



Implementation of VirtualClock Scheduling Algorithm in OPNET

Nazy Alborz and Ljiljana Trajkovic
{nalborz, ljilja}@cs.sfu.ca

Communication Networks Laboratory
School of Engineering Science
Simon Fraser University



Road map

- QoS
- Packet scheduling
- VirtualClock algorithm
- VirtualClock implementation
 - Process model
 - QoS configuration object
- Model verification
- Simulation results
- VirtualClock comparison with other scheduling algorithms
- Conclusion and future work



Quality of Service (QoS)

- Ability to provide guaranteed services to different applications.
- Measured in terms of:
 - throughput
 - loss
 - end to end delay
 - delay jitter.
- Important in today's networks running applications with various performance requirements.



Providing QoS

- End to end mechanisms:
 - operate at the two ends of a connection (call acceptance control).
- Edge mechanisms:
 - operate at user-network interface (shaping and policing).
- Core mechanisms:
 - operate at network switching nodes like routers and switches (buffering, queue management, and **scheduling**).



Packet scheduling

- Provides QoS.
- Performed at:
 - network switching nodes (routers, switches)
 - output interface of switching nodes
 - usually belongs to the network layer.
- Allows statistical multiplexing.
- Provides protection between traffic streams.



Packet scheduling (cont.)

- Determines:
 - which packets get transmitted
 - when packets get transmitted.
 - which packets are discarded upon buffer overflow.
- Promises:
 - guaranteed throughput
 - bounded end to end delay and delay jitter
 - limited loss probability.



