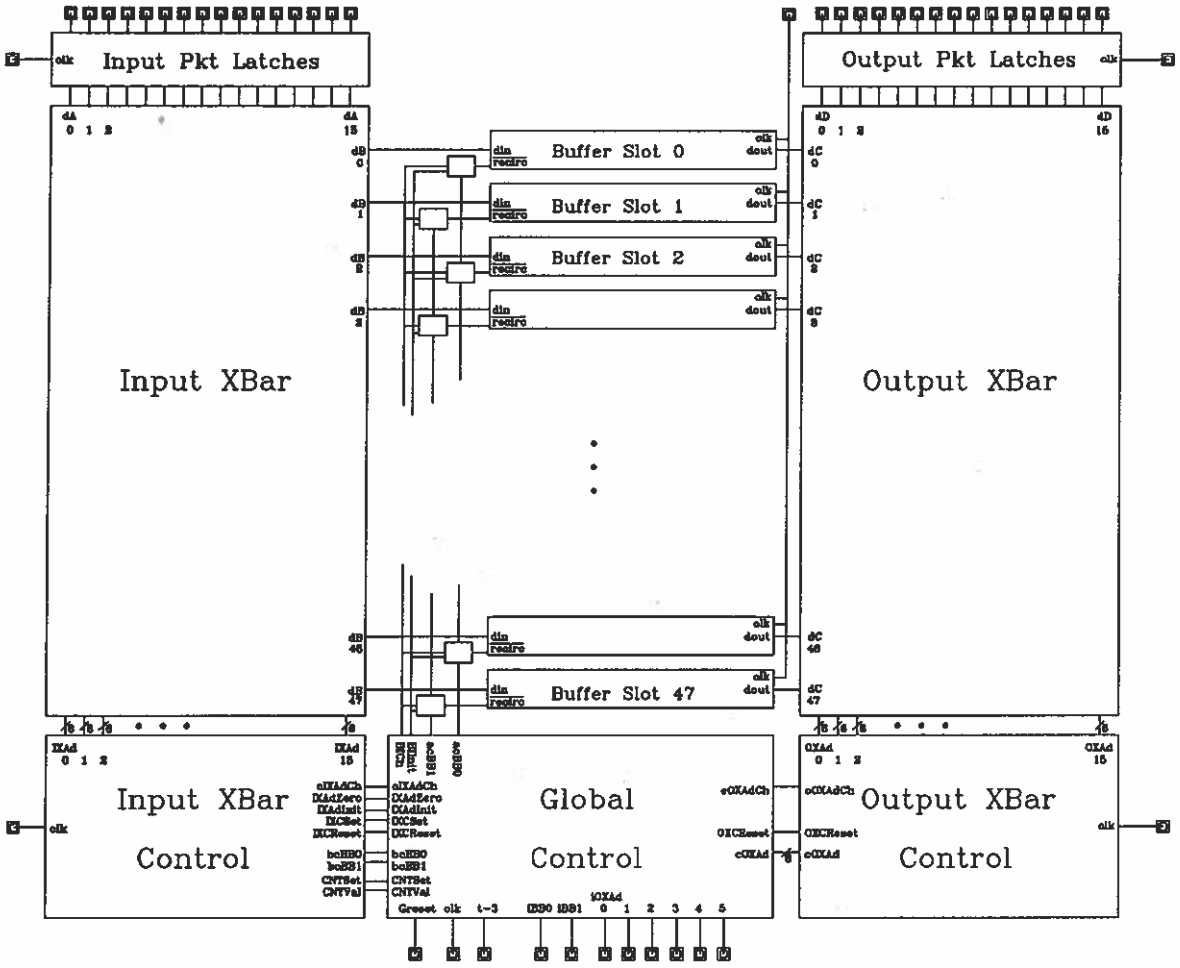
Switching Systems for Gigabit Networks

Progress Report, June 1992, NCR-8914396



Computer and Communications Research Center
Washington University, St. Louis

Overview

This research centers on the design and analysis of switching systems capable of supporting gigabit networks with general multicast communication capabilities. The activity centers on the design of such a system using a modified version of the broadcast packet switch technology, previously developed at Washington University. In addition, however, we are studying alternative architectures, with a particular interest in adapting them to better support multicast. The major research accomplishments of the past year are briefly summarized below.

