Dynamic Reconfiguration of an Optical Interconnect

Praveen Krishnamurthy
Mark Franklin
Roger Chamberlain

Praveen Krishnamurthy, Mark Franklin, and Roger Chamberlain, "Dynamic Reconfiguration of an Optical Interconnect," in *Proc. of the 36th Annual Simulation Symposium*, April 2003, pp. 89-97.

Computer and Communications Research Center
Washington University
Campus Box 1115
One Brookings Dr.
St. Louis, MO 63130-4899

Dynamic Reconfiguration of an Optical Interconnect

Praveen Krishnamurthy, Mark Franklin and Roger Chamberlain
Computer and Communications Research Center
Washington University in St. Louis, Missouri
praveen, jbf, roger@ccrc.wustl.edu

Abstract

The advent of optical technology that can feasibly support extremely high bandwidth chip-to-chip communication raises a host of architectural questions in the design of digital systems. Terabit per second (and higher) bandwidths have not previously been available at the chip level. In this paper we examine the use of this optical technology in the design of a network router, a system for which previous designs have been I/O limited at the chip level. Specifically, we examine the benefits of bandwidth reconfigurability, a capability of the proposed switch fabric, and illustrate the performance impact associated with different reconfiguration rates.

1. Introduction

One of the significant issues the networking industry faces is meeting continually increasing bandwidth requirements. Optical technology has made significant contributions to this need by providing high bandwidth link solutions in the form of long distance fiber optics. Significant additional benefits can be attributed to the development of Wavelength Division Multiplexing (WDM) and Dense Wavelength Division Multiplexing (DWDM) [4], which provide additional capacity on existing fiber optic channels.

The above advances focus on link technologies. In this paper our interest is on router technology, traditionally an all-electronic function. Core routers are responsible for connecting links at a low-level network protocol layer (technically layer 2 of the OSI model)[1]. Higher capacity routers, and the switch fabrics within them, are needed to fully utilize the capabilities of current link technology.

There has been a significant effort focused on *all-optical switching*, which completely eliminates the signal conversion between the optical and electronic domains. Fast optical switching components, however, are still expensive thus limiting commercial options in this area. Research has also focused on identifying suitable architectures and associated router switch technologies (e.g, ring, mesh, multiring [7], etc.) and on routing issues [1, 8].

Our approach exploits the high bandwidth of optics for chip-to-chip communication, but retains the benefits of traditional silicon CMOS technology for decision, control, and signal switching functions. This paper presents a simulation-based performance analysis of a router's switch fabric constructed using optical technology for communicating between the CMOS chips that implement the fabric. The extremely high bandwidth afforded by the optics is enhanced further by the ability to reconfigure the fabric, essentially matching the bandwidth capability of the switch to the bandwidth requirements imposed on the switch by its external environment.

Although the ability to reconfigure a switch fabric is not unique to an optically-connected system, the specific mechanisms employed are pertinent to the architecture of the fabric. As a result, this study focuses on a specific system design tailored for use with optical chip-to-chip communications capability.

With the ability to reconfigure the fabric comes the responsibility of controlling the reconfiguration. We have previously investigated this issue in an environment where the bandwidth requirements are known *a priori*, and the configuration can be performed statically [3]. In this paper we consider the situation where the bandwidth requirements are unknown until traffic passes through the switch. Thus dynamic reconfiguration techniques must be employed if one is to exploit the full bandwidth capabilities available. We consider a particular switch topology (i.e., multiring) that derives from the characteristics of the optical technologies employed and explore the performance of a particular dynamic reconfiguration algorithm. A key aspect of this exploration is the time period associated with reconfiguration activities.

The outline of the paper is as follows. Section 2 describes the design of the switch fabric, and Section 3 presents the simulation model of the system. Section 4 fol-

[1] Note that routers that are located near the network periphery may also perform higher level layer functions.

lows with performance results, and Section 5 concludes.

2. System Description

At the core of the proposed switch fabric are two dimensional arrays of lasers and detectors. The lasers are Vertical Cavity Surface Emitting Lasers (VCSELs) which emit light perpendicular to the surface of the chip. Chip-to-chip optical interconnections can then be supported through the use of flexible fiber image guides. Figure 1 illustrates an optical ring that connects four chips.