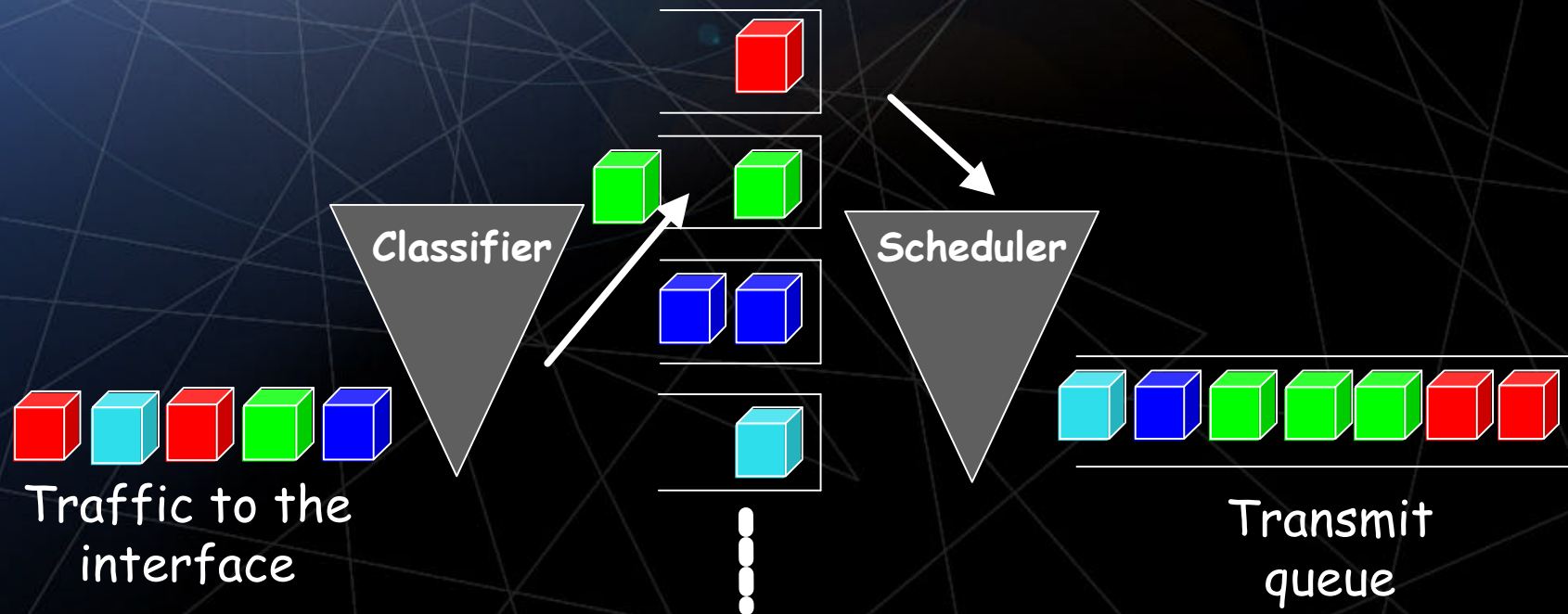
Scheduling mechanisms

- An ideal scheduling mechanism should:
 - be fair
 - be easy to implement
 - provide performance bounds.
- Two categories of scheduling mechanisms:
 - work-conserving: never idle when there is packet waiting in output queues (WFQ and VC).
 - non-work-conserving: waits until the packet's assigned sending time (Stop and Go).



Scheduler model

- Two main components:
 - Classifier: allocates packets to different queues
 - Scheduler: selects packets to send from the queues.





VirtualClock mechanism

- Derived from a Time Division Multiplexing (TDM) system.
- Allocates guaranteed throughput to traffic flows according to their expected packet arrival rates.
- Provides firewalls between traffic flows.
- Statistically allocates bandwidth to traffic flows.
- The algorithm performs two functions:
 - Data forwarding: allocates queues to traffic flows and decides service order of the packets.
 - Flow monitoring: the flow's conformance is checked and, if needed, a control message is sent to the source.



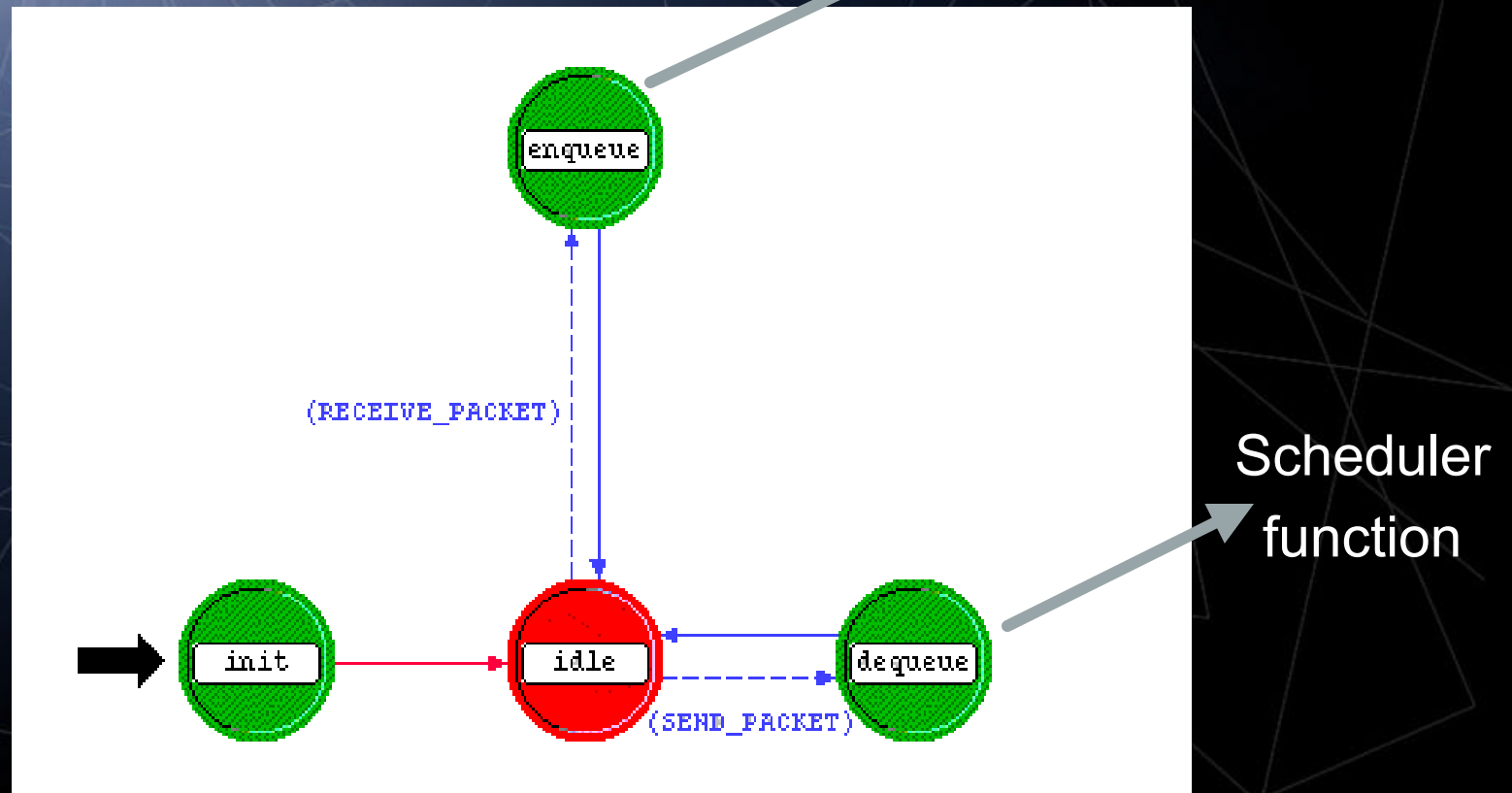
How data forwarding function works?