- We have designed a way of providing 2.4 Gbps ports using a switch fabric which has an internal data rate of 800 Mb/s using a mechanism called port sharing. This takes advantage of the cell resequencing mechanism described in our last report.
- We are making continuing progress in the design of a pair integrated circuits to implement a 16 port switch element with shared buffering that can support external link speeds of 600 Mb/s. The data slice will be fabricated this summer and the control chip is nearing completion.
- We have carried out a quantitative comparison of many different ATM switch architectures. The methodology developed for this comparison carefully separates architectural issues from implementation issues and provides a uniform comparison that clearly demonstrates the advantages and disadvantages of different approaches. Surprisingly, we have found cost differences of as much as two orders of magnitude among various architectures.
- We have continued to develop and evaluate the fast buffer reservation mechanism for handling bursty traffic. In particular, we have carried out a detailed simulation study to compare the link efficiencies achievable, relative to statistical multiplexing. Particularly in the case of heterogeneous traffic, fast buffer reservation offers significant advantages. We have also found and corrected an error in the original call acceptance algorithm. While the new algorithm is somewhat more complex than the original, it still allows call acceptance decisions to be made in sub-millisecond times.
- The queueing performance of buffered multistage interconnection networks has attracted a lot of interest from various researchers recently. This year, we refined our queueing model for networks constructed from shared buffer switch elements, making it far more accurate than previously.

Design of a Second Generation Broadcast Packet Switch

In our last progress report, we described a series of refinements we have made to the broadcast packet switch architecture. During the last year, we have continued to make progress on the design of two integrated circuits that implement a 16 port shared buffer switch element, which forms the core of the revised architecture. The design of the data slice for the switch element is nearly complete and is described in detail in [1]. We will have this chip fabricated in 1.2 μm CMOS this summer. The control chip has turned out to be the major challenge. As described in the last progress report, we developed a control design using a novel arbitration array which allows stored cells to contend for multiple outputs simultaneously in a fair and efficient way. (The multiple contention is required of course for multicast communication.) The key challenge has been to manage the timing of events so that the circuit can operate within the time constraints imposed by the short ATM cell size. The major components have now all been completed and simulated at 100 MHz. What remains is to integrate the various components together.

One key challenge for high speed networks is to provide economical support for gigabit transmission links in a context where many users require only lower speed access. If gigabit networks are to be broadly successful, we must handle a wide range of link speeds within networks and individual switching systems. Figure 1 shows the design of a port controller that can be used in conjunction with a second generation broadcast packet switch fabric to provide 2.4 Gb/s external links through a technique called *port sharing*. The port controller consists of three chips, a Link Interface Chip (LI), a Virtual Circuit/Port Translation Chip (VXT) and a Transmit Buffer Chip (XMB). The core of the port controller employs 32 bit wide data paths and a clock rate of 100 MHz, which is sufficient to support an external link at 2.4 Gb/s. On the switch fabric side, the data path is divided into four independent ports of eight bits each. Each of these ports is connected to a different port of the switch fabric and traffic is distributed across these ports in a load-sharing fashion. Since cells involved in different virtual circuits may be sent on different ports, cells must be resequenced on the output of the switch. However this function is already a part of the architecture, as described in the previous progress report, and in detail in reference [7]. Hence, the only new requirement is that cells sent on different ports during the same operation cycle be labeled with a different time stamp to indicate the port that was used. In addition, the buffers in the VXT and XMB chip must provide a multiport interface. This appears to be straightforward, using the VRAM style memory design used in the first generation switch.

Note that this same approach can be used in a switch designed for 150 Mb/s ports, to allow it to support a small number of 600 Mb/s ports. We envision a campus ATM network including a large number of small concentrators providing 150 Mb/s access to desktop workstations, connected via 600 Mb/s links to one or more central switches with higher speed internal fabrics. These central switches could be linked to one another and to remote networks via either 600 Mb/s or 2.4 Gb/s facilities. Hence, port sharing provides a key element in the construction of networks supporting a rich hierarchy of transmission speeds.