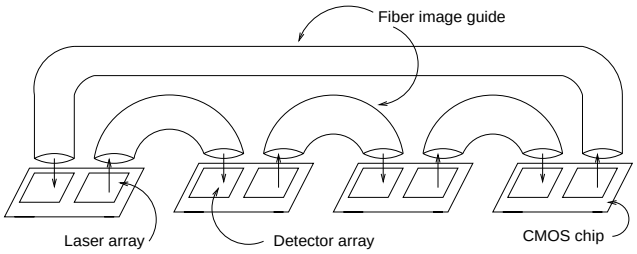

Figure 1. A 4 chip optical ring topology using flexible fiber image guides

The availability of a large number of lasers and detectors on a single CMOS chip facilitates the partitioning of these the optical links into channels, such that each individual channel is composed of multiple light paths or optical links. Figure 2 illustrates the allocation of lasers and detectors for a four channel system utilizing 32×32 arrays of optical elements. As shown in the top of the figure, one quarter of the elements are used for each channel. Each square in the top view of Figure 2 contains a single laser or detector. If the individual element communicates at 1 Gb/s, this yields $(32 \times 32)/4 = 256$ Gb/s per channel. The side view of the figure illustrates (conceptually) how two adjacent chips communicate with each other using a flexible optical pipe. The feasibility of such an interconnect system has been demonstrated by Plant et al. [10]. Their system flip-chip bonded a 32×16 array of inter-digitated lasers and photodiodes to a CMOS chip using heterogeneous integration techniques. Derivative systems are now commercially available from Teraconnect Inc. (e.g., $Teralink^{TM}24$ and $Teralink^{TM}48$ modules [12]).

Our proposed design uses this technology to build a switching fabric for network routers. Though there has been much work done in the area of optical switching, fabrics are still primarily all electronic/electrical designs. Optical technology on the other hand, has primarily been used as a link technology. We propose a dual approach, where the optical paths (laser-detector pairs) are used for chip-to-chip communication and CMOS logic provides control and switching

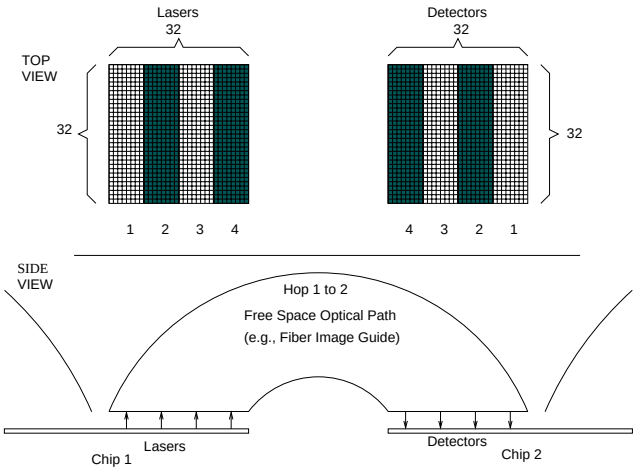

Figure 2. Allocation of laser-detector pairs to a four channel system. 32×32 laser-detector arrays are used, with a 8×32 array allocated to each channel

functions. This system will then exploit the high bandwidth available with optics and also the simplified switching inherent in using CMOS.

We next describe the architecture of the switch as well as its reconfigurability capabilities. The reconfiguration capability provides for flexible control over the bandwidth allocations to the flows supported by the switch.

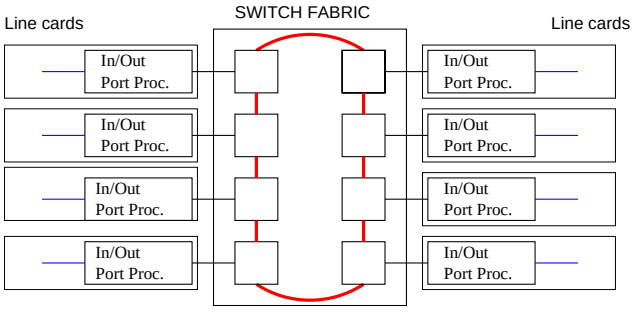

Figure 3. Optical chip-to-chip communication as part of a router's switch fabric. The heavy lines indicate optical links.

Figure 3 shows a logical view of an eight port switch. The switch ports are attached to a multiring topology with each subring being dedicated to delivering traffic to a particular destination chip (output port). As illustrated in Figure 4 for a 4 node multiring, one subring is associated with each of the ports (or destinations).

The subrings within the multiring share the bandwidth

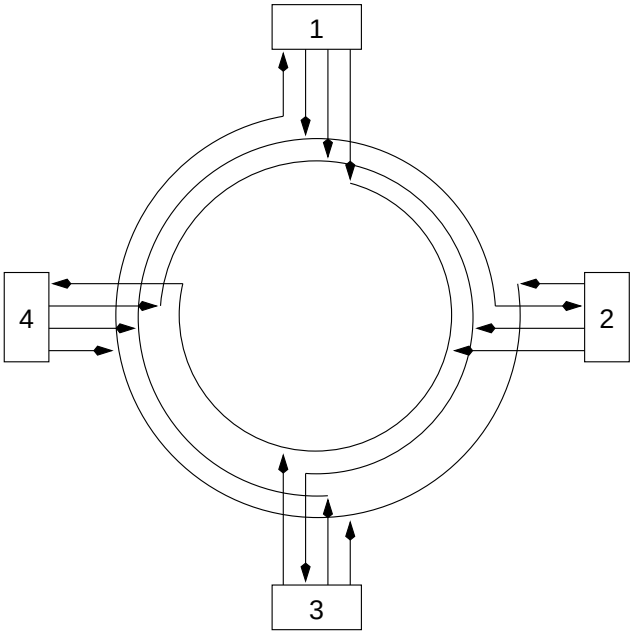

Figure 4. Four node multiring topology showing logically different channels (subrings)

of the entire system where the total bandwidth is determined by the number of laser-detector pairs available and the bandwidth associated with each pair. Bandwidth sharing is done by allocating a portion of the laser-detector pairs to each subring. Within each subring only one source can be active a time. The Deficit Round Robin scheduling algorithm has been selected to arbitrate across competing sources for access to each channel or subring.