- Each data flow has a virtual clock.
- Each clock advances upon packet arrival from the flow.
- The tick step is equal to flow's packet inter arrival time.
- Arriving packets are stamped with flow's:
 - virtual clock
 - aux virtual clock: updated virtual clock to real time.
- Packets are sent in ascending order of their **aux virtual clock** stamps.



The process model

Classification function



State transition diagram of the VirtualClock algorithm.



Enqueue state

- Get the incoming packet.
- Allocate a queue to the packet.
- When first packet arrives to the queue, initialize virtual clock and aux virtual clock.
- **If** there are no packet currently transmitted:
 - extract the packet from queue
 - calculate the packet transmission time
 - issue an interrupt for (simulation time + transmission time).
- **Else:**
 - calculate the tick step
 - update virtual clock and aux virtual clock values
 - stamp the packet with values.
- Return to the idle state.



Dequeue state

- Send the packet to the link interface.
- **If** no packet in any queue:
 - return to the idle state.
- **Else:**
 - choose the packet with the lowest aux virtual clock stamp
 - extract the packet from the queue
 - calculate the packet transmission time
 - issue an interrupt at (simulation time + transmission time).
- Return to the idle state.



QoS configuration

- VC profile compound attribute is added to common attribute object "QoS Configuration object".
- Queues' attributes, configurable by user are defined as:

Arrival Rate	Maximum Queue Size	Classification Scheme	RED Parameters
0.3	500	(...)	Disabled
1.45	500	(...)	Disabled
100	500	(...)	Disabled
140	500	(...)	Disabled

