Simulation of the XUNET Testbed

Lui Chan*

Computer Sciences Department

University of Wisconsin - Madison

August 11, 1992

1. Introduction

The XUNET switch consists of many components that operate independently at different rates. The interfaces between components usually have a double buffer for pipelining. To maximize backplane utilization, there are clever handshakes between different components, so it is hard to estimate switch performance without a detailed simulation. In this paper, we present a fine grained simulation model of the XUNET switch and describe results obtained from experiments with this model. Beside estimating the switch's performance, simulation can also be used to locate bottlenecks under different traffic patterns and configurations, to provide a low-cost environment for testing proposed hardware changes and to evaluate different service disciplines and congestion control mechanisms.

The first section of the paper gives a brief description of the XUNET Switch. The second section describes the simulator. The third section describes the model used to generate traffic. The last section demonstrates the use of the simulator by evaluating alternative priority structures for the switch's *queue module random access memory (QRAM).*

2. XUNET Switch Description

The XUNET 2 Switch [Kalmanek91] is a descendent of Datakit VCS. A switch consists of an *address translation module* and *trunk interface modules* for up to eight transmission lines. Each trunk interface module consists of a *queue module* and a *line card.* The unit of transmission is an *ATM* cell, with

*This work was supported by the National Science Foundation and the Defense Advanced research projects Agency under Cooperative Agreement NCR-8919038 with the Corporation for National Research Initiatives.

5 bytes of header and 48 bytes of data.

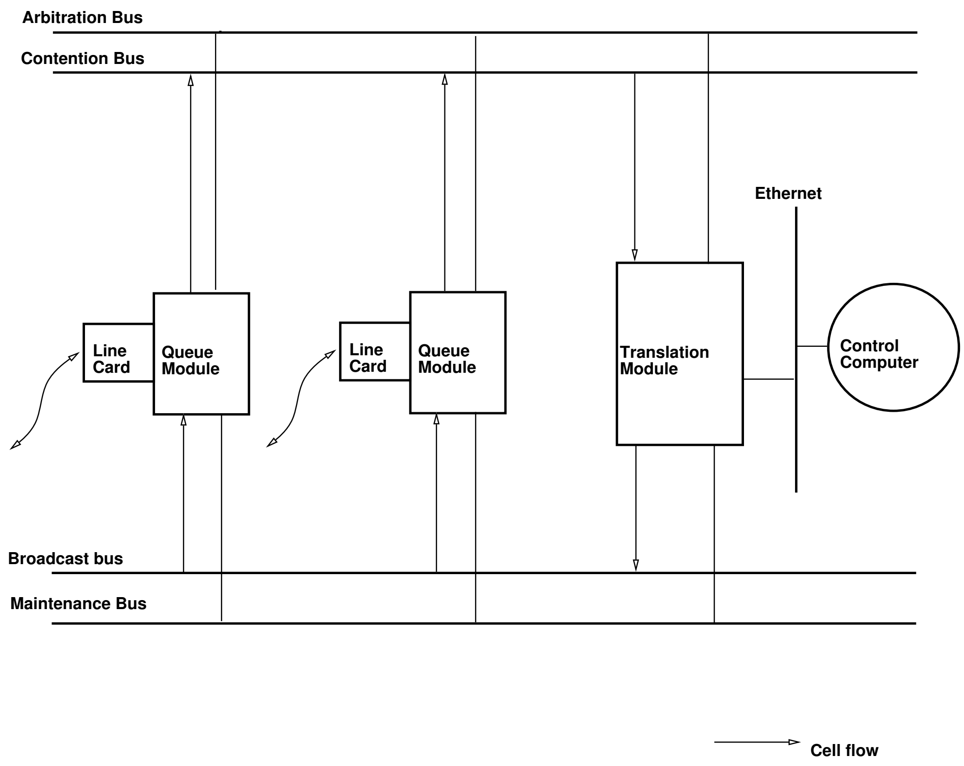

Figure 1. Switch Design

Each switch has four busses. The *contention bus* carries cells that have come into the switch via the trunk interface module to the address translation module. The *broadcast bus* distributes cells to the trunk interface modules for output from the switch. The *arbitration bus* is used by an arbitration protocol to determine which queue module will next have permission to transmit on the contention bus. An arbitration protocol using round robin with two priority levels will be implemented. The *maintenance bus* transfers commands and status reports between the various modules.

The components of the XUNET Switch run at different rates. For example, the queue module can read or write a cell from or to the QRAM every 280 ns while the backplane (contention bus/broadcast bus) can handle a cell every 756 ns. Between two components that run at different rates, there is usually a double buffer for pipelining (Figure 2). We call the buffer that is closer to the sender, *buffer A* , the buffer that is closer to the receiver, *buffer B.* Each buffer can hold a complete ATM cell.

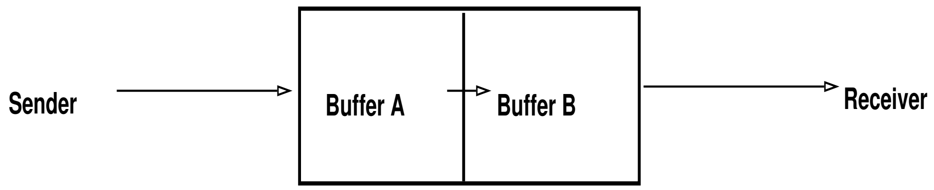

Figure 2. Interface Double Buffer

2.1. Queue Module

The queue module buffers cells in transit between a trunk and the backplane of the XUNET switch. It contains 32 Megabytes of memory (QRAM) which can store up to 512 K ATM cells. Each virtual circuit (VC) has its own queue. The separate queue for each VC provides a firewall between VCs and also allows the use of different service disciplines for different VCs. The service discipline and congestion signaling are controlled by table-driven hardware which can be remotely downloaded over the network. The maximum queue size is 64 K cells.

2.2. Line Card

The line card handles the link level interface to the transmission line.

2.3. Translate Module

The translate module determines the new *virtual circuit identifier (VCI)* to be put into a cell before sending the cell to the broadcast bus. It can hold translation information for up to 512 K full duplex VCs. Cell addresses can be rewritten in 756 nanoseconds.

The translate module can also communicate with the switch's *control computer* over an Ethernet cable. The control computer can change the VCI mapping in the translate module to setup or tear-down VCs.

2.4. Handshakes Between Line Card and Queue Module

The direction from the line card to the queue module is called the *FIO* direction, and the opposite direction is called the *TIO* direction (Figure 3). There is a double buffer between the line card and the queue module in each direction. There are two handshake signals between the queue module and the line card: the *FIOBUSY* signal is lowered by the queue module to grant the line card permission to send another cell to the queue module and the *TIOACK* signal is lowered by the line card to grant the queue module permission to send another cell to the line card.

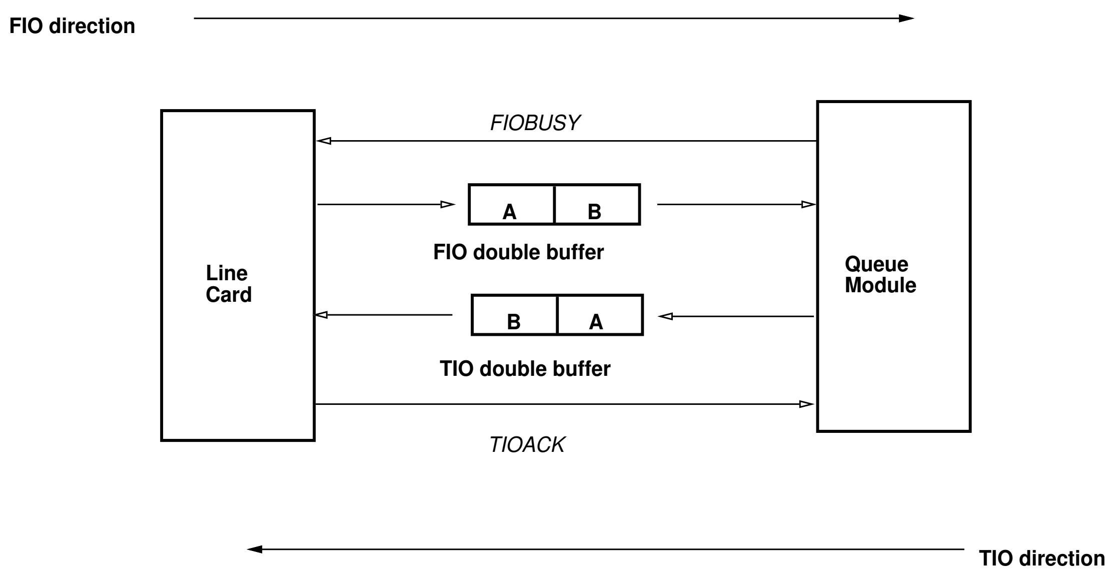

Figure 3. Line Card/Queue Module Interface

In the FIO direction, the line card can deliver a cell to the queue module in two of its cycles. It can start delivering the cell at any of its cycles. The queue module will lower the FIOBUSY signal as long as there is an empty buffer (buffer A or buffer B) in the FIO double buffer. When the signal FIOBUSY from the queue module goes low, the line card will start delivering a cell to the FIO double buffer at the next cycle.