Thus, this architecture gives us two levels of bandwidth allocation in the multiring. The first level is among the subrings for sharing the total bandwidth, and the second is among sources within each subring for sharing the bandwidth of a single subring. These two, when combined, give us the flexibility to allocate bandwidth on a per-flow basis. There are two methods of reconfiguring the interconnect corresponding to the two levels of bandwidth allocation. The first method is called Laser Channel Allocation (LCA). With LCA, bandwidth is allocated by changing the number of laser-detector pairs on a per-subring (per-channel) basis. This determines the absolute bandwidth allocated to that particular subring.

For the second method, within a subring, media access is arbitrated using the Deficit Round Robin (DRR) fairness protocol [6, 11]. The protocol supports the assignment of arbitrary bandwidth ratios to sources in a subring, accomplished by varying the "quanta" allocated to each of the sources. Over a given time period, each source has a quanta (i.e., equivalent to a bandwidth resource) that it can utilize. The DRR protocol determines whether a source has used all its quanta and determines interconnect access so that each source obtains its minimum quanta over the specified time period. Thus, within a subring, the bandwidth associated with each of the sources can be matched to the needs of the application. Note that with DRR, unused quanta are reallocated to sources requesting bandwidth in a round robin fashion. These methods are described in greater depth in [3, 5].

In this analysis each of the ports associated with the multiring interconnection network is connected to a router line card that acts both as a source and a sink of communications (e.g., packet) traffic. If one knows *a priori* the bandwidth requirements and routing statistics (i.e., the probabilities of a given packet arriving on port i being sent to port j) associated with traffic coming into a port, then LCA and DRR can be adjusted so the network bandwidth is used optimally and the latencies through the network are minimized [3]. However, telecommunications traffic is difficult to predict and changes over time. Thus, a dynamic approach to utilizing these two reconfiguration methods is taken. That is, periodically the methods are applied and the multiring and DRR characteristics are optimized for a given time period. This is done based only on the current state of the system (i.e., the control system is memoryless) as measured by the queue lengths associated with the system sources. If it is done often enough, then system performance is improved. The remainder of the paper considers a set of simulation experiments whose goal is to quantify these gains and examine the effects of frequency of reconfiguration on performance.

We now consider the control algorithm itself, that is, the way source queue lengths are used to change source bandwidth allocations. Our control algorithm synchronously changes the per flow bandwidth based on the instantaneous queue lengths for each source destination pair. We first divide the total bandwidth available in the system across the 8 channels, giving each an amount directly proportional to the backlog at each destination port. We also ensure that each subring is given a bandwidth of at least $8\ Gb/s$. We then proceed to allocate relative bandwidth for the sources within each subring. This is done by changing the quantum associated with each source in the DRR scheduling algorithm. The quantum each source gets is proportional to its relative load within each subring. In this case also we give each flow a minimum quantum to prevent starvation of any source.

3. Simulation Model and Experiments

To study the performance of this reconfigurable switch fabric, an eight port switch is simulated using the ICNS framework [2]. The ports are arranged in a multiring configuration as described in Section 2.

3.1. Traffic Modeling

At the highest level, traffic associated with each port is generated by applications (e.g., interactive file transfer) that execute on some remote site. Such traffic has been modeled using a *self-similar* (i.e., bursty) input arrival distribution. The distribution itself is implemented in the simulations (discussed in the next section) using the approach described in [9]. The size of the bursts generated in this manner correspond to a *heavy-tailed* distribution that is characterized by:

$$P[X > x] \sim x^{-\alpha} \quad x \to \infty$$

where $0 < \alpha < 2$. Such heavy-tailed distributions are obtained from the $Pareto$ distribution, where the probability density function is given by:

$$p(x) = \alpha k^{\alpha} x^{-\alpha-1}$$

where $\alpha, k > 0$ and $x \geq k$. The cumulative distribution function has the form:

$$F(x) = P[X \leq x] = 1 - (k/x)^{\alpha}$$

where the parameter k represents the smallest possible value of the random variable. The interarrival time between two such bursts is exponentially distributed and corresponds to the OFF period of the ON/OFF model [13]. The mean burst size is $\approx 4.1KB$ as obtained from [9]. Empirical results from [9, 13] reinforce the validity of this traffic arrival model.