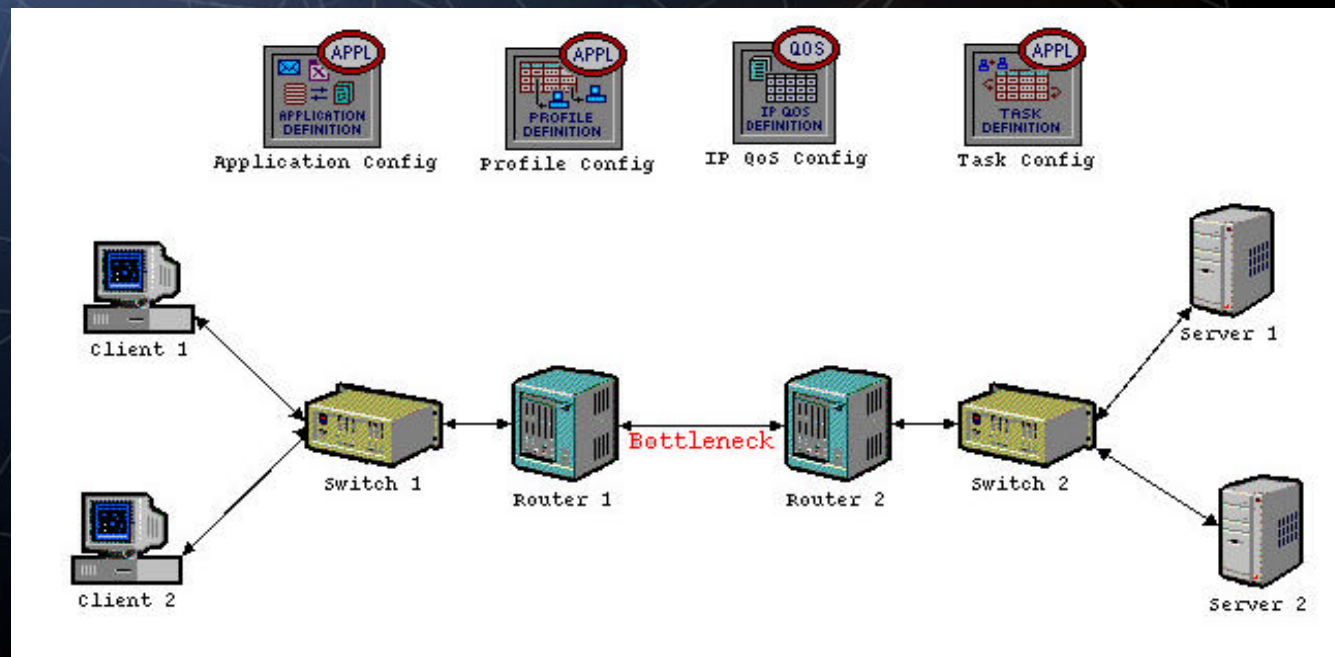
4 Rows

Details Promote Delete Cancel OK



Model verification

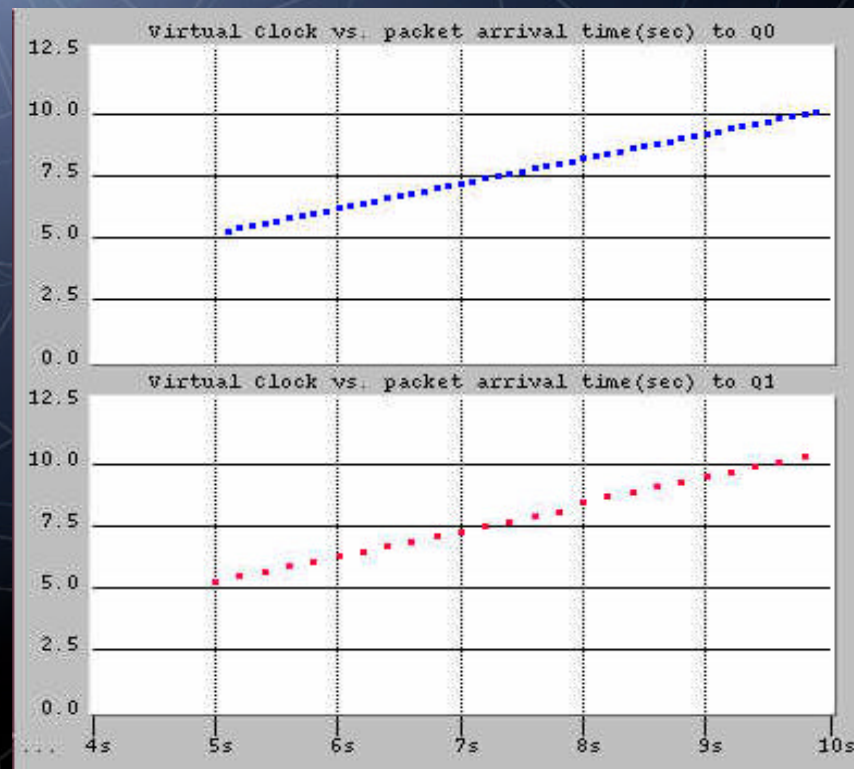
- Client 1: CBR of rate 10 (packets/sec)
- Client 2: CBR of rate 5 (packets/sec)
- Both following their expected packet rate (conforming traffic)