In the TIO direction, when there is an empty slot of the TIO double buffer, the line card will immediately lower the TIOACK signal to inform the queue module that it is ready to receive another cell.

2.5. Handshakes between Queue Module and Contention Bus

The queue module and the contend controller communicate via the *BUSY* flag (Figure 4). When the contend controller is ready to accept a cell from the queue module to contend, it will set the BUSY flag to FALSE. The queue module will send a cell to contend only when the BUSY flag is FALSE. When the queue module decides to send a cell to contend, it sets the BUSY flag to TRUE, to signal the contend controller that it wants to contend for the backplane. The queue module will set the BUSY flag to TRUE before the cell has been strobed into the double buffer, so that it has a better chance to meet the contend deadline.

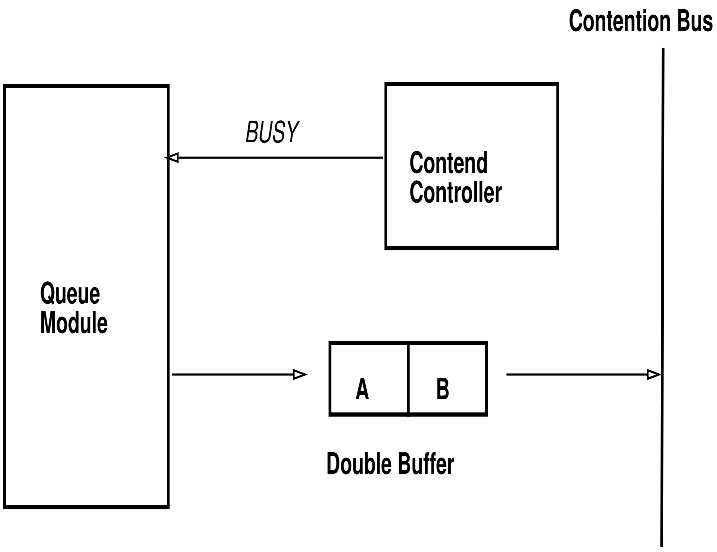

Figure 4. Queue Module/Contention Bus Interface

2.6. QRAM Timing

Each QRAM cycle is divided into 7 ticks, each of 40 ns. There are seven actions in a QRAM: (1) write from line card, (2) read to line card, (3) write from backplane, (4) read to backplane, (5) write from *queue manager*, (6) read to queue manager, and (7) QRAM refresh.

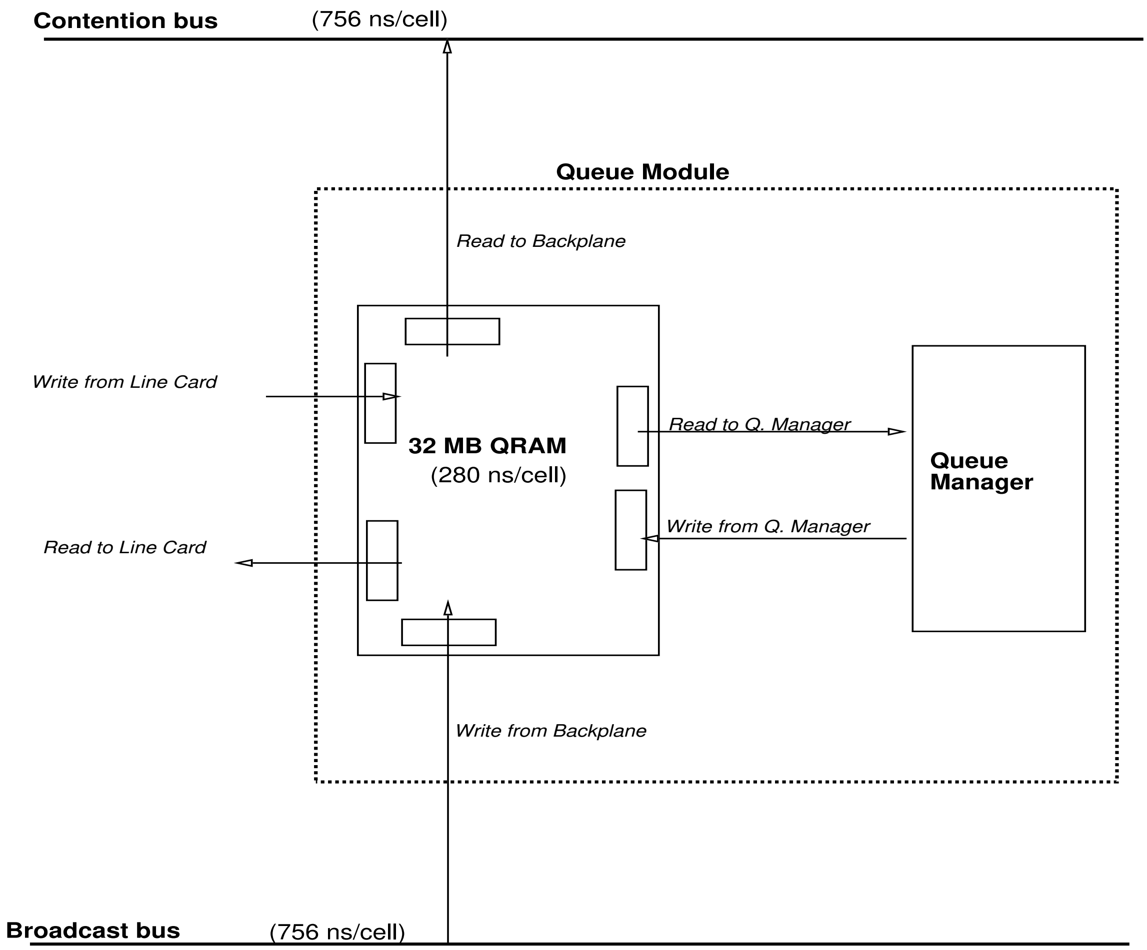

Figure 5. QRAM Actions

Write from backplane (broadcast bus) has the highest priority. QRAM refresh takes place every 33 QRAM cycles and has the second highest priority.

We assume that the decision of which action will be granted QRAM access is made at the start (tick 0) of the current QRAM cycle. If QRAM access is granted, the whole cell will be written into or read out of QRAM at the end of the current cycle. An action will be granted access for the current cycle only if there are no ready competing actions with higher priority. A write is considered ready to compete for the current cycle if the cell to be written was put into the interface double buffer before the start of the current cycle. Read to line card is considered ready to compete if the TIOACK signal from the line card was lowered before the start of the current QRAM cycle.

Read to backplane is slightly different: it is considered ready to compete if the BUSY flag was set to FALSE by the contend controller before the second tick of the previous QRAM cycle (i.e., 5 ticks or 200 ns before the start of the current QRAM cycle.) If it is granted QRAM access, it will set the BUSY flag to TRUE at tick 1 of the current QRAM cycle to inform the *contend controller* that it will contend for the next backplane cycle. It will finish reading the whole cell to the double buffer between the QRAM and the contention bus at the end of the current QRAM cycle.

2.7. Contend Controller

A backplane cycle of 756 ns is divided into 14 ticks. All queue modules that set the BUSY flag to TRUE before the contend deadline (7th tick) in the current backplane cycle are allowed to contend for the next backplane cycle. The winner will start sending a cell to the translate module at the start of the next backplane cycle. At the 8th tick of a backplane cycle, the controller will set the BUSY flag to FALSE if (1) buffer B of the double buffer between the queue module and the contention bus (Figure 4) is empty, or (2) the queue module is putting a cell on the contention bus at this backplane cycle. At the beginning of each backplane cycle, the cell in buffer A will be moved to buffer B if buffer B is empty.

3. Simulator Description

The simulation package consists of two parts: the simulator itself and user utilities.

The simulator is implemented in *DeNet* [Livny89, Livny90], a discrete event simulation language based on Modula-2. The organization of the switch simulator resembles that of a real switch. There are five types of DEVMs (DeNet modules): the *Line DEVM*, *Queue Module DEVM*, *Contend DEVM*, *Translate DEVM* and *Host DEVM*, each simulates a corresponding type of component in a network of switches. A switch (excluding its line cards) is simulated by a collection of Queue Module (Qcard) DEVMs, a Contend DEVM and a Translate DEVM (Figure 6).