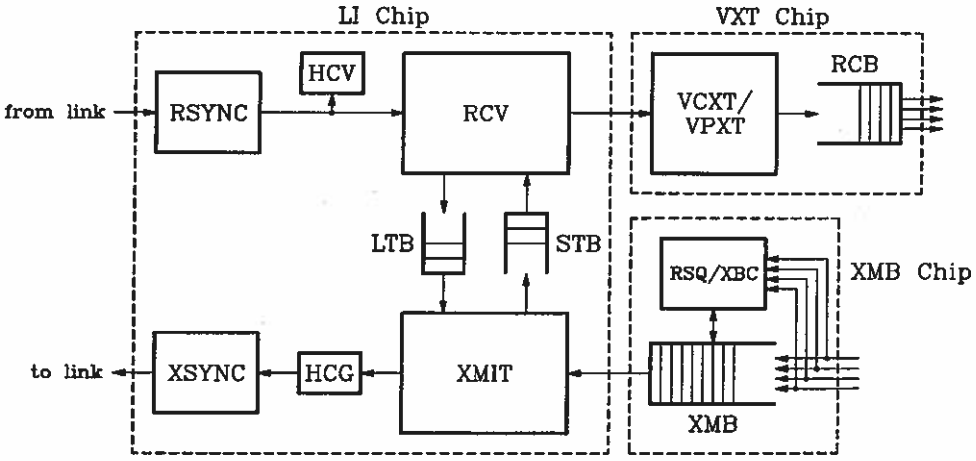


Figure 1: A 2.4 Gb/s Port Controller

Quantitative Evaluation of Switching Networks

There have been a number of architectures for ATM switching systems proposed in the literature with extensive performance data, but little in the way of comparison to indicate which architectures are preferable from a cost standpoint given specific performance requirements. Any reasonable architecture can be configured to provide a given level of performance, but the associated costs can be quite different. In this study, we compare networks on the basis of both cost and performance, to determine which architecture provides a specified level of performance for the lowest cost. To measure cost we count the number of chips needed to realize the architecture, taking into account both pin constraints and device density. We have chosen chip count because it is a dominant component in the cost of a switching system.

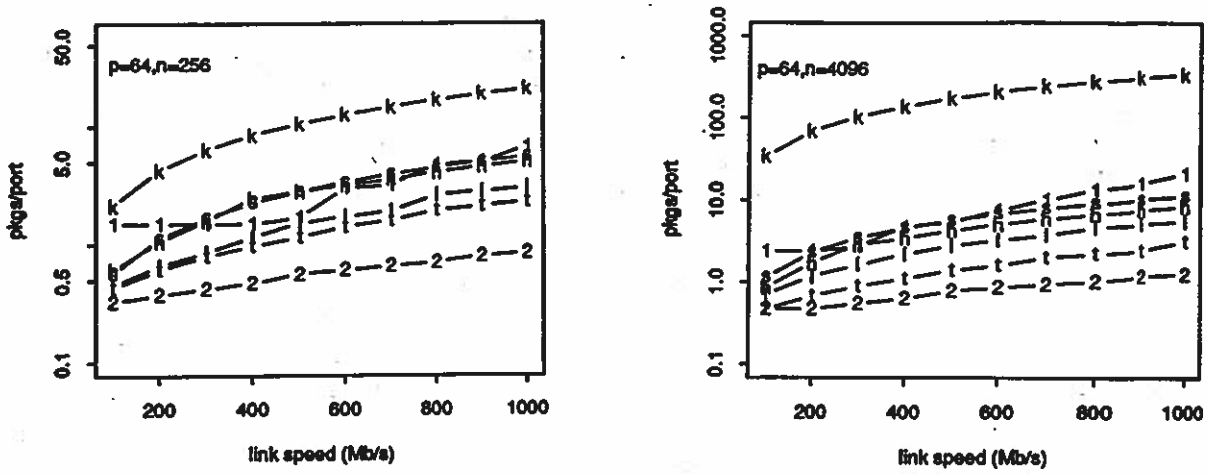
We use the term *switching system* to refer to the functional unit that interconnects the external data links. The switching system is responsible for receiving packets from external links, routing them as appropriate and transmitting the packets on external links. Within the switching system there is a *network* or *switching fabric* that performs the actual routing function. Many of the networks we consider are constructed by interconnecting multiple copies of some smaller building block. We use the term *switch element* to refer to the smaller building block.