Network model for performance verification of the VirtualClock algorithm.



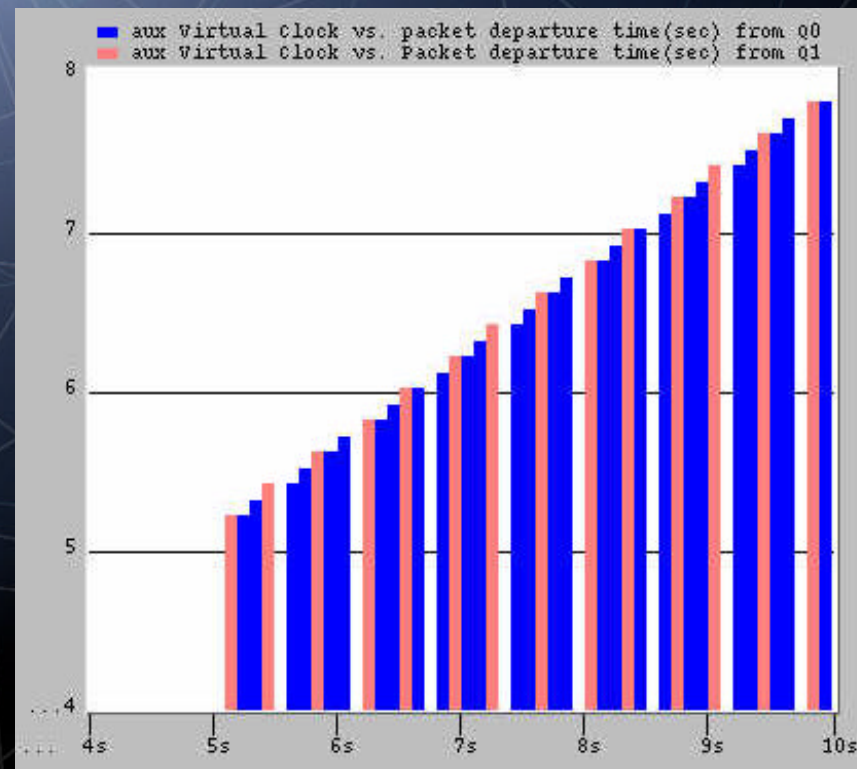
Simulation results



Virtual clock stamps vs. packet arrival time (sec) for queues:
Q0 (top) and Q1 (bottom).



Simulation results



Auxiliary virtual clock vs. packet departure time (sec) for queues Q0 and Q1.