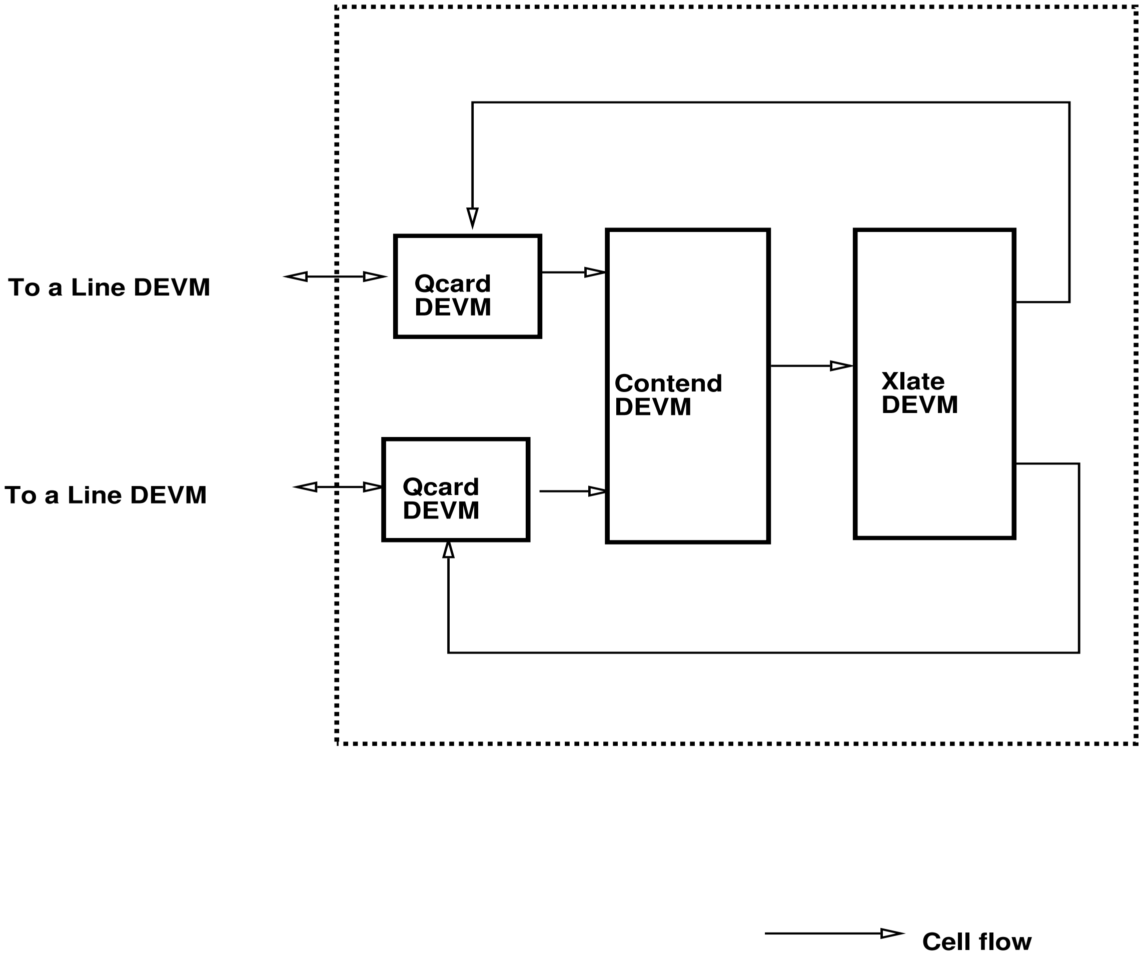

Figure 6. XUNET II Switch Simulator Structure
(Only 2 slots shown)

The sources and sinks of traffic are simulated by Host DEVMs. A network of multiple switches can be simulated by connecting Host DEVMs and DEVMs that simulate the switches with Line DEVMs, as shown in Figure 7.

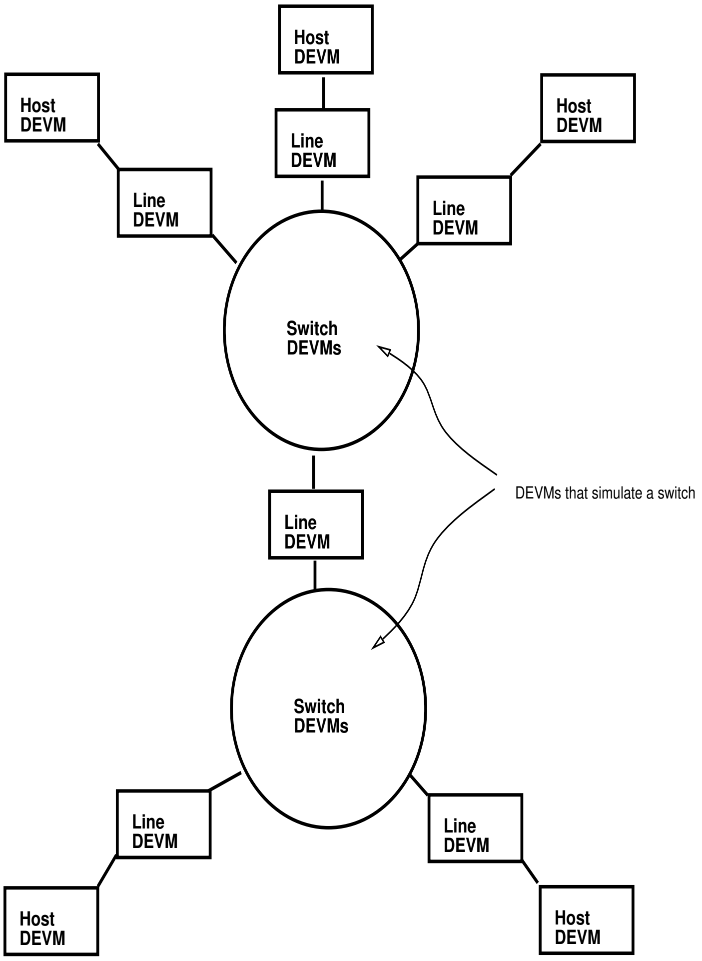

Figure 7. Simulating a Network Consisting of Two Switches and Five Hosts

Some user utilities are provided to help the user set up an experiment, to run it and to gather results of the experiment.

The following subsections give a brief description for each DEVM type.

3.1. Line DEVM

This DEVM type simulates a physical line and the two line cards at the two ends of the line. We assume the design of the Fibre line card developed by Murray Thompson [Thompson91]. There is flow control between the two line cards so that the FIFO at the destination line card will not overflow.

The attributes of a Line DEVM include propagation delay, line speed, FIFO size of the line cards and the flow control type between the two line cards.

We assume that the line speed is 800 million bits per second (Mb/s) or 480 ns/cell.

3.2. Queue Module DEVM

This DEVM type simulates a queue module in a switch. Each VC has its own queue in the queue module. There is a service class associated with each VC and the service disciplines of different service classes are programmed in software of the queue module.

The attributes of a Queue Module DEVM include QRAM cycle time (280 ns), QRAM size , time offsets in a QRAM cycle that the QRAM controller will check and set the BUSY flag, and QRAM priority structure.

3.3. Contend DEVM

This DEVM type simulates the contend controller that carries out the contention algorithm.

The attributes of a Contend DEVM include the backplane cycle time and also the time offsets in a backplane cycle when the contend controller will check the BUSY flag and turn off the BUSY flag.

3.4. Translate DEVM

This DEVM type simulates the translate module in a switch. It provides the VCI mapping. The attributes of this DEVM include queuing time of a cell spent in the translate module. In all the experiments we ran to date, we assume the queuing time in the translate module to be 757 ns.

3.5. Host DEVM

This DEVM type simulates the sources and sinks of traffic.

3.6. The Output of an Experiment

The output of an experiment includes the contention bus utilization of each switch, the throughput, average transit time and average switch resident time of cells for each VC, the number of cells lost, queue size and queuing time for each VC in each queue module, the average queue size and queuing time in each line card.

The throughput of a VC is equal to the amount of data (payload only) received by the destination host divided by the time interval from when the source host sent the first cell to when the destination host received the last cell of the VC. The transit time of a cell is the time interval from when the cell left the line card of the source host to when the cell arrived at the destination host.

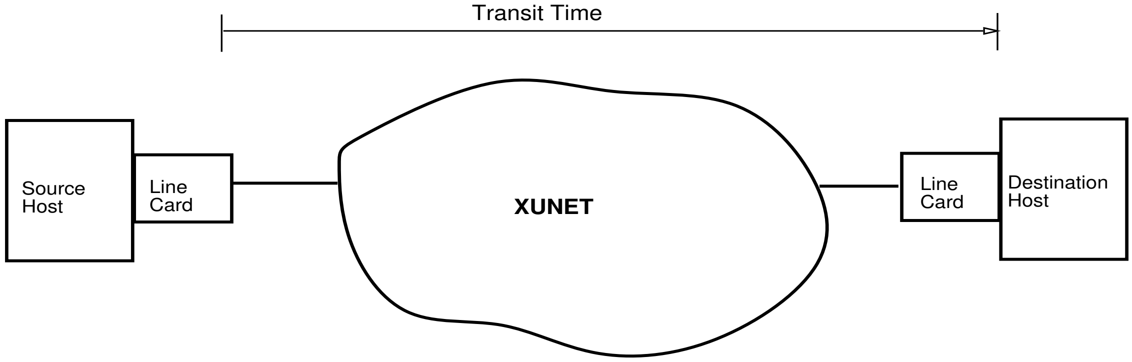

Figure 8. Transit Time of a Cell

The switch resident time is the time interval from when the cell enters the input queue module of a switch to when the cell leaves the output queue module of the switch.