We are interested in differences between switching systems based on architectural choices as opposed to details of implementation. Thus, we consider several broad categories of systems based on high level architecture choices. Within each category we consider one or more alternatives and develop an equation for the chip count for each alternative. These equations are used to make plots of chip count for each network over a range of parametric values. We then compare the chip counts of the various networks. Clearly the chip count depends on the strategy used to assign components of the system to chips. For each architecture, we develop a packaging strategy using as few chips as possible for that architecture, within the constraints on package size and transistor count for the chip. In considering the chip count, we focus on the switching network of each switching system and ignore the input and output circuits which interconnect the external data links. This is done on the grounds that the input and output circuit complexity is comparable for each of the networks.

In this study, we have considered only point-to-point networks. The following networks were examined:

- *Crossbar networks.* In particular, we consider the Knockout architecture in some detail.
- *Sorter Based Networks.* In particular, we study the Sunshine network and Lee's hybrid network, which includes a number of sorter based modules which feed into Knockout type output concentrators.
- *Unbuffered Networks with Deflection Routing.* In this category, we consider Tobagi's Tandem Banyan network and the Shuffleout network of Dècina, et. al. In both cases, we consider configurations with recirculation, which provide the least cost for a given performance level.
- *Buffered Beneš Networks.* Here there are many possible variations. We studied in particular, a network with fixed path routing and output buffering, and a second network with per cell routing and shared buffering.

For each of the networks, a configuration was chosen that allows the network to be essentially nonblocking and have acceptably low cell loss



k: Knockout
s: Sunshine
l: Lee's network
t: Tandem banyan
h: Shuffleout
1: buffered Beneš – fixed path/output
2: buffered Beneš – per cell/shared

Figure 2: Comparison of Network Architectures

rates. In some cases, this requires a speed advantage for the network's internal data paths that is typically realized through added parallelism within the network.

Figure 2 shows a comparison of the networks studied for configurations with 256 inputs and outputs (left side) and networks with 4096 inputs and outputs (right side). In both plots, each integrated circuit was constrained to have no more than 64 inputs and 64 outputs and the transistor count per package was limited to 500,000. The plots show how the chip counts grow as a function of the speed of the switching system's external data links. Notice that the y-axis is logarithmic and gives the number of chips per input and output. Note that in the larger system, the various systems have costs that differ by more than two orders of magnitude. Even excluding the Knockout, which is very poor in large configurations, there is a striking difference among the various alternatives.

More details on this work can be found in [11].

Design of a General Purpose Switching System Performance Evaluation and Visualization System

The evaluation of switching systems is a complex task, because there are many different components which interact in subtle and unexpected ways. Simulation is an essential tool for deriving insight into the way systems perform, as it allows the designer to reproduce the precise conditions under which a system will be used, while allowing him or her to observe the system's behavior at either a macroscopic or microscopic level.

Unfortunately, the design of effective simulators for switching systems is a time-consuming chore, since each switching system has its own set of characteristics and idiosyncrasies that must be captured, and since careful programming is necessary to achieve acceptable performance for system configurations of practical interest. We have initiated work on a simulation tool that will make it possible to simulate a wide variety of different systems with little or no programming on the part of the performance analyst. The performance analyst will be able to specify the components of the system and the way in which they are interconnected, by way of a graphical user interface, with menus from which components can be selected and powerful network construction operators, which provide common interconnection patterns. Once the system is specified, the analyst will then be able to simulate the it using appropriate traffic models, also selected and modified through menus, while monitoring the traffic parameters of interest. The graphical user interface comes into play during simulation as well, allowing the user to observe the operation of the system through a continuous animation and/or through continuous plotting of the desired data. The system will provide several advantages over traditional approaches.

- It will allow the analyst to construct a specific network and traffic configuration with an absolute minimum of effort and verify that the network operates as expected using the animation features.
- It will make it much easier to compare different configurations. Because the different networks are constructed within the same environment, they can be subjected to identical traffic and compared with far greater precision than when simulations are done independently.
- The visualization features are an excellent vehicle for illustrating a system's operation. They are also an excellent way to obtain a detailed understanding of transient behavior.