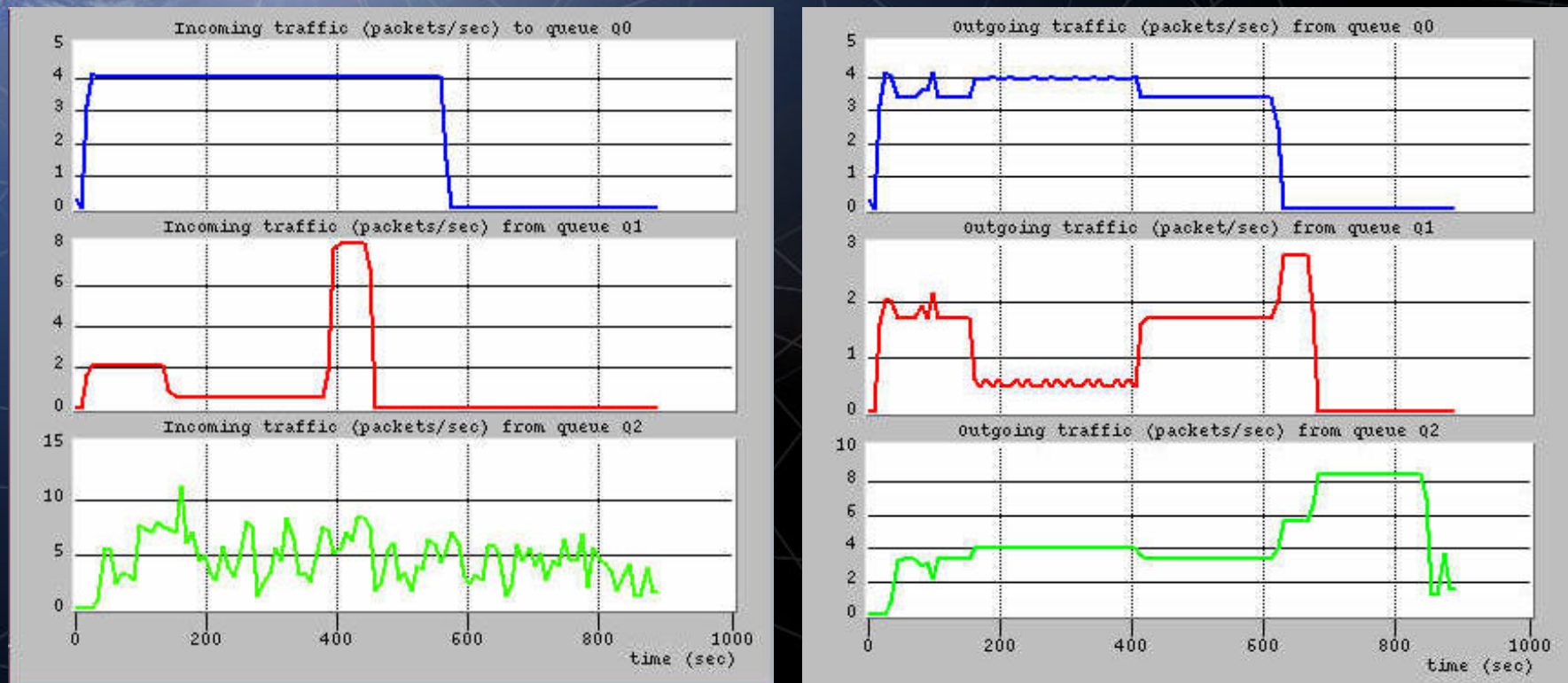


Simulation experiment

- Network model is similar to model used in the verification scenario.
- Sources' expected packet rate:
 - client 1: 4 (packets/sec)
 - client 2: 2 (packets/sec)
 - client 3: 4 (packets/sec).
- Sources' real traffic:
 - client 1 (conforming): CBR of rate 4 (packets/sec)
 - client 2:
 - phase 1 (conforming): CBR of rate 2 (packets/sec)
 - phase 2 (nonconforming): CBR of rate 0.5 (packets/sec)
 - phase 3 (conforming): CBR of rate 8 (packets/sec).
 - client 3 (conforming): self-similar
 - average rate: 4 (packets/sec)
 - Hurst parameter: 0.9
 - fractal onset time scale: 0.1.