Such heavy-tailed distributions lead to the generation of very long bursts. Both for design purposes (e.g., buffer sizes, etc.) and to reduce the potential of such long bursts from capturing network resources for long time periods (and thus blocking other users), we divide all input bursts into packets that are a maximum of 64000 bytes in length. The DRR protocol gives access to the link based, in part, on the size of each packet at the beginning of each source queue in the subring. This ensures that all the flows have the opportunity to share the link (based on their DRR priority), even when a particular flow is transmitting a very large burst. The bandwidth requirement for each flow is characterized by its instantaneous state which is captured by input queue length associated with each subring source.

3.2. The Simulation

Due to the effects of using a heavy-tailed distribution some very long bursts are generated and, under different reconfiguration approaches, halting the simulation and gathering statistics while there remain packets in the various queues can produce misleading results. Let us call the case having uniform allocation of bandwidth across the sources in each multiring and not performing any reconfiguration the *Base Case*. Say we would like to compare the *Base Case* with the case of using periodic reconfiguration with a given period p (called the *RC(p) Case*). If both cases are halted at the same arbitrary time point, the result will be that each case processes a different number of packets and a direct performance comparison will be misleading. To deal with this, the simulations have been implemented so that each simulation is halted when the same number of packets have been processed. In practice this means that after a certain number of packets have been generated, the packet traffic generation component of the simulation is halted, however, the simulation continues until all packets remaining in the system are processed. Thus, for the experiments to be described, the generation process is halted after 40 *ms*. Each simulation experiment is performed multiple times to ensure the statistical significance of the results. The specific traffic pattern that is present for an individual experiment is referred to as an input pattern.

We study the effect of reconfiguring the system periodically with the period, p, being the reconfiguration parameter of interest. In particular, we reconfigure the system with $p = 4$ *ms*, 400 μ*s*, and 4 μ*s*. These correspond to 10, 100 and 10,000 reconfigurations for each input pattern. The minimum period between reconfigurations is 4 μ*s* (i.e., the *RC*(4 μ*s*) *Case*). This time is dependent on the underlying signaling mechanism for used for distributing data and control messages which has a reset period of 4 μ*s* [6]. Additionally, a burst is divided into packets with a maximum size of 4 μ*s* (i.e., 64,000 Bytes). Thus, in a system where there is no overhead associated with reconfiguration, this represents the fastest rate at which reconfiguration can occur and results in the best possible performance for the system. Later, overhead delays are included in the overall model and these effects on performance are examined.

4. Results

The results presented in this section correspond to an offered load to the switch of 50%. At loads closer to 100% utilization, the switch is busy all the time, thus there is no significant advantage to system reconfiguration. All the resources are being fully utilized. Similarly, at very low loads there is enough bandwidth to fully satisfy all requests and thus there are no performance benefits to reconfiguration. Thus, it is in this middle load range, where some links may be bottlenecks while others are underutilized, that reconfiguration can aid performance.

4.1. Queue Lengths

The first set of performance graphs (Figure 5), illustrate the behavior of the system for a particular input traffic

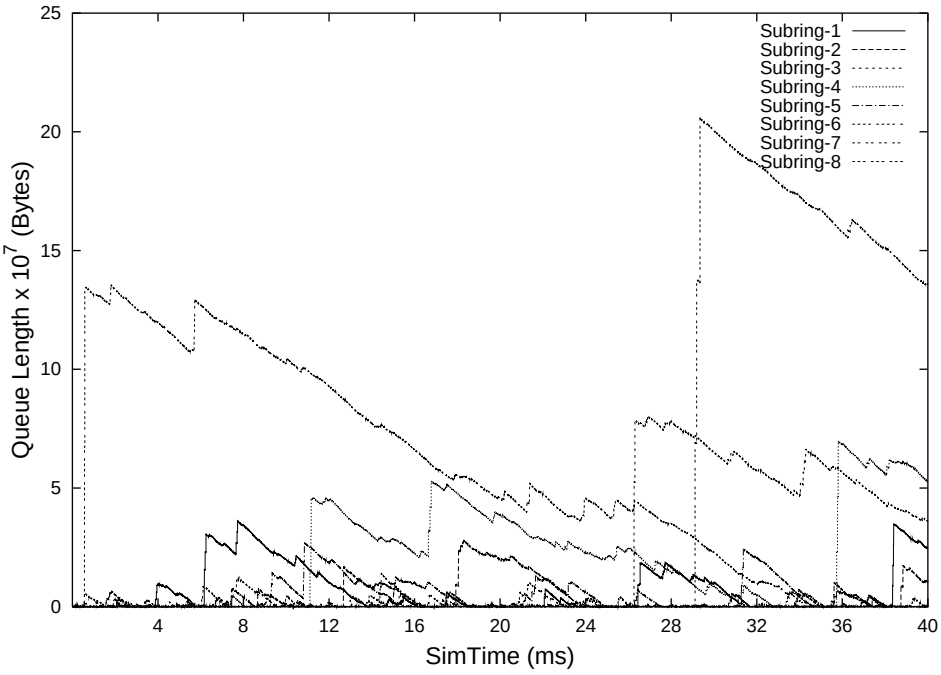

Figure 5a. Instantaneous queue lengths with no reconfiguration (Base Case)