This work has been inspired in part, by an earlier animation of the broadcast packet switch simulator. This tool, while relatively crude, has proven to be extremely useful for explaining the operation of the system to visitors and for developing an understanding of certain unexpected situations that arose when the prototype system was tested. The new tool provides similar animation features, but provides far more flexibility in how networks are configured, simulated and measured. Figure 3 shows an example of a simulation window containing a simple network with input-buffered switches preceded by a set of traffic sources and input buffers, and followed by a set of output buffers and traffic sinks. The tools menu shown in the figure provides primitives for selecting and instantiating basic components, repositioning them and connecting them together.

A number of pulldown menus provide additional capabilities. The *File menu* allows a given network to be saved or restored from a file and allows the contents of a simulation window to be printed. The *Specify menu* provides a means for changing the parameters

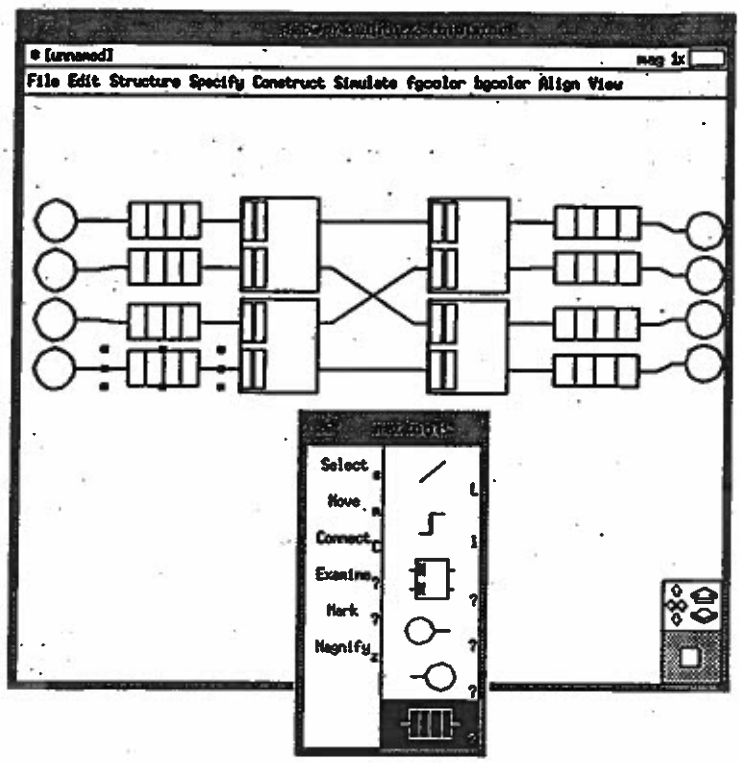


Figure 3: Example Simulation Window

of various components. For example, the number of inputs or outputs of a switch element can be varied, as can the size and placement of buffers (input, output or shared), the queueing discipline (fifo, age, priority, lifo), the type of flow control (none, grant or acknowledgement) and the function (route, distribute, copy). For traffic sources, the peak and average loads and burst length can be varied. For lookup tables, the table size and initial contents can be specified. The *Construct menu* includes options for series or parallel construction of networks, allowing large networks to be specified with just a few steps. The *View menu* controls the visual appearance of the simulation and allows multiple views of the same simulation to be shown. It also provides access to a graph editor, which is used to specify plots which can be attached to variables within the simulator, allowing the user to observe various traffic parameters as the simulation proceeds. The *Simulate menu* provides commands for controlling the simulation and includes both single-step and multi-step commands. It also includes commands for suppressing screen updates during multi-step commands, to speed up system operation.

Using the *Examine* command in the net tools window, information about each of the objects can be examined and modified. For example, if one selects a packet, one gets a dialog box containing information about the packet contents; the contents of any of the packet's fields can be modified through the dialog box, as can its color, making it possible to observe the progress of a particular packet as it passes through the network. Similarly, one can change the load offered by a source or the mapping provided by a lookup table.

Most of the key design issues for the system center on the competing objectives of generality and performance. For example, one issue arises from the question of how to route packets in networks that can be constructed with arbitrary topology. To handle this problem in full generality, each switch might require a

different routing table specifying the switch output to use to reach any given network output. In most common situations, a single table would suffice for all switches in a given stage, but users can certainly construct networks where this would not be the case.