The simulator can also generate traces of the queues in the queue modules for each VC.

4. Traffic Model

In our model, there are two parameters that determine the traffic pattern of a VC: the *inter-burst time* and the *burst size.*

The inter-burst time is the time interval from when a source begins generating a burst to when it begins generating the next burst. The burst size is the size of a burst measured as a multiple of one million bit quantities (Mb).

Both parameters can be fixed or random (exponential,) so we end up with four possible combinations: (1) fixed inter-burst time and burst size, (2) fixed inter-burst time and random burst size, (3) random inter-burst time and fixed burst size, and (4) random inter-burst time and burst size.

Once a burst is generated, the source sends it to the first switch at the maximum line speed.

5. QRAM Actions

There are seven actions in a QRAM: (1) write from line card, (2) read to line card, (3) write from backplane, (4) read to backplane, (5) write from queue manager, (6) read to queue manager, and (7) QRAM refresh.

The assigning of priority to these actions has a major impact on the switch performance, especially with a high speed line card. The performance of different service disciplines, congestion control and flow control mechanisms will all be affected by the priority structure chosen.

We ignore reads to the queue manager and writes from the queue manager. Among the rest of the seven actions, writes from the backplane must be given the highest priority or a cell could be lost. QRAM refresh must be given the second highest priority. How to assign priority to the rest is not clear. Priority among reads is determined by software (service decoder.) Priority between write and read is determined by the hardware.

We tested six Priority Structure options:

Current I	Write from backplane QRAM refresh Write from line card Read to line card Read to backplane	Current II	Write from backplane QRAM refresh Write from line card Read to backplane Read to line card
Proposed I	Write from backplane QRAM refresh Read to line card Read to backplane Write from line card	Proposed II	Write from backplane QRAM refresh Read to backplane Read to line card Write from line card
Alternating I	Write from backplane QRAM refresh Alternate Write From line card and reads Higher read prio. to read to line card	Alternating II	Write from backplane QRAM refresh Alternate Write From line card and reads Higher read prio. to read to backplane

To evaluate the above six structures, we ran simulations with five configurations, two of which involved a single switch (Configurations 1 and 2) the rest involved three switches (Configurations 3, 4 and

5). In all configurations, we assumed that all lines were of 800 Mb/s, the backplane bandwidth was 506 Mb/s, the QRAM cycle time was 280 ns and all VCs have the same contention priority.

5.1. Results with Configuration 1

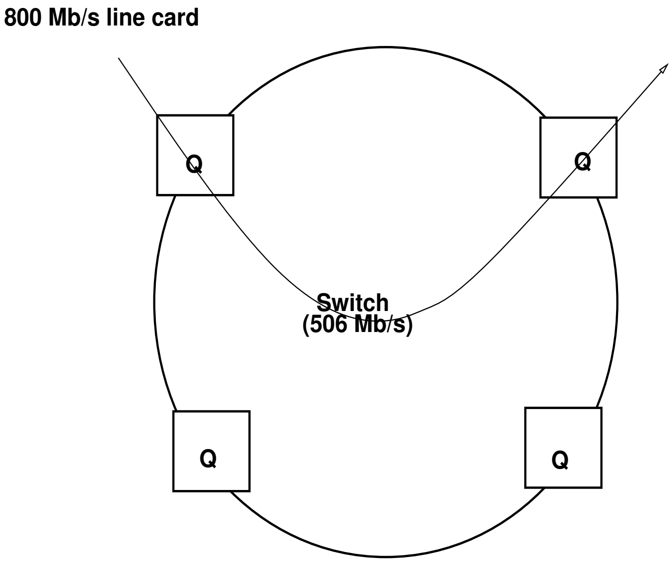

Figure 9. Configuration 1 - 1 Switch, 1 VC

In this configuration, all input traffic consisted of a burst of infinite size.

5.1.1. Performance of Current I and II with Configuration 1

With these two structures, the contention bus utilization was 0.78 and the throughput was 394 Mb/s. The rate of cells being written into the input QRAM from the input line card was 684 Mb/s. The queue at the input queue module grew at a rate of 289 Mb/s (684-394 Mb/s,) and it took about 87 ms to overflow a queue of 64K cells. The output queue was empty most of the time and its maximum size was 2 cells. The growth rate of the FIFO in the input line card was 116 Mb/s. In this case, the contention bus utilization determined the throughput (394 Mb/s = 0.78 X 506 Mb/s)

The reason that the contention bus was not fully utilized was that (1) with a high speed line card, cells could arrive from the line card at a very high rate; (2) since the write from the line card was given higher priority than all reads, the reading of a cell to contend for the backplane could be postponed due to a cell arrival from the line card; and (3) the contend controller could signal the queue module to send another cell (by turning off the BUSY Flag) at most once each backplane cycle.

The following was what we observed: when there was a spare QRAM cycle, no cell might be read to contend because the contend controller had not yet turned off the BUSY flag. After the contend controller turned off the BUSY flag, it was possible that a cell might have just arrived from the line card, which was given higher priority to QRAM cycles, and the read of a cell to contend had to be postponed. When finally the QRAM read a cell to contend, it might have already missed the deadline for the current backplane cycle, and hence wasted a backplane cycle (If other queue modules were also contending, then no backplane cycle would be wasted, but that was not true in this configuration where only one queue module was contending). In conclusion, there were spare QRAM cycles but they could not be put to use because of the asynchronous nature of the QRAM and backplane cycles.

This leads us to consider limiting the line speed to some optimal rate at which the switch could achieve a higher throughput with the current priority structures. Figure 10 shows the graph of contention bus utilization vs. line speed for Configuration 1.

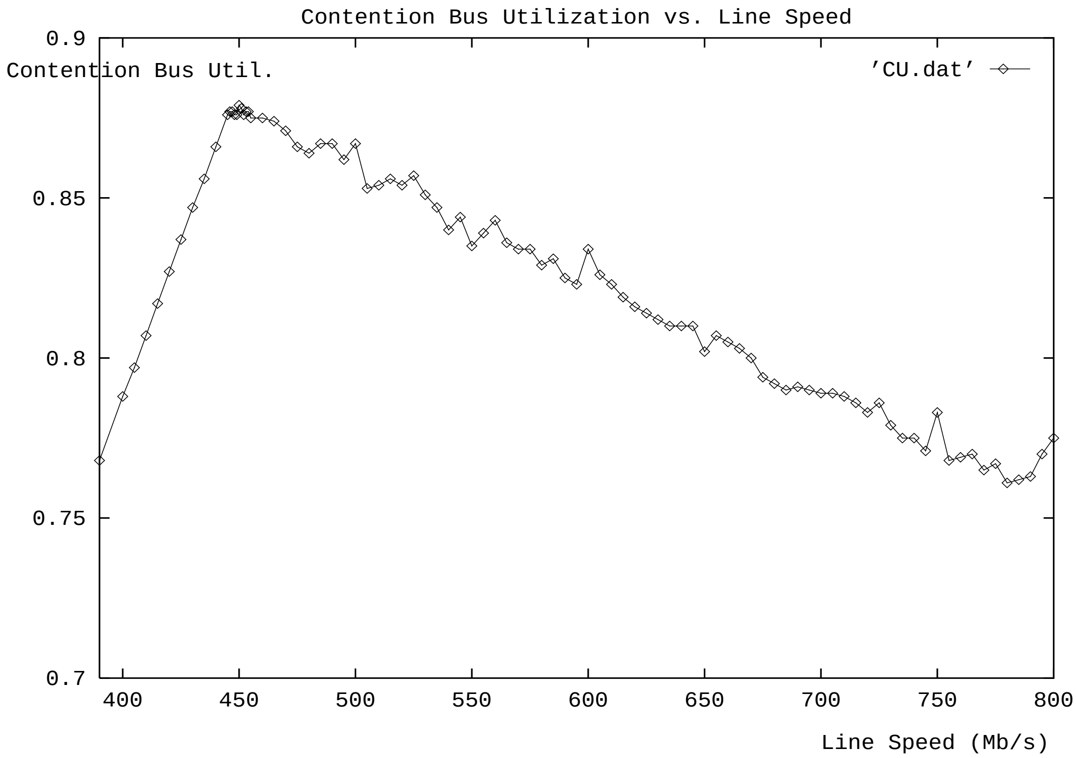

Figure 10. Contention Bus Utilization vs. Line Speed

We observed that the optimal line speed was 450 Mb/s, at which contention bus utilization was 0.879 (equivalent to a throughput of 445 Mb/s). The rate of cells being transferred from the line card to the QRAM was 449.66 Mb/s, the queue growth rate at the input queue module was 3.99 Mb/s and the growth rate of the FIFO in the input line card was 450 - 449.66 = 0.34 Mb/s.