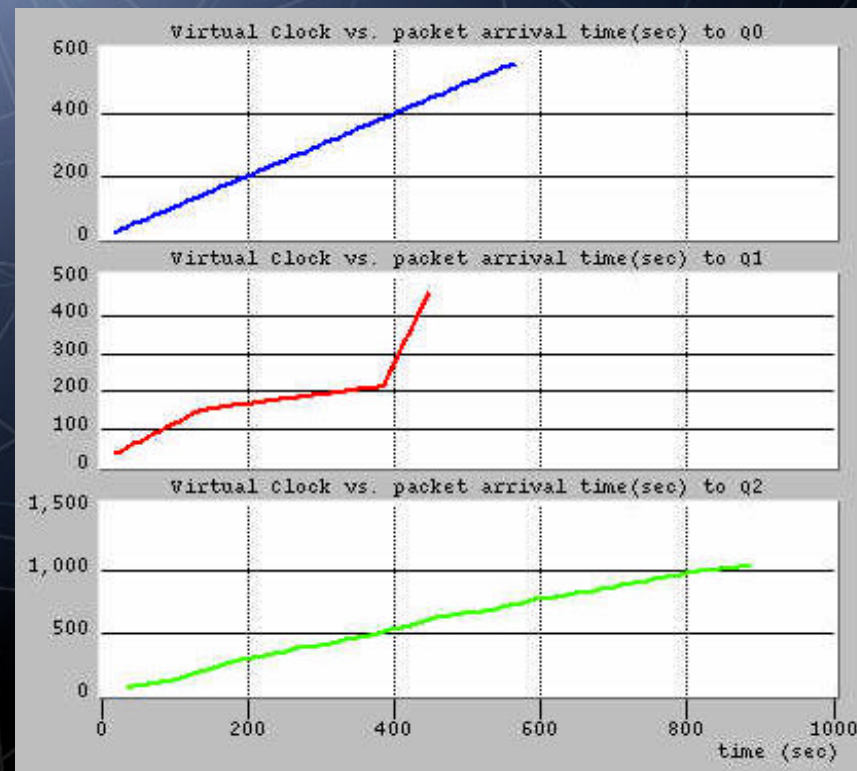
Simulation results



Incoming traffic (left) to and outgoing traffic (right) from Q0 (top), Q1 (middle), and Q2 (bottom).



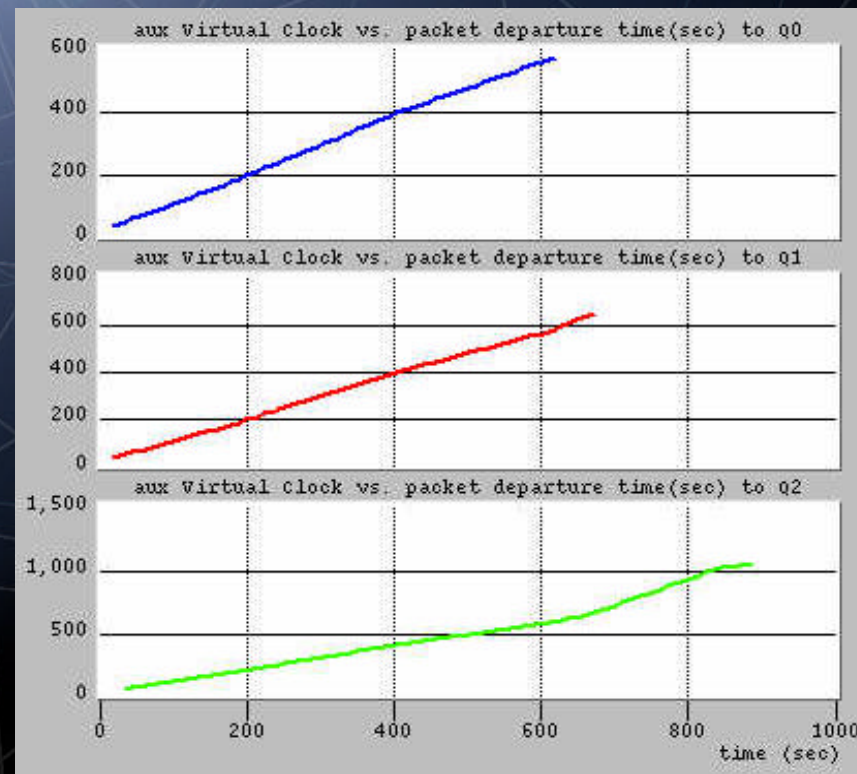
Simulation results



Virtual clock stamps vs. packet arrival time (sec) for queues Q0 (top), Q1 (middle), and Q2 (bottom).