The system is being written in C++ and is based on the InterViews user interface toolkit, which in turn is based on the X-windows system. Each of the graphical objects is implemented as a C++ class and includes member functions for accessing internal state, executing a simulation step and updating its on-screen representation. At this writing, the system is still in a preliminary stage of development. While many of the desired capabilities have been implemented, others are still being developed.

Congestion Control Using Fast Buffer Reservation

In our previous progress report, as well as in reference [6], we have described a novel approach to resource management in virtual circuit packet networks that offers the first really complete approach to the problem. In [6] we studied the implementation in some detail to obtain a full understanding of how the scheme would impact a high speed switch architecture. During the past year, we have studied the performance of fast buffer reservation relative to statistical multiplexing, via simulation and analysis.

Figure 4 is a typical comparison showing the performance of fast buffer reservation in a mixed traffic environment. This figure shows the packet (not cell) loss rate for transport protocol packets, for a multiplexed combination of two different source types. The characteristics of the source types are illustrated at the top. Both send with a peak rate of 30 Mb/s when active and have a peak-to-average ratio of 4:1. They differ in the average duration of their bursts and in the size of the transport protocol packets, which are 200 Kbytes for type 1 sources and 20 Kbytes for type 2 sources. The top two curves are for ordinary statistical multiplexing, while the bottom two (which are virtually indistinguishable) are for fast buffer reservation. Note that fast buffer reservation reduces the packet loss rate for type 1 sources by more than an order of magnitude and provides far more consistent performance than statistical multiplexing.

One advantage of fast buffer reservation is that it makes packet loss rate largely independent of packet size. This allows transport protocols to increase the packet size they use, in order to reduce software overhead at the hosts. The use of very large packets (tens of kilobytes) is already common in local area supercomputer networks. As speeds continue to move up into the gigabit range, techniques like fast buffer reservation will allow packet sizes to scale in order to avoid host bottlenecks, without requiring major transport protocol changes.

In the past year, we have also discovered and corrected and error in the call acceptance algorithm associated with the fast buffer reservation scheme. The corrected algorithm is described in [9].

