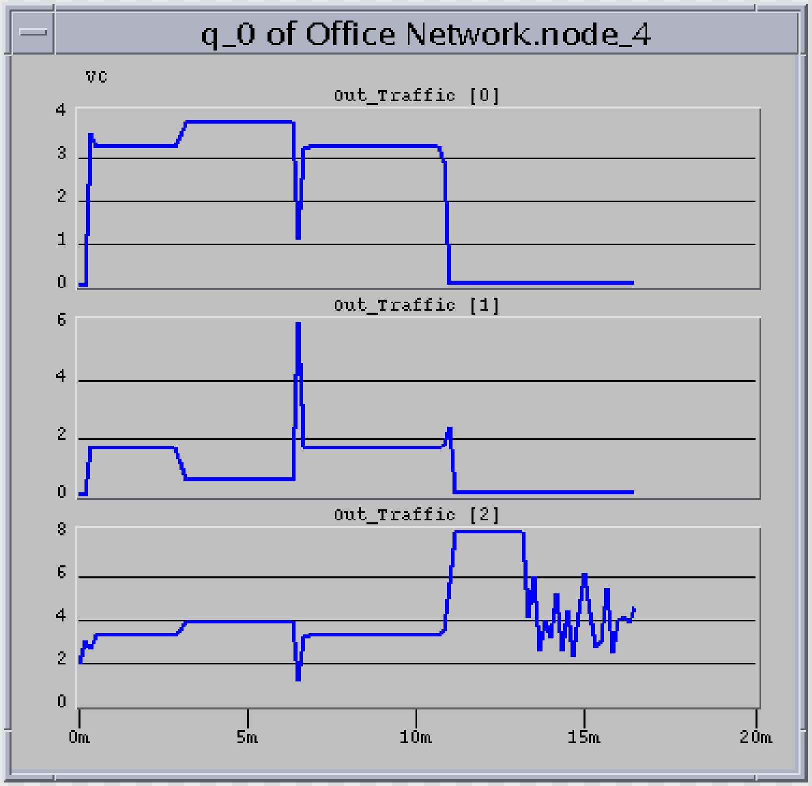
Result 2:
Verification of Fairness

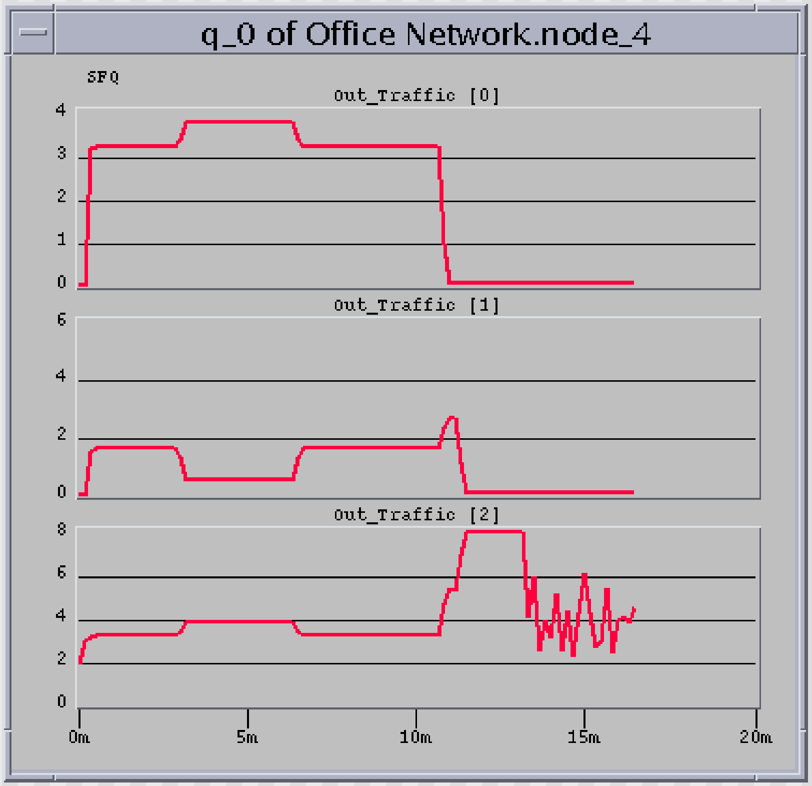


- Source traffic as in VC paper [Alborz01]
  - src1: CBR 4 pk/s
  - src2:misbehaves
    - Fair queues make it wait
  - src3: VBR, avg. 4 pk/s
- Congestion