The optimal line speed is lower than we expected. We were expecting it to be around the backplane bandwidth, 506 Mb/s. This is mainly due to the asynchronous interface between the line card and the queue module and between the queue module and the backplane. When we traced the experiments with 506 Mb/s line speed and assumed no QRAM refreshment, we observed the following contention bus utilization pattern: The contention bus could be fully utilized for 154 backplane cycles. For the next 104 backplane cycles, the most frequent pattern was 0111011011, where each digit stands for a backplane cycle, a "0" stands for an idle backplane cycle, a "1" stands for a utilized backplane cycle. Table 1 explains the 0111011011 pattern. Note that times given in bold type indicate the beginning of backplane cycles.

Time(us)	QRAM activities	Contention bus activities
389.60	Busy = TRUE (miss contend deadline)	
389.88		Busy = FALSE (on time)
391.64	Cell 47 arrived from the line card	
392.00	Finish reading cell 44 to the backplane	
393.12		**IDLE (0)**
394.80	Finish writing cell 47 from the line card	
395.20	Busy = TRUE (buffer B already contending)	
397.60	Finish reading cell 45 to the backplane	
399.23	Cell 48 arrived from the line card	
400.68		**SEND cell 44 (1)**
403.20	Finish writing cell 48 from the line card	
405.00		Busy = FALSE (miss read)
406.82	Cell 49 arrived from the line card	
408.24		**SEND cell 45 (1)**
411.60	Finish writing cell 49 from the line card	
412.0	Busy = TRUE (on time to contend)	
412.56		Busy = FALSE (miss read)
414.40	Finish reading cell 46 to the backplane	
414.41	Cell 50 arrived from line card	
415.80		**SEND cell 46 (1)**
420.00	Finish writing cell 50 from the line card	
420.12		Busy = FALSE
420.40	Busy = TRUE (miss contend deadline)	
421.99	Cell 51 arrived from the line card	
422.80	Finish reading cell 47 to the backplane	
423.36		**IDLE (0)**
425.60	Finish writing cell 51 from the line card	
426.00	Busy = TRUE (buffer B already contending)	
428.40	Finish reading cell 48 to the backplane	
429.58	Cell 52 arrived from the line card	
430.92		**SEND cell 47 (1)**
434.00	Finish writing cell 52 from the line card	
435.24		Busy = FALSE (miss read)
437.17	Cell 53 arrived from the line card	
438.48		**SEND cell 48 (1)**
442.4	Finish writing cell 53 from the line card	
442.8	Busy = TRUE (miss contend deadline)	
442.8		Busy = FALSE (on time)
444.76	Cell 54 arrived from the line card	
445.20	Finish reading cell 49 to the backplane	
446.04		**IDLE (0)**
448.00	Finish writing cell 54 from the line card	
448.40	Busy = TRUE (buffer B already contending)	
450.80	Finish reading cell 50 to the backplane	
452.35	Cell 55 arrived from the line card	
453.60		**SEND cell 49 (1)**
456.40	Finish writing cell 55 from the line card	
457.92		Busy = FALSE (miss read)
459.94	Cell 56 arrived from the line card	
461.16		**SEND cell 50 (1)**
464.80	Finish writing cell 56 from the line card	

465.20	Busy = TRUE (miss contend deadline)	
465.48		Busy = FALSE (on time to read)
467.53	Cell 57 arrived from the line card	
467.60	Finish reading cell 51 to the backplane	
468.72		**IDLE (0)**

Table 1.

5.1.2. Performance of Proposed I and II with Configuration 1

These two structures gave the same results with Configuration 1. Contention bus utilization was 0.988 and throughput was 501 Mb/s. The rate of cells written into the QRAM from the line card was 621 Mb/s and the queue at the input queue module grew at a rate of 120 Mb/s. There was no output queue buildup.

We could see that the proposed priority structures offered better performance than the current priority structures. The contention bus was almost fully utilized. The reason for the better performance was that reads had higher priority than writes from the line card; hence the read to the backplane was not affected by cell arrivals from the line card.

When we have to choose between writing a cell from the line card and reading a cell to the backplane, it is better to choose reading to the backplane. If we write the cell from line card first, we might miss a backplane cycle (757 ns.) By giving the reads to the backplane higher priority, we make sure that the backplane will be utilized as much as possible. With the double buffer between the line card and the QRAM, a cell will always be available to be written into the QRAM from the line card when there is a spare QRAM cycle.

There is a cost to the proposed priority structures. By giving write from the line card lower priority, the queue module will accept cells from the line card at a lower rate, which means that the line card will need a larger FIFO to store cells. The growth rate of the queue in the line card under the proposed priority structures is 179 Mb/s, compared to 116 Mb/s under the current priority structures. We assumed that there was link level flow control between adjacent line cards, so the FIFOs in the line cards would not overflow. However, a larger FIFO at the line card has a negative effect on higher priority traffic, since there are no per-VC queues in the line card and all cells will be served in a strict FIFO order in the line card.

5.1.3. Performance of Alternating I and II with Configuration 1

With these two priority structures, contention bus utilization was 0.979 and throughput was 496 Mb/s. The rate of cells written into the QRAM from the line card was 627 Mb/s. The queue at the input queue module grew at a rate of 130 Mb/s. There was no queue buildup at the output queue module.

Although the throughput under this priority structure was slightly less than the throughput under the proposed priority structures, this priority structure did have the advantage of having a slightly smaller queue at the line card. The growth rate of the queue in the line card was 173 Mb/s, compared to 179 Mb/s under the proposed priority structures.

5.2. Results with Configuration 2

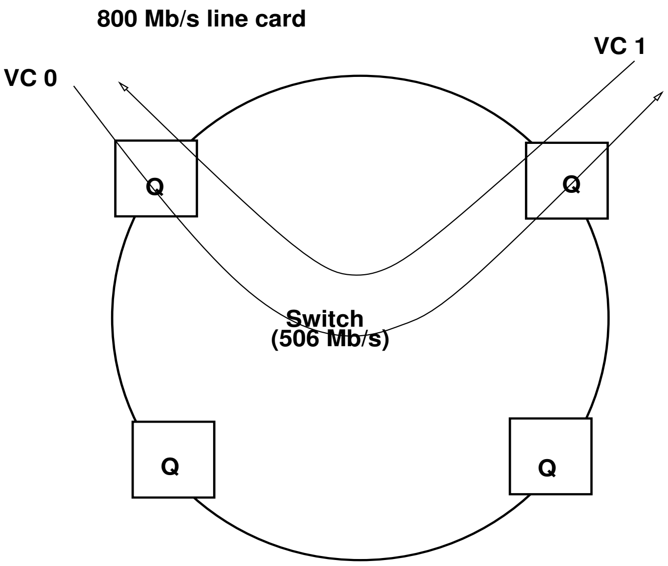

Figure 11. Configuration 2 - 1 Switch, 2 VC

In this configuration, all input traffic had a burst of infinite size. VC 0 started transmitting at time 0 and VC 1 started transmitting 3 backplane cycles latter.

5.2.1. Performance of Current I with Configuration 2

VC 1 was "choked"; only one of its cells got through the backplane. Since only VC 0 was able to contend, the contention bus utilization was the same as the case with Configuration 1 under Current I, 0.78. However, the throughput of VC 0 was 309 Mb/s, less than the 394 Mb/s in Configuration 1 under Current I.

Throughput was lower even though the contention bus utilization was the same as that in Configuration 1 because in Configuration 2 there were cell arrivals from the line cards at both of the queue modules. These cells had higher priority for QRAM cycles than cells in the output queues waiting to be read to the line cards. Hence cells of VC 0 departed the output queue at a slower rate than in Configuration 1 under Current I, and VC 0's output queue started to grow. The rates of cells written into the QRAM from the line cards of VC 0 and VC 1 were 685 Mb/s and 620 Mb/s respectively. The queue growth rates of VC 0 at the input queue module and at the output queue module were 290 Mb/s and 85 Mb/s respectively.

The reason that VC 1 was choked was that there were no spare QRAM cycles for VC 1 to contend at queue module 1. This can be demonstrated by the following calculation:

QRAM bandwidth = 3463K cells/s (QRAM refresh also taken into account)

Arrival rate of VC 0 from the backplane at the output queue module of VC 0 = 394 Mb/s = 1026K cells/s

Arrival rate of VC 1 from the input line card = 620 Mb/s = 1614K cells/s

With Current I, the reads to the backplane were given the lowest priority. There were not even enough QRAM cycles for actions with higher priority than the reads to the backplane. We can calculate the number of QRAM cycles required per second to write cells of VC 0 from the Broadcast Bus (1026K cells), to write cells of VC 1 from the line card (1614K cells) and to read cells of VC 0 to the line card (1026K cells): 1026K + 1614K + 1026K = 3666K cells, which is already larger than the QRAM bandwidth of 3463K cells per second.

Two conditions must be satisfied for choking to occur: (1) continuous cell arrivals from the backplane; and (2) continuous cell arrivals from the line card. So, if cell arrivals of VC 1 from the line card stopped at some time after VC 1 was choked, VC 1 would no longer be choked since there would be spare QRAM cycles for VC 1 to contend and send cells across the backplane. If VC 0 still had cell arrivals from the line card, then VC 0 would be choked till either (1) VC 1 stops sending cells across the backplane; or (2) cells of VC 0 stop arriving from the line card. Hence Current I is not stable.