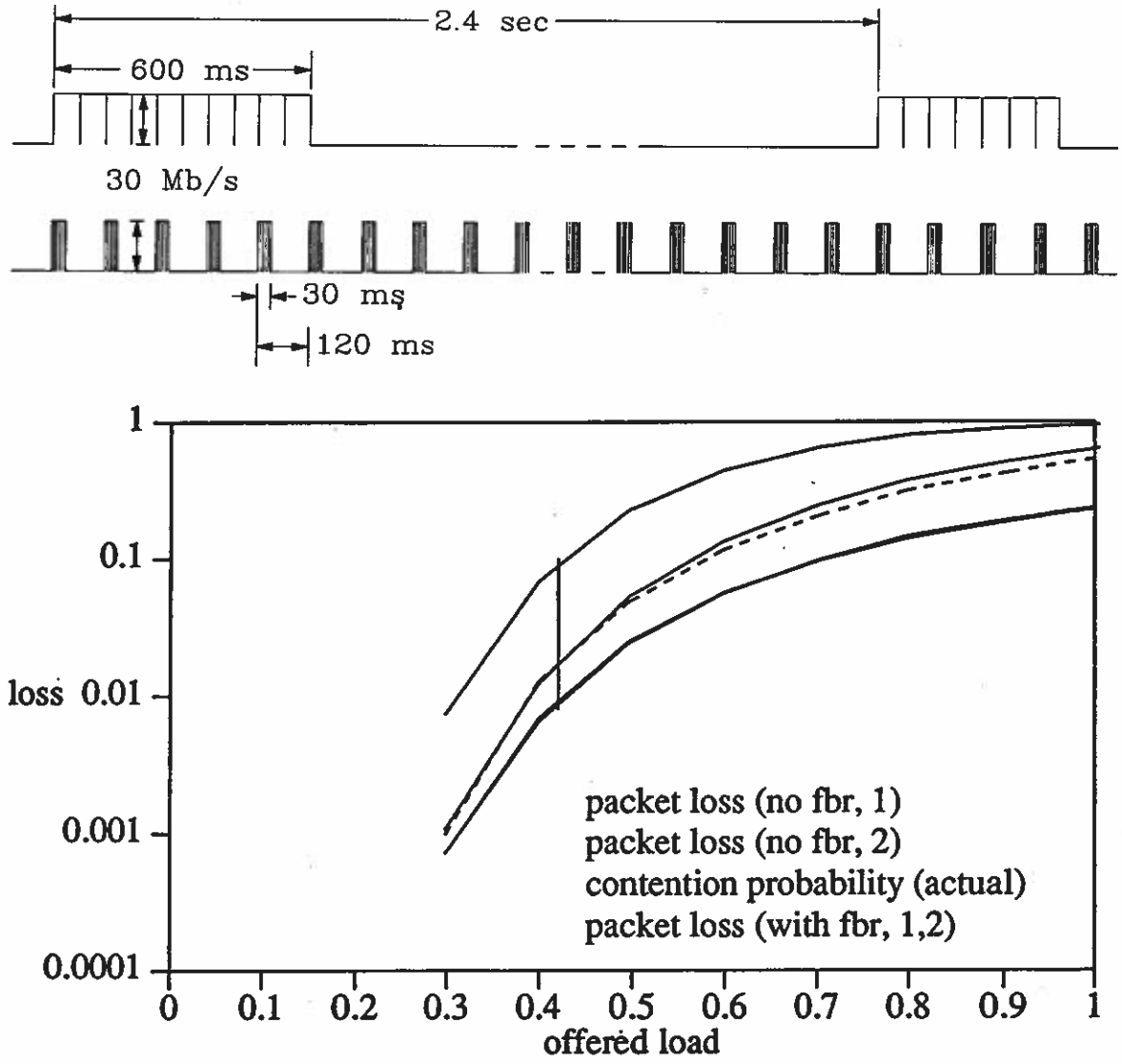


Figure 4: Transport Packet Loss Rate for Fast Buffer Reservation and Statistical Multiplexing

Improved Queueing Analysis of Buffered Switching Networks

Reference [5], analyzes the queueing performance of switching networks comprising switches with shared buffering and flow control. This analysis leads to a fast computational procedure for determining the delay and throughput of such networks.

We model each switch in the network as a $B+1$ state Markov chain. We let $\pi_i(s)$ be the steady state probability that a stage i switch contains exactly s packets and we let $\lambda(s_1, s_2)$ be the probability that a switch with s_1 packets during a given cycle contains s_2 packets in the subsequent cycle. Let $p_i(j, s)$ be the probability that j packets enter a stage i switch that has s packets in its buffer and let $q_i(j, s)$ be the probability that j packets leave a stage i switch that has s packets in its buffer. Then

$$\lambda_i(s_1, s_2) = \sum_h p_i(h, s_1) q_i(h - (s_2 - s_1), s_1)$$

Let a_i be the probability that any given predecessor of a stage i switch has a packet for it. Then if we let $m = \min\{d, B - s\}$,

$$\begin{aligned} p_i(j, s) &= \binom{m}{j} a_i^j (1 - a_i)^{m-j} \\ a_i &= \sum_{0 \le j \le B} \pi_{i-1}(j) \left[1 - (1 - 1/d)^j\right] \end{aligned}$$

Let b_i be the probability that a successor of a stage i switch provides a grant and let $Y_d(r, s)$ be the probability that a switch that contains s packets, contains packets for exactly r *distinct* outputs. Then

$$\begin{aligned} q_i(j, s) &= \sum_{j \le r \le \min\{d, s\}} Y_d(r, s) \binom{r}{j} b_i^j (1 - b_i)^{r-j} \\ b_i &= \sum_{0 \le h \le B-d} \pi_{i+1}(h) \\ &\quad + \sum_{0 \le h \le d-1} \pi_{i+1}(B - h) h/d \end{aligned}$$

Y is easily calculated, assuming all distributions of s packets to the d outputs are equally likely. We compute performance parameters by assuming a set of initial values for $\pi_i(j)$, then use the equations given above to compute $\lambda_i(s_1, s_2)$. These, together with the balance equations for the Markov chain are used to obtain new values of $\pi_i(j)$ and we iterate until we obtain convergence.

In this analysis, we represent the state of a switch by the number of packets it contains and *assume* that the stored packets are equally likely to be destined for any of the switch's outputs. This assumption is used in the equation for a_i and again in the equation for $Y_d(r, s)$. This assumption ignores the correlations between packet destinations that develop as packets contend with one another. Comparing the results of analysis with simulation, we have identified conditions under which the analysis overestimates a network's maximum throughput by as much as 30%.