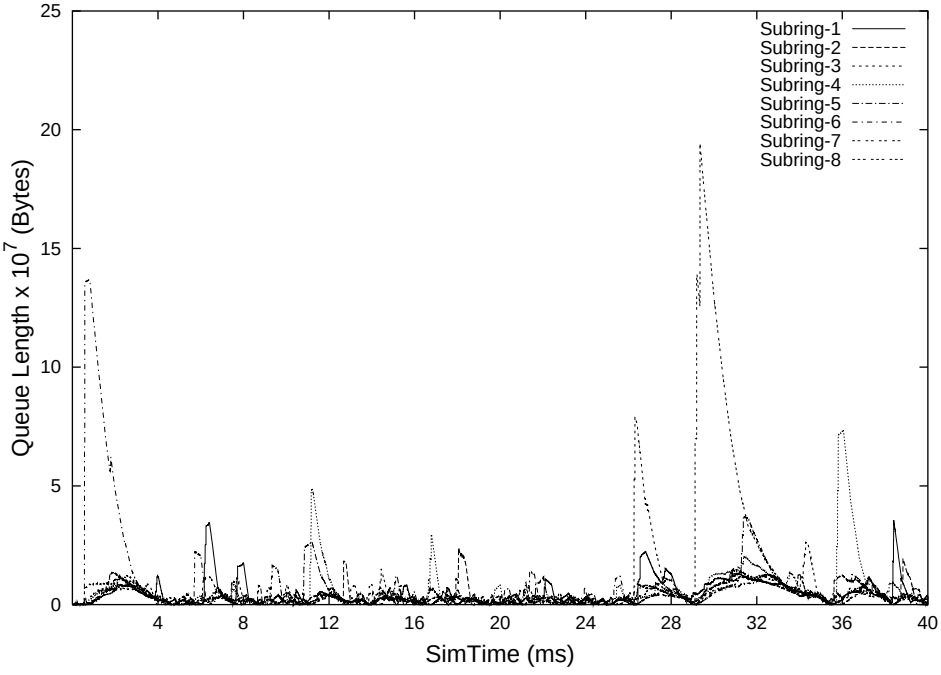

Figure 5b. Instantaneous queue lengths with reconfiguration RC(400 μs)

pattern. The different plots show the instantaneous queue length associated with each destination port corresponding to a particular subring. Each plot is designated by a different printing pattern (e.g., dashed, dotted, etc.). This queue length information is used as a basis for reconfiguration. Figure 5a corresponds to the *Base Case* where there is no system reconfiguration and the bandwidth is evenly divided across all sources and subrings. Since all sources have the same priority and bandwidth, the rate at which packets are removed from the destination queues is the same for all queues. This dequeue rate corresponds to the downward slope of each queue length plot and can be seen to be constant and roughly equal. The random vertical jumps correspond to traffic being generated for the switch that is targeted to a particular destination port. Note also that large bursts can occur since the traffic model is heavy tailed. These bursts represent requests for bandwidth and stay in the system for long periods since, in the *Base Case*, no extra bandwidth is allocated to them. Thus, they will have an adverse effect on overall performance averages.

In contrast to this, Figure 5b illustrates the response of the system for the same input traffic pattern used above, but with the system reconfigured every 400μs (i.e., *RC(400 μs) Case*). The slope of the plots when dequeuing occurs are significantly different than in the *Base Case*. They are steeper for more heavily loaded sources (indicated by high queue lengths) and flatter for the ports with lower traffic. This illustrates that the redistribution of bandwidth based on source queue length using LCA and DRR is operating as expected. Thus, the higher the load on a source, the greater the bandwidth allocated to it.

From the plots, it is also clear that the average queue length over all sources is significantly lower for the *RC(400 μs) Case*. This is quantified in Table 4.1 where the average queue lengths over the entire simulation are given for each of the destination nodes. This data is obtained by simulating 10 different input patterns. Over all the nodes there is a 72% reduction in queue length for the *RC(400μs) Case* over the *Base Case*.

4.2. Packet Delays

Figure 6 illustrates the average packet delay over the entire system and also the average delay across individual subrings. Figure 6a shows the overall packet delay, aggregated from 10 different runs. The overall packet delay for the *RC(4 ms)* reconfiguration case is higher than the uniform allocation case. The *RC(400 μs)* and *RC(4 μs)* reconfiguration cases, on the other hand, give an improvement in the mean packet delay in the system.