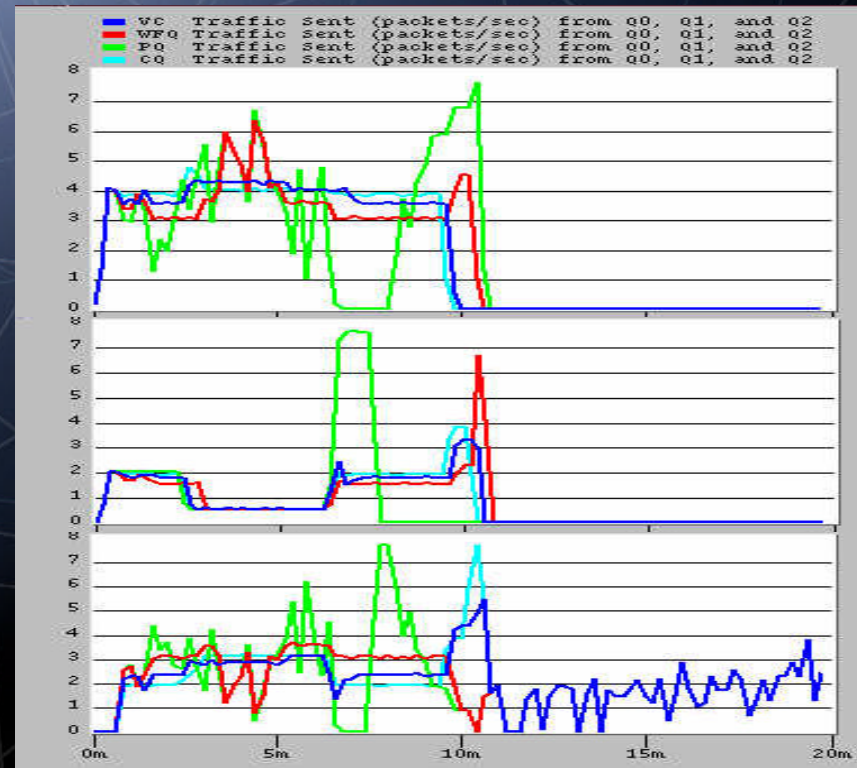
Simulation results



Auxiliary virtual clock stamps vs. packet departure time (sec) for Q0 (top), Q1 (middle), and Q2 (bottom).



Comparison with other scheduling algorithms



Outgoing traffic after being scheduled by scheduling mechanisms: VC, WFQ, PQ, and CQ. Shown are queues Q0 (top), Q1 (middle), and Q2 (bottom).



Conclusions and future work

- We implemented a process model for data forwarding function of VirtualClock algorithm.
- The functionality of the model is verified in a simple network model.
- VirtualClock is used in a complex simulation scenario.
- We used the same scenario with three OPNET algorithms: WFQ, Custom Queuing, Priority Queuing.
- We compared the simulation results with the VirtualClock algorithm.
- Future plans: implementation of the flow monitoring function of the VirtualClock algorithm.



References

- [1] Cisco Systems, Inc. documentation on QoS:
<http://www.cisco.com/warp/public/732/Tech/quality.shtml>.
- [2] S. Keshav, *An Engineering Approach to Computer Networking*. Reading, MA: Addison Wesley, 1998, pp. 209 - 262.
- [3] B. Ryo, "A tutorial on fractal traffic generators in OPNET for Internet simulation," *OPNETWORK2000*, Washington DC, Aug. 2000.
- [4] H. Zhang, "Service disciplines for guaranteed performance service in packet-switching networks," in *Proc. of the IEEE*, vol. 83, no. 10, Oct. 1995.