Pattavina and Monterosso [3] call the above model the *scalar model* and have proposed instead, a *vector model* in which the state of a switch is represented not by the number of stored packets, but by a vector containing the number of packets for each destination. The vector model is exact for a single stage network and is reasonably accurate for multistage networks as well. On the other hand, the state space grows exponentially with the size of the switches, making it applicable only to networks with up to four ports per switch.

We have developed an alternative scalar model that seeks to match the accuracy of the vector model while avoiding its computational complexity. This model is based on the observation that when a switch is in the steady state, the average number of arriving packets destined for a particular switch output port equals the average number of packets departing via that output port. When the original scalar model is compared to the vector model, it's easy

to see that the scalar model in effect, implicitly assigns probabilities to the vector model states according to a multinomial distribution (when $d = 2$, it is a binomial distribution). However, the observation concerning the balance of arriving and departing packets suggests that the probabilities of these states should be approximately equal. This leads to an alternative scalar model in which uniform probabilities are assigned to these states.

The *uniform scalar model* yields good results for networks with large buffers ($B > d^2$), relative to d, precisely the case where the original scalar model was least accurate. For these cases, the predicted throughput is generally within 5% of that predicted by simulation. However, substantial inaccuracies remain in the practically important case of large d and $B/d \leq 4$. The problem here is that for such switches, boundary states (states in which there are some outputs for which there are no packets in the buffer) are very common, but occur with lower probability than is assigned to them by the uniform scalar model.

To compensate for this, we have developed a *bidimensional model* in which the state includes a second variable which represents the number of outputs for which there are cells (the number of active outputs). This method, while computationally more expensive, is much more accurate, predicting network traffic capacity within a fraction of a percent in most cases. We have also devised an intermediate method, called the threshold method which keeps track of whether the number of active outputs is above or below some fixed threshold, but does not track the exact number. The threshold method yields surprisingly accurate results, rivaling the bidimensional method in accuracy but with substantially lower computational cost.

See [9] for details.

References

[1] Hill, Rex. "Implementation of a Gigabit Packet Switch Element," Washington University Computer and Communications Research Center, WUCCRC-91-3.

[2] Jenq, Yih-Chyun. "Performance Analysis of a Packet Switch Based on a Single-Buffered Banyan Network," *IEEE Journal on Selected Areas in Communications*, vol. SAC-1, no. 6, 12/83, 1014–1021.

[3] Pattavina, Achille and Roberto Monterosso. "A Vector Model for Analysis of Buffered Switching Networks," CEFRIEL technical report, 1991.

[4] Szymanski, Ted and Salman Shaikh. "Markov Chain Analysis of Packet-Switched Banyans with Arbitrary Switch Sizes, Queue Sizes, Link Multiplicities and Speedups," *Proceedings of Infocom 89*, 4/89.

[5] Turner, Jonathan. "Queueing Analysis of Buffered Switching Networks," *Proceedings of the International Teletraffic Congress*, 6/91. To appear in *IEEE Transactions on Communications.*

[6] Turner, Jonathan. "A Proposed Bandwidth Management and Congestion Control Scheme for Multicast ATM Networks," Washington University Computer and Communications Research Center, WUCCRC-91-1.

[7] Turner, Jonathan. "Resequencing Cells in an ATM Switch," Washington University Computer Science Department, WUCS-91-21. Submitted to *IEEE Transactions on Communications.*

[8] "A Practical Version of Lee's Multicast Switch Architecture," Turner, Jonathan. Washington University Computer Science Department, WUCS-91-46. To appear in *IEEE Transactions on Communications.*

[9] Turner, Jonathan. "Bandwidth Management in ATM Networks Using Fast Buffer Reservation," to appear in *Networks Magazine*, 8/92.

[10] Turner, Jonathan. "Improved Queueing Analysis of Shared Buffer Switching Networks," Washington University Computer Science Department, WUCS-92-19. Submitted to *Infocom 93.*

[11] Witte, Ellen. "A Quantitative Comparison of Architectures for ATM Switching Systems," Washington University Computer Science Department, WUCS-91-47. Submitted to *IEEE Journal on Selected Areas of Communications.*