Choking not only wasted backplane cycles, it also made it harder to support high priority traffic. Higher priority traffic can be choked, and not even its higher contend priority and service priority will help, since the choking is caused by not having any spare QRAM cycles to read cells to contend.

This leads us to consider giving reads to the backplane higher priority than reads to the line card (Current II).

5.2.2. Performance of Current II with Configuration 2

By giving reads to the backplane higher priority than reads to the line card, we solved the problem of choking. The throughput of VC 0 and VC 1 were 166 Mb/s and 170 Mb/s respectively, for a total throughput of 336 Mb/s. The contention bus was fully utilized. The transfer rates of cells of VC 0 and VC 1 from the line cards to the QRAMs were 651 Mb/s and 654 Mb/s respectively. The queue growth rate of each VC at the input queue modules was 399 Mb/s, and that of the queue at the output queue modules was 84 Mb/s. The queue growth rate of the FIFO in the input line cards was about 148 Mb/s.

The following calculation shows that, unlike under Current I, VC 1 now has QRAM cycles to read cells to the backplane. At the time that VC 1 was about to contend for the first time, cells of VC 0 were arriving at queue module 1 from the backplane at a rate of 394 Mb/s or 1026 K cells/s, and we assumed that the arrival rate of VC 1 from the line card at queue module 1 was 684 Mb/s or 1781K cell/s (most pessimistic.) The number of QRAM cycles consumed by activities that had higher priority (writes from the line card and writes from the backplane) than reads to the backplane was 1781K + 1026K = 2807K cells/s, which was less than the QRAM bandwidth.

Even though there was no choking and the backplane was fully utilized, the total throughput of the two VCs was 336 Mb/s, much less than the backplane bandwidth. The reason was that cells accumulated in the output queues due to cell arrivals from the line card, which had higher priority to the QRAM cycles than reads to the line cards at the output queue modules of the VCs.

This leads us to consider giving writes from the line card lower priority than all reads, so that cell arrivals from the line card will not affect cells contending for the backplane or leaving for the line card.

5.2.3. Performance of Proposed I and II with Configuration 2

With these two structures, the contention bus was fully utilized and the throughput of each VC was 253 Mb/s, so the switch was fully utilized. Clearly the proposed priority structures performed better than the current priority structures. However, since we were giving the write from the line card the lowest priority, the rate of cells being transferred from the line card to the QRAM was less than that in the current priority structures. We measured that to be 483 Mb/s. This means a smaller queue at the input queue module, but a larger queue at the input line card. The queue growth rate of each VC at the queue module was 228 Mb/s and the maximum size of the output queue was 2 cells. The growth rate of the FIFO at the input line cards was 800 - 483 = 317 Mb/s.

5.2.4. Performance of Alternating I with Configuration 2

With this structure, VC 1 was choked, just as under structure Current I. The contention bus utilization was 0.979 and the contention bus was used by VC 0 only. The throughput of VC 0 was 415 Mb/s.

The reason that VC 1 was choked was that when a QRAM cycle was available for read, the cycle was used to read a cell of VC 0 to the line card, since read to the line card was given higher priority. By alternating access to QRAM between the writes from the line card and reads, more cycles were available to reads. However, all these extra read cycles were used by the reads to the line card. Reads to the backplane got no benefits from these extra read cycles.

The calculation that shows why VC 1 was choked is as follows. Cells of VC 0 were arriving from the backplane at a rate of 496 Mb/s or 1292K cells/s. This meant that only 3463K - 1292K = 2171K cycles/s were available for the writes from the line card of VC 1 and all reads. Since we alternated access between read and write to the QRAM, this meant that only 2171K/2 = 1086K cycles/s were available for all reads. Since cells were arriving at the output queue at a rate of 1292K cells/s from the backplane and only 1086K cycles were available for all reads (to the backplane and to the line card), and the reads to the line card was given higher priority than the reads to the backplane, all these read cycles were used to read cells of VC 0 to the line card.

The only improvement that this structure had over the current priority structures was that it resulted in a higher throughput. This occurred because by alternating access to the QRAM between reads and writes, cells of VC 0 were guaranteed a greater portion of the QRAM cycles to read to the line card at its output queue module.

5.2.5. Performance of Alternating II with Configuration 2

With this structure, the performance was the same as under the proposed structures, i.e., the switch was fully utilized. The rate of cells being transferred from the line card into the QRAM was 537 Mb/s, the queue growth rate at the input queue module was 283 Mb/s and the growth rate of the FIFO in the line card was 263 Mb/s.

There is an advantage of this structure over the proposed priority structures: a smaller queue at the line card. The growth rate of the FIFO in the input line card had been cut down from 317 Mb/s to 263 Mb/s. This would help high priority traffic.

5.3. Results with Configurations of Three Switches

In the next three sub-sections, we evaluate the current, proposed and alternating priority structures in configurations involving multiple switches.

The configurations are described in Figures 12, 13 and 14. Each rectangle represents a host and each circle represents a switch. The numbers in them are the host/switch IDs. The number on a line between a host and a switch or between two switches is the propagation delay between them.

In all three of the multiple switches configurations, both VCs started sending at time 0, stopped sending 1000 ms later, and both of them have fixed inter-burst time and a fixed burst size of 2 Mb. For each configuration, we choose the minimum inter-burst time (hence maximum input rate) without losing any cells in the queue modules under the current, proposed and alternating priority structures. By experiments, we found that as the inter-burst time decreased (i.e., input traffic rate increased), the current priority structure was the first one to lose cells, If both VCs were going at the same direction, i.e., both VCs entered the switch via the same queue module, then under the current priority structure, cells would start to be lost if each VC had an inter-burst time less than 9.22 ms (> 217 Mb/s input rate). If the two VCs were going at the opposite direction, i.e., cross traffic, then under the current priority structure, cells would start to be lost if each VC had an inter-burst time less than 8.55 ms (> 234 Mb/s input rate). This is the reason that we choose 9.22 ms as the inter-burst time for Configuration 3 and 8.55 ms for Configuration 4 and 5.

The maximum queue size per VC at each queue module is 64K cells. We choose the value 30K cells, which is large enough for the three configurations not to lose any cells.

The result for each configuration is summarized in the form of a table. In the table, for each VC, "Transit time (std.)" stands for the average transit time, with its standard deviation in parenthesis. "InQ, S1 (Max)" stands for the average input queue size at the input queue module of switch 1, with the maximum queue size in the parenthesis. "OutQ, S1 (Max)" stands for the average output queue size at switch 1, with the maximum queue size in parenthesis. "LC Q, S1, (Max)" stands for the average line card FIFO size at switch 1, with the maximum queue size in parenthesis.

5.3.1. Results with Configuration 3

In this configuration, both VCs have a fixed inter-burst time of 9.22 ms and a fixed burst size of 2 Mb. This amounts to a fixed average rate of 217 Mb/s for each VC. Both VCs were going in the same direction.

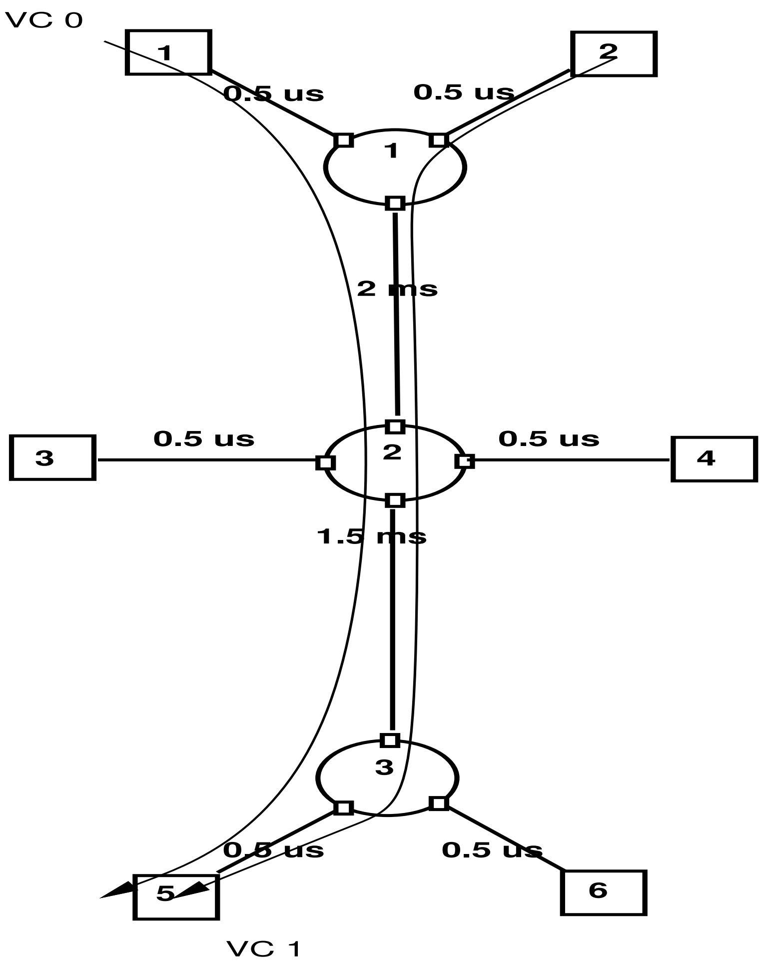

Figure 12. Configuration 3 - 3 Switches, 2 VCs

The results of Configuration 3 are summarized in Table 2. All output queues are empty most of the time.