Figure 6b gives the delay numbers for the individual subrings across 10 runs. We see here that the spread of these numbers in the uniform allocation is much higher than the

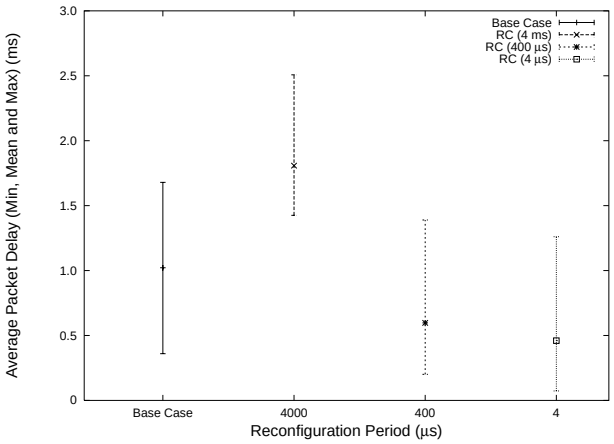

Figure 6a. Average packet delay in the system (overall)

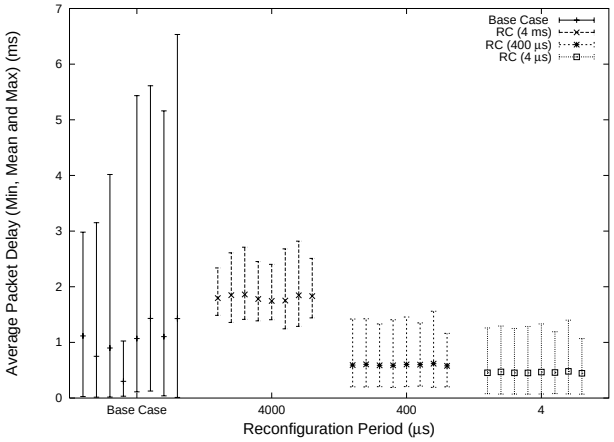

Figure 6b. Average packet delay in the system (across subring)

reconfigured cases, which is expected, because of the control algorithm controlling the variability in the reconfigured runs.

The degradation in mean packet delay performance for the *RC(4 ms)* reconfiguration is a result that is common to memoryless control systems. The system is configured based on demand at a particular point in time. At a later time, however, the demand is potentially significantly altered, and unless the control system reacts to the demand change, the system itself is poorly configured to service the actual demand present. An illustration of this effect is shown in Figure 7, which shows the instantaneous queue lengths for a 4 ms reconfiguration period (*RC(4 ms)*). The input pattern generating the trace is the same as is shown in Figure 5.

We see that at time 28 ms when the system is recon-

Table 1. Mean queue lengths (bytes)

	Port 1	Port 2	Port 3	Port 4	Port 5	Port 6	Port 7	Port 8	Average
Base Case	6835328	4158080	20745216	20634496	4724224	57565632	53452800	46317824	20227008
400 μs	4645056	4125824	5144960	6070784	4878656	7358528	3556608	9388416	5646 080
%Improv.	32.04%	0.77%	75.19%	70.57%	-3.26%	87.21%	93.35%	79.73%	72.09%

figured, the number of packets in queue for port 8 is quite small. This will imply the bandwidth allocated to that port is low. This port then receives a huge burst after a short time, which we see is not detected by the control algorithm until the next period (at time $32\ ms$). Until this later reconfiguration, few packets in this queue are serviced, which is seen by the increasing slope of the queue length for destination 8. This increases the average packet delivery time due to a poorly configured system. Further analytical analysis of this effect is given in [5].

One metric that is of interest is the *Speedup* of the reconfigurable system over the *Base Case*. The speedup is defined as:

$$S = \frac{AverageDelay_{Uniform}}{AverageDelay_{Reconfig}}$$

For the *RC(4 μs) Case* and the *RC(400 μs) Case* the speedups are 1.71 and 2.22, respectively, when comparing the average packet delay for the overall system. The *RC ($4\ ms$)* case has a speedup of 0.57, suggesting that a $4\ ms$ reconfiguration period is clearly inappropriate for this system.

4.3. Delay Variation

Another important performance metric is the variability in packet delay. Given the control algorithm in use, we expect a significant reduction in the delay variability for the reconfigured system. Table 2 presents the standard deviation of the packet delays experienced over all 10 simulation runs.

Our reconfiguration mechanism, as mentioned earlier, makes the bandwidth allocations proportional to the load on individual flows. As expected, the variability of packet delivery times decreases with the number of times the system is reconfigured. It is also seen that the numerical difference is not significant between the $400\ \mu s$ and the $4\ \mu s$ reconfiguration periods.

4.4. An Expanded Performance Model

The reconfiguration performance results shown up to now don't consider the cost of reconfiguring the system. We now present an analytic model that includes this overhead in a performance prediction of mean delay.