Result 2: Verification of Fairness

Throughput per flow

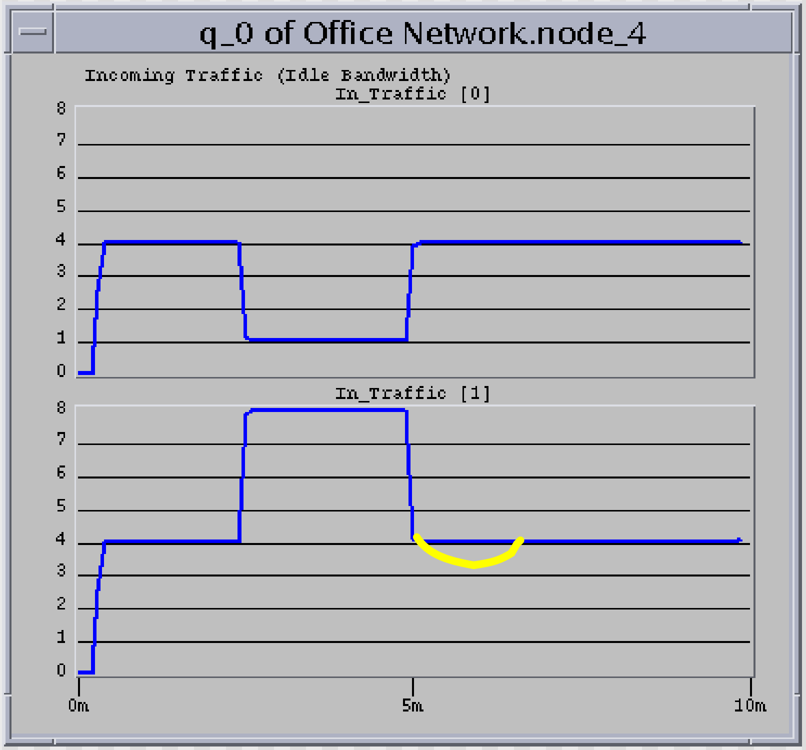




Virtual Clock =

SFQ =

Result 3: Fair Use of Idle Bandwidth

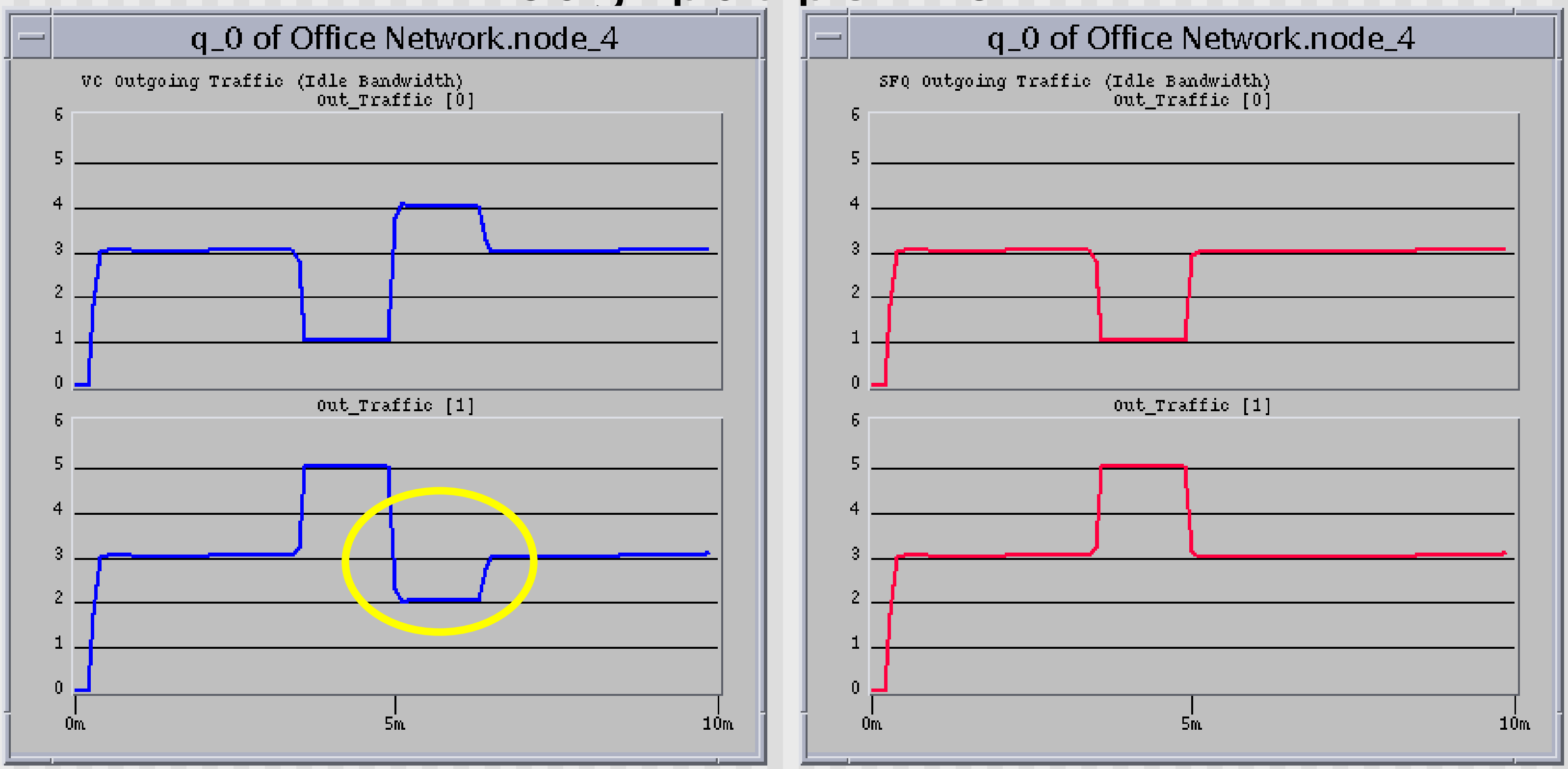


Input Traffic per flow

- 2 clients
  - Client 0 (4 pks/sec) has a period of low arrival rate
  - Client 1 (4 pks/sec) has a period of high arrival rate at same time
  - There is congestion (8 pks/sec sharing 6 pks/sec)

Result 3:
Fair Use of Idle Bandwidth

Throughput per flow



Virtual Clock =

SFQ =

Conclusion:
Project Proposal - Revisited

- ✓ Implemented SFQ in OPNET as a queue module
- ✓ Created network incorporating SFQ in a router
- ✓ Ran simulations to compare performance with other scheduling algorithms (FIFO, WFQ and Virtual Clock)
- ✓ Used a real traffic trace (Star Wars MPEG)
- ✓ Converted/Implemented Virtual Clock as a queue module in Opnet

References (1)

[Alb01] N. Alborz and Lj. Trajkovic, "Implementation of VirtualClock scheduling algorithm in OPNET" OPNETWORK 2001, Washington, DC, Aug. 2001.
http://www.ensc.sfu.ca/~ljilja/papers/opnetwork01_nazy.pdf (14.Feb.02)

[Chu02] T.A. Chu, "Police Your Packets: Traffic Management Part 1; Keys to Implementation" CommsDesign, February 1, 2002
http://www.commsdesign.com/story/OEG20020201S0007 (14.Feb.02)

[Cis95] "Interface Queue Management" Cisco Systems White Paper