	VC #	Current I & II	Proposed I & II	Alternating I& II
Transit time (std.) (ms)	0 1	9.41 (2.43) 9.41 (2.43)	6.26 (1.58) 6.26 (1.58)	6.65 (1.8) 6.65 (1.8)
Throughput (Mb/s)	0 1	215.17 215.17	216.49 216.49	216.34 216.34
InQ, S1, (Max) (cells)	0 1	1280 (3283) 1280 (3282)	1245 (3192) 1245 (3191)	1250 (3190) 1250 (3190)
InQ, S2, (Max) (cells)	0 1	1630 (3615) 1630 (3614)	23.7 (61) 23.7 (61)	223 (542) 223 (542)
InQ, S3, (Max) (cells)	0 1	19.8 (488) 19.8 (488)	1.07 (4) 1.07 (4)	0.752 (20) 0.751 (20)
LC Q, S1, (Max) (cells)	0 1	108 (745) 108 (745)	144 (966) 144 (966)	144 (966) 144 (966)
LC Q, S2, (Max) (cells)		0.575 (2)	0.592 (2)	0.577 (2)
LC Q, S3, (Max) (cells)		0.571 (2)	0.586 (2)	0.574 (2)

Table 2.

In this configuration, the proposed priority structures offered the best performance in terms of average transit time and standard deviation of transit time, followed by the alternating priority structures. The proposed priority structures required smaller queues at input queue modules than the current priority structures, especially at switch 2, at which it reduced the average queue size per VC from 1630 cells to 23.7 cells and reduced the maximum queue size per VC from 3615 cells to 61 cells. It also reduced the queue size at switches 1 and 3. The cost of this improvement is the increase in the line card FIFO size at switch 1 from 108 cells to 144 cells. However, this cost is small compared to the improvement.

The alternating priority structures seem to offer no advantage over the proposed priority structures in this configuration: its throughput and transit time were not as good as the proposed priority structure and it did not have significantly smaller FIFOs at line cards.

5.3.2. Results with Configuration 4

In Configuration 4, both VCs had a fixed inter-burst time of 8.55 ms and a fixed burst size of 2 Mb. This amounted to a fixed average rate of 234 Mb/s for each VC. The two VCs were going in opposite directions.

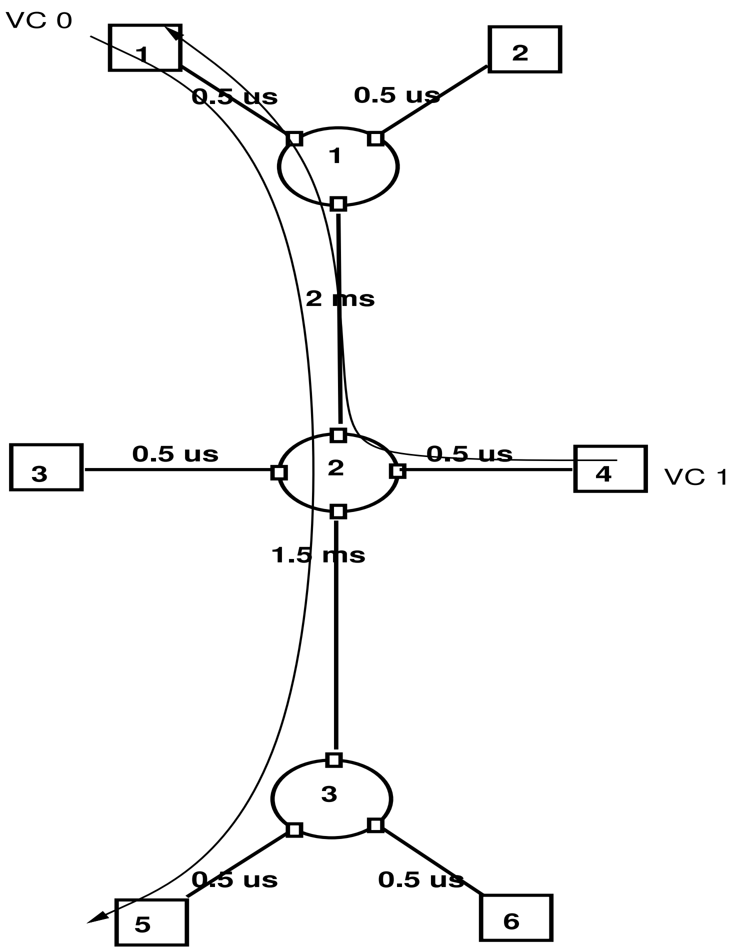

Figure 13. Configuration 4 - 3 Switches, 2 VCs

The results of Configuration 4 are summarized in Table 3.

	VC #	Current I	Current II	Proposed I & II	Alternating I	Alternating II
Transit time (std.) (ms)	0 1	10.2 (2.12) 5.77 (2.35)	6.01 (1.3) 4.52 (1.27)	5.72 (1.03) 4.22 (1.03)	6.66 (0.494) 4.97 (0.771)	5.76 (1.07) 4.26 (1.07)
Throughput (Mb/s)	0 1	232.83 233.36	233.53 233.88	233.78 234.13	233.65 234.04	233.74 234.09
InQ, S1, (Max) (cells)	0 1	2900 (6725) 747 (3624)	987 (2717) 269 (740)	448 (1570) 533 (1300)	293 (1740) 1120 (2800)	492 (1730) 551 (1280)
OutQ, S1, (Max) (cells)	0 1	46.1 (1032) 246 (2163)	0 (4) 17.1 (233)	0 (2-3) 0 (2)	40.3 (458) 0 (2)	0 (5) 0 (2)
InQ, S2, (Max) (cells)	0 1	621 (3191) 976 (3242)	270 (745) 998 (2757)	533 (1300) 490 (1835)	1160 (2800) 240 (1140)	551 (1280) 517 (1890)
OutQ, S2, (Max) (cells)	0 1	0 (2) 10.5 (280)	0 (2) 0 (4)	0 (2) 0 (2-3)	0 (2) 52 (458)	0 (2) 0 (5)
InQ, S3, (Max) (cells)	0 1	1.45 (112) -	2.91 (84) -	0.598 (7) -	5.83 (73) -	0 (10) -
OutQ, S3, (Max) (cells)	0 1	0 (2) -	0 (2) -	0 (2) -	0 (2) -	0 (2) -
LC Q, S1, (Max) (cells)	0 1	173 (1170) 0.751 (77)	131 (842) 0.309 (3)	236 (1360) 11.7 (103)	195 (1240) 43.4 (431)	210 (1250) 0 (3)
LC Q, S2, (Max) (cells)	0 1	0.897 (318) 117 (745)	0 (3) 117 (745)	11.7 (103) 194 (1140)	58.9 (431) 189 (1140)	0 (3) 185 (1100)
LC Q, S3, (Max) (cells)	0 1	0 (2) -	0 (2) -	0 (3) -	0 (2) -	0 (2) -

Table 3.

Choking could happen with Current I and Alternating I, which explained the large difference in transit time between the two VCs under the two priority structures. With the other priority structures, choking did not occur and the transit time of VC 0 is about 1.5 ms (the propagation delay between switch 2 and 3, the extra "miles" that VC 0 had to go) larger than VC 1's.

As in Configuration 3, the proposed priority structures offered the best performance in terms of average transit time and standard deviation of transit time, followed by the alternating priority structures.

One advantage of the proposed and Alternating II priority structures is that they require a very small output queue at queue modules, which make the allocation of output queue size trivial. We can see that the maximum output queue size is 3 cells under the proposed priority structures, 5 cells under Alternating II.

In this configuration, the alternating priority structures did have some advantages over the proposed priority structures: they reduced the average input line card FIFO size of the input switch of VC 0 from 236 cells to 210 cells, and that of VC 1 from 189 cells to 180 cells. Their throughput and transit time are very close to those offered by the proposed priority structures.

5.3.3. Results with Configuration 5

Configuration 5 has the same traffic input as Configuration 4: both VCs have a fixed inter-burst time of 8.55 ms and a fixed burst size of 2 Mb. This amounts to a fixed average rate of 234 Mb/s for each VC. The two VCs were going in opposite directions.