Table 2. Overall standard deviation over all runs

Reconfiguration Period	Std. Dev. (ms)
Base Case	10.74
RC ($4\ ms$)	1.32
RC($400\ \mu s$)	0.9172
RC ($4\ \mu s$)	0.9142

We add a post simulation reconfiguration penalty delay for all the packets present in the system at times when the system is to be reconfigured. This reconfiguration cost is obtained using *Little's law* and simulated mean packet delivery time. We estimate the number of packets present in the system at each reconfiguration period and add penalties to all these packets, averaging the total reconfiguration cost over the total number of packets delivered by the system. The average reconfiguration penalty per packet (R_c) can be derived as

$$R_c = \frac{Q_t \times P \times M}{N}$$

where Q_t is the mean queue length during the simulation run, P is the reconfiguration penalty, M is the number of times the system is reconfigured during the entire simulation run and N is the total number of packets delivered during the entire simulation run. From Little's law we know that $Q_t = \lambda \times W_t$, where W_t is the wait time in the system and λ is the mean arrival rate. Using the numbers we have from the simulation, the expression for R_c now becomes

$$R_c = \frac{\lambda \times W_t \times P \times M}{N}$$

where $\lambda = \frac{N}{Simulationduration}$. This now reduces the expression for R_c to

$$R_c = \frac{W_t \times P \times M}{40\ ms}$$

Since Little's law holds only for steady state conditions in the system, it is important to show its validity in computing the reconfiguration cost. To estimate the error introduced by using Little's relationship we simulated a pair of input patterns in which a delay histogram with a bin size of $4\ \mu s$ was constructed. We proceeded to calculate the reconfiguration penalty for each bin assuming the mean of the bin

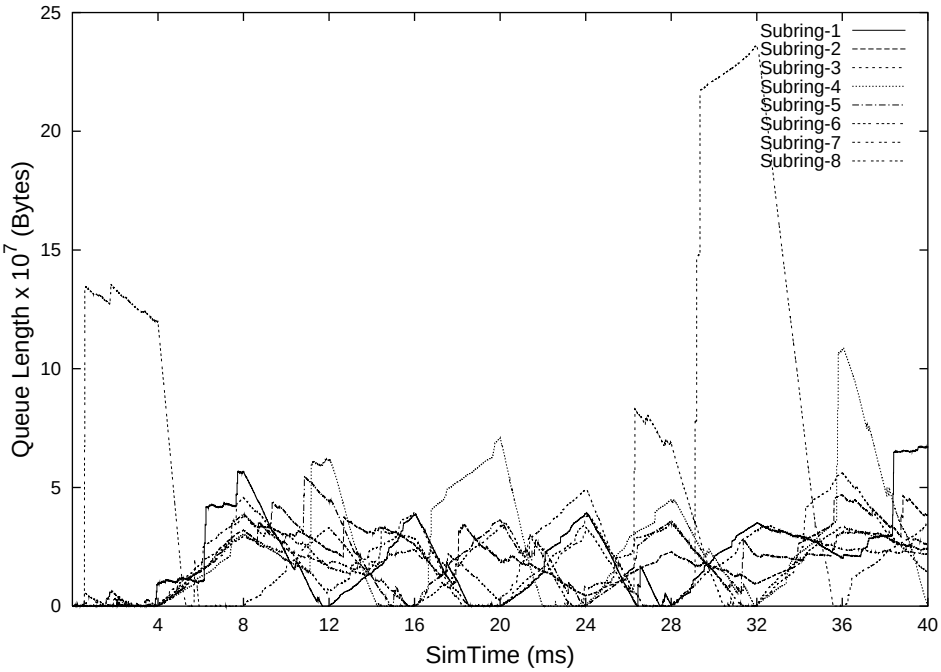

Figure 7. Instantaneous queue lengths with reconfiguration RC ($4\ ms$).

Table 3. Comparison between simulation model and Little's law

Experiment	Histogram	Little's law
RC ($4\ ms$*)* (run 1)	0.42580	0.42575
RC ($4\ ms$*)* (run 2)	0.58895	0.58890
RC ($400\ \mu s$*)* (run 1)	0.87331	0.87295
RC ($400\ \mu s$*)* (run 2)	2.86251	2.86218
RC ($4\ \mu s$*)* (run 1)	53.40197	53.37588
RC ($4\ \mu s$*)* (run 2)	248.02417	248.00300

corresponded to the mean delay for the packets in that particular bin. We calculated the probability that a packet with this packet delay is present in the system during reconfiguration. With this probability, the number of packets in each bin, and the penalty P associated with each reconfiguration, we calculated the reconfiguration penalty.

The results are tabulated in Table 3, which shows the per packet additional delay (to account for the reconfiguration cost) for both Little's law and the histogram that results from the simulations. When comparing the numbers in Table 3 we can conclude that using Little's law gives us a fair estimate of the reconfiguration cost. The numbers presented in the table correspond to a reconfiguration penalty of 1 ns ($P = 1\ ns$), i.e., it takes 1 ns to reconfigure the interconnect. Figure 8 shows the mean delay in a system under different reconfiguration penalties.