http://www.cisco.com/warp/public/614/16.html (14.Feb.02)

[Der90] A. Demers, S. Keshav, and S. Shenker, "Analsyis and Simulation of a Fair-queueing Algorithm", Proc. ACM SigComm 89, pp1-12, also to appear in Journal of Internetworking, Vol. 1, No. 1, 1990.
http://netweb.usc.edu/cs551/papers/Demers.pdf (14.Feb.02)

[Goy96] P. Goyal, H. Vin, and H. Chen. "Start-Time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks". In Proceedings IEEE SIGCOMM'96, August 1996.
http://www.cs.columbia.edu/~danr/6762/week4/stfq.pdf (14.Feb.02)

References (2)

[Mo99] J. Mo, R. J. La, V. Anantharam, and J. Warland, ``Analysis and comparison of TCP Reno and Vegas,'' IEEE INFOCOM, New York, May 1999.

[Nag84] J. Nagle. "Congestion Control in IP/TCP Internetworks." Computer Communication Review, 14(4), October 1984.
http://www.acm.org/sigcomm/ccr/archive/1995/jan95/ccr-9501-nagle84.pdf (12.Mar.2002)

[Nag85] J. Nagle, "On packet switches with infinite storage" RFC970, Dec-01-1985
http://globecom.net/ietf/rfc/rfc970.html (14.Feb.02)

[Par93] A. K. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case. IEEE/ACM Transactions on Networking, 1(3):344--357, June 1993.
http://www.stanford.edu/class/ee384x/Papers/parekh.pdf (12.March.2002)

[Wal00] J. Walrand and P. Varaiya, High-performance Communication Networks, Second edition, Morgan Kaufmann, 2000.