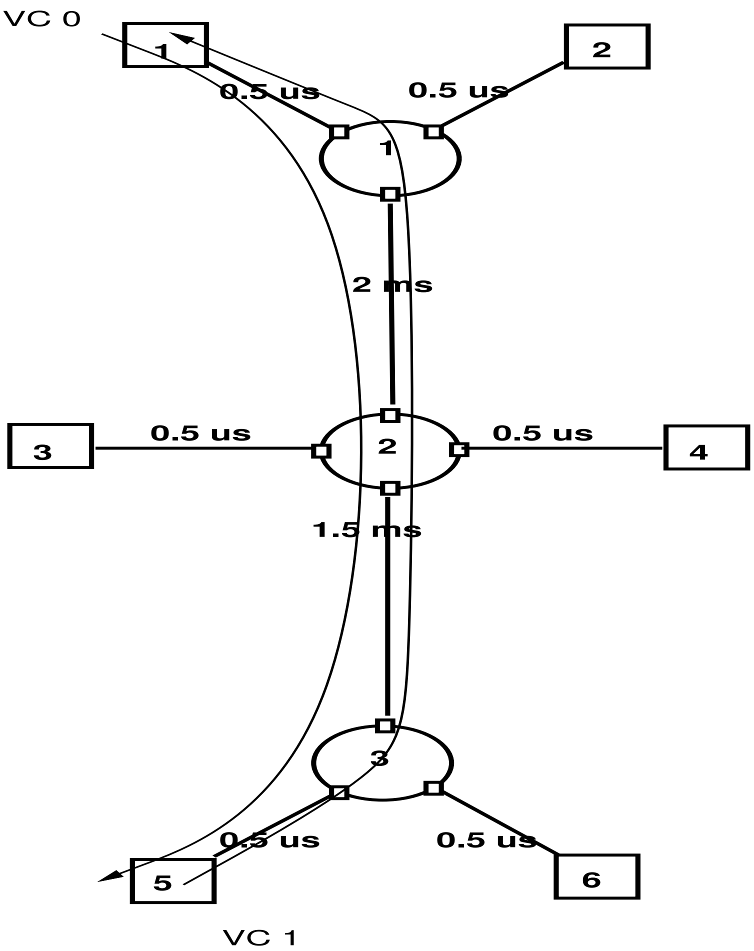

Figure 14. Configuration 5 - 3 Switches, 2 VCs

The results of Configuration 5 are summarized in Table 4.

	VC #	Current I	Current II	Proposed I & II	Alternating I	Alternating II
Transit time (std.)	0	17.5 (4.69)	6.44 (1.82)	6.22 (1.55)	6.38 (1.63)	6.22 (1.55)
(ms)	1	13.9 (3.76)	6.44 (1.82)	6.21 (1.55)	6.37 (1.67)	6.21 (1.55)
Throughput	0	229.60	233.37	233.38	233.35	233.38
(Mb/s)	1	230.67	233.32	233.32	233.28	233.32
InQ, S1, (Max)	0	6520 (15179)	1180 (3165)	861 (2182)	987 (3060)	968 (2430)
(cells)	1	353 (6296)	1.06 (181)	0.7-0.8 (324)	21.2 (2110)	0.8 (312)
OutQ, S1, (Max)	0	16.7 (1614)	0 (5)	0 (2-3)	0 (118)	0 (5)
(cells)	1	97.7 (2215)	115 (654)	0 (2)	0 (53)	0 (3)
InQ, S2, (Max)	0	561 (5947)	180 (1460)	204 (2268)	244 (4700)	201 (2220)
(cells)	1	509 (5672)	180 (1282)	205 (1945)	157 (1850)	201 (1910)
OutQ, S2, (Max)	0	10.2 (1097)	0 (4)	0 (2)	0 (172)	0 (3)
(cells)	1	12.8 (982)	0 (4)	0 (2)	1.64 (688)	0 (4)
InQ, S3, (Max)	0	446 (5212)	4.99 (252)	4.86 (312)	36.4 (3420)	4.99 (318)
(cells)	1	4670 (10754)	1180 (3160)	862 (2182)	1070 (4500)	969 (2430)
OutQ, S3, (Max)	0	97.3 (1969)	116 (663)	0 (2)	2.16 (623)	0 (3)
(cells)	1	22.8 (1102)	0 (5)	0 (2)	1.32 (590)	0 (5)
LC Q, S1, (Max)	0	156 (1140)	155 (974)	433 (2060)	328 (1830)	331 (1720)
(cells)	1	0.661 (247)	0.308 (2)	0.315 (26)	0.372 (106)	0.308 (2)
LC Q, S2, (Max)	0	0.62 (176)	0.31 (3)	1.8 (169)	2.12 (721)	0.316 (3)
(cells)	1	1.02 (274)	0.309 (2)	1.7 (142)	0.493 (157)	0.308 (2)
LC Q, S3, (Max)	0	0.717 (269)	0.308 (2)	0.485 (22)	0.643 (243)	0.308 (2)
(cells)	1	154 (1170)	154 (969)	433 (2070)	340 (2510)	331 (1720)

Table 4.

In this configuration, the proposed priority structures and Alternating II offered the best performance in term of average transit time and standard deviation of transit time. Compared to the proposed priority structures, Alternating II has the same transit time and throughput , but it reduced the average line card FIFO size at switch 1 and 3 from 433 cells to 331 cells, and the maximum line card FIFO size from around 2060 cells to 1720 cells. So Alternating II seem to be the best choice for this configuration.

The large difference in transit time of the two VCs under Current I is due to choking of VC 0 at switch 2. The propagation delay between switch 2 and 3 is 1.5 ms, smaller than the propagation delay of 2 ms between switch 1 and 2. Hence the bursts of VC 1 arrived at switch 2 earlier than those of VC 0 did and gained control of the backplane of switch 2.

As in Configuration 4, the maximum output queue size at queue modules is 3 cells under the proposed priority structures and 5 cells under Alternating II.

5.3.4. Results with a Random Traffic Pattern

When both VCs in Configurations 3, 4 and 5 were changed to have a random pattern with the same parameter values as mean values, cells were lost due to queue overflow at the queue modules (i.e., the queues at the queue modules grew faster than the single FIFO at the line cards).

With a random traffic pattern, the proposed priority structures not only offered the best performance in terms of transit time and standard deviation of transit time, but also in terms of throughput. This was because under the proposed priority structures, a lower cell loss rate occurred.

6. Conclusion

With a detailed simulation of the XUNET switch, we were able to evaluate different switch design options and run experiments that currently cannot be run on existing hardware. In particular, we evaluated different QRAM priority structures in configurations involving high speed line cards (which are not yet available).

Current I has the problem of choking when there is cross traffic between two trunks in a switch. Current II does not have this problem. Current II has much better performance than Current I when there is cross traffic. The advantage of the current priority structures is that they have the smallest queues at the input line cards than under any other priority structures. This is because, under this structure, the queue modules are very aggressive at accepting cells from the line card. However, for the same reason, the queue modules may miss the deadline of a contention cycle or delay the delivery of cells to the line card, hence the current priority structures, especially Current I, also give the worst performance (high transit time, low throughput).

Under the proposed priority structures, the choice of which reads (the reads to backplane or the reads to the line card) are given priority has little effect on performance (throughput and transit time) and queue size. The performance of the proposed structures are much better than the current priority structures under configurations where the bulk of the traffic comes to the switch via a single line. Their performance is slightly better than Current II in the configurations where there is cross traffic.

The advantages of the proposed priority structures are: (1) good performance: low transit time and high throughput, (2) smaller queue size at the input queue modules, (3) fewer cells lost in the queue modules, (4) small maximum output queue size (3 cells), which makes the allocation for output queues trivial. The disadvantage is that under these structures, there will be more cells stored in the receiver buffer at the input line card. And since there are no per-VC queues, higher priority traffic may suffer.

The alternating structures are compromises between the current priority structures and the proposed priority structures. Alternating I has the problem of choking in the configuration with cross traffic, while Alternating II does not. Alternating II's performance is comparable to the proposed structures, and it requires a somewhat smaller queue at the input line card.

In conclusion, Proposed I and II consistently offered the best performance, but they also need larger queues at the line cards than the other structures. Current I offers the worst performance. Since Alternating I and Current I may not be stable in cross traffic due to choking. the reasonable options to consider are Current II, Proposed I and II, and Alternating II. In traffic configurations where the bulk of traffic comes to a switch via more than one trunk, the above four options offer comparable performance. In configurations where the bulk of traffic comes to a switch via a single trunk, Current II can perform much worse than the other three options. However, Current II does have the advantage of requiring a smaller FIFO at the line cards. Alternating II's performance is slightly worse than Proposed I and II, but it requires a smaller FIFO at the line cards than the proposed priority structures.

Acknowledgements

The proposed structures were suggested by Murray Thompson and the alternating priority structures were suggested by Charles Kalmanek. I would like to thank Lawrence Landweber for his numerous suggestions and for reviewing the report.

References

[Kalmanek91] Charles R. Kalmanek. *Xunet 2: A Nationwide Testbed in High-Speed Networking.* March, 1991

[Livny89] Livny, Miron. *DeNet Overview.* November, 1989.

[Livny90] Livny, Miron. *DeNet User's Guide.* February, 1990.

[Thompson91] Murray Thompson. *New "55 Byte" Fiber Link Line Card.* December, 1991