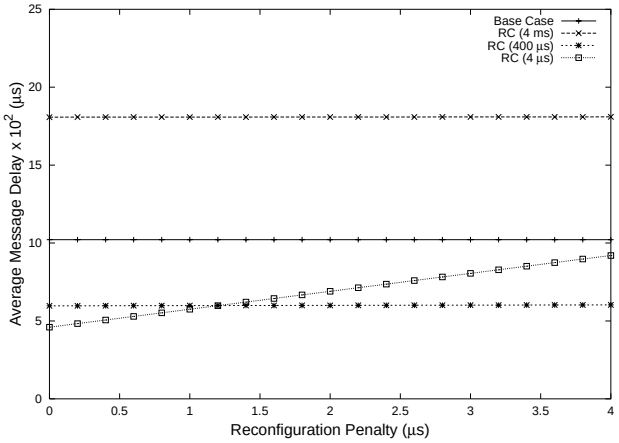

Figure 8. Average packet delay with reconfiguration cost (using Little's law)

Figure 8 shows the effect of adding reconfiguration cost to the mean delay numbers. This figure corresponds to reconfiguration times in the range of $0 - 4\ \mu s$, which is a good estimate for the time to reconfigure the system. It is shown that a system with $4\ ms$ or $400\ \mu s$ reconfiguration penalty has almost zero slope for the entire range. This does not hold for the $4\ \mu s$ reconfiguration period, as there are large numbers of reconfigurations of the system. The benefits obtained by reconfiguring the system frequently is

consumed by the overhead cost associated with those reconfigurations. As shown in Figure 8 the $4\ \mu s$ reconfiguration case gives the same performance as the $400\ \mu s$ case at around $1.2\ \mu s$ penalty for reconfiguration and becomes worse at higher penalties. It tends to approach the uniform allocation case at around $4\ \mu s$ penalty.

5. Summary and Conclusions

This paper investigates the performance implications of dynamically reconfiguring a router switch fabric constructed using optical chip-to-chip communication. A simulation-based performance study explores the question of setting an appropriate reconfiguration period for the system, as well as quantifying the performance benefits that can be expected due to reconfiguration.

The speedup of the system, which is measured as the ratio of the average packet delay under uniform allocation to the reconfigured case, ranges from 0.57 to 2.22. A speedup less than unity for the $4\ ms$ reconfiguration case demonstrates that a poorly chosen reconfiguration period can have undesirable effects on system characteristics. The speedup of 1.71 for the $400\ \mu s$ reconfiguration period illustrates the clear potential for overall improved performance due to reconfiguration without the need to unduly burden the system with frequent reconfiguration operations. The mean queue length improvement of 72% and dramatic improvement in delay standard deviation for this case are further evidence of the appropriateness of this reconfiguration period.

The use of optical chip-to-chip communication enables the construction of a network router switch fabric that can support aggregate throughputs of 1 Tb/s. The ability to reconfigure the fabric enables one to utilize the bandwidth resources even more effectively.

6. Acknowledgements

This research is supported in part by DARPA under grant DAAL01-98-C-0074 and the NSF under grants CCR-0217334 and ACI-0203869. The authors would also like to thank Ch'ng Shi Baw and Abhijit Mahajan for developing the infrastructure for modeling a reconfigurable multiring system.

References

[1] A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, B. Schieber, and M. Sudan. Efficient routing in optical networks. *Journal of the ACM*, 43(6):973–1001, 1996.

[2] R. Chamberlain, Ch'ng Shi Baw, M. Franklin, C. Hackmann, P. Krishnamurthy, A. Mahajan, and M. Wrighton. Evaluating the performance of photonic interconnection networks. In *35th Annual Simulation Symposium*, April 2002.

[3] R. Chamberlain, M. Franklin, and P. Krishnamurthy. Optical network reconfiguration for signal processing applications. In *Proc. of the IEEE International Conference on Application-Specific Systems, Architectures and Processors*, pages 344–355, 2002.

[4] M. Gandluru. Optical networking and dense wavelength division multiplexing (DWDM). Technical report, Ohio State University, 1999.

[5] P. Krishnamurthy. Reconfiguration in an optical multiring interconnection network. Master's thesis, Washington University, Saint Louis, MO, 2002.

[6] A. Mahajan. Performance analysis of an optical interconnection network. Master's thesis, Washington University, Saint Louis, MO, 2000.

[7] M. Marsan et al. All-optical WDM multi-rings with differentiated QoS. *IEEE Communications Magazine*, pages 58–66, Feb. 1999.

[8] R. K. Pankaj. *Architectures for linear lightwave networks*. PhD thesis, Massachusetts Institute of Technology, 1992.

[9] K. Park, G. Kim, and M. Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. Technical Report 1996-016, Boston University, 1996.

[10] D. Plant, M. Venditti, E. Laprise, J. Faucher, K. Razavi, M.Chateauneuf, A. Kirk, and J. Ahearn. 256-channel bidirectional optical interconnect using VCSELs and photodiodes on CMOS. *IEEE Journal of Lightwave Technology*, 19(8):1093–1103, Aug 2001.

[11] M. Shreedhar and G. Varghese. Efficient fair queueing using deficit round robin. In *Proc. of SIGCOMM*, pages 231–243, Aug. 1995.

[12] TeraConnect Inc. Two Dimensional Opto-Electronics (DOE) - Shattering the Bandwidth Bottleneck, White Paper, 2002.

[13] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Transactions on Networking*, 5(1):71–86, 1997.