Reducing Communication Latency with Path Multiplexing in Optically Interconnected Multiprocessor Systems

Chunming Qiao

Department of Elec. & Comp. Engg
State University of New York
Buffalo, New York 14260
qiao@photon.eng.buffalo.edu

Rami Melhem

Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260
melhem@cs.pitt.edu

Abstract

A physical link can be time-multiplexed to create several *time slots*, each of which corresponding to a virtual link. A conventional approach establishes a connection along a path using a set of *independent* time slots (or virtual links) and thus requires the use of switching devices capable of interchanging time slots. This paper proposes a different approach to all-optical *Time Division Multiplexed* (TDM) communications in multiprocessor systems. The idea is to establish a connection along a path using a set of time slots (or virtual links) that are *dependent* on each other, so that no time-slot interchanging is required. It is found that, despite of the possibility that establishing a connection may take a longer time, the proposed approach will result in lower *overall* communication latency as it eliminates the delays introduced by the time-slot interchanging switching devices.

1 Introduction

Optical interconnects offer many advantages over its electronic counterpart including high connection density and relaxed bandwidth-distance product [1, 2, 3]. They have already been considered in the design of both the *Touchstone* and the *CM-2* systems to reduce the wiring density and to increase the system scalability [4, 5]. This paper focuses on *Time-Division Multiplexed* (TDM) communications in optically interconnected multiprocessor systems as a way to achieve significant advancement in performance beyond the state-of-the-art.

Time-Division Multiplexed (TDM) communication techniques are useful in creating multiple *virtual links*, each of which corresponds to a *time slot* on a single *physical* link in interconnection networks. Having virtual channels increases bandwidth utilization and facilitates *adaptive routing algorithms* as well as the mapping of applications [6, 7, 8]. *TDM* techniques are also useful for tolerating the propagation latency in optically interconnected multiprocessor systems [9, 10, 11, 12], and for reducing the control complexity of channel allocation in TDM systems [13, 14] and in *Wavelength Division Multiplexed* (WDM) systems [15, 16].

A key issue to be addressed in optically interconnected multiprocessor systems is to reduce the *communication latency* which is a performance bottleneck in such systems. In this paper, we describe a new multiplexing approach for establishing all-optical connections in multiprocessor systems and compare it with a conventional multiplexing approach.

The paper is organized as follows. In Section 2, we briefly describe the general concept of establishing connections in multiplexed dynamic and direct networks [17]. In such networks, each link is time multiplexed with multiple time slots (i.e. virtual links) and a connection is established during a set of time slots, one on each link that forms a path.

Section 3 describes a conventional approach which we call *Link Multiplexing* (or LM). Using LM, a connection may be established by selecting a time slot on one link *independently* of the time slots selected on other links forming the same path. Thus, in order to transfer messages between two possibly different time slots, the switches in the network are required to have the capability of interchanging time slots.

In Section 4, we describe the proposed approach called *Path Multiplexing* (or PM). Using the proposed approach, the time slots selected on the links forming the path are *dependent* on each other, such that no time-slot interchanging is needed to transfer message along a path. For example, under special circumstances, all the time slots on the links forming the path would be the same.

We compare the two approaches in terms of overall communication latency, whose components are examined in Section 5.1. Intuitively, given a network containing a set of existing connections, it is more likely that a set of *independent* time slots, rather than a set of time slots that are *dependent* on each other, is available for establishing a new connection. Therefore, the conventional *LM* may result in lower *connection-establishment* latency. However, time-slot interchanging in LM will introduce delays as a part of network propagation latency. Thus, PM may reduce the *overall* communication latency, as well as the hardware and control complexities, due to the elimination of time-slot interchanging.

In the rest of Section 5, we present both analytic and simulation results which show that, for the range of parameter values we considered, the proposed PM approach reduces the overall communication latency when compared to the LM approach. Thus, PM may be used in the design of high performance, low complexity optically interconnected multiprocessor systems. We summarize this paper in Section 6.

2 Establishing Connections

We consider a network which consists of switches having a fixed number of inputs and outputs. Figure 1 shows an example with a mesh-like topology. Each switch in this network has four pairs of *external* links to adjacent switches and one pair of *internal* links to a processing element (PE). Each of these links can be time-multiplexed to create multiple virtual links in multiple time slots.

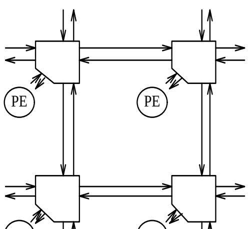


Figure 1: A direct network with a mesh-like topology

Processors communicate with each other by sending and receiving messages, which can be done via either *circuit-switching* or *packet-switching*. In this paper, we study *circuit-switching* which, despite of its shortcomings such as setup delays and potential waste of bandwidth, is a viable alternative to packet-switching in optically interconnected multiprocessor systems for the following reasons. First, with optical interconnects replacing electronic interconnects, packet-switching would require conversions between electronic and optical signals at intermediate switches for address decoding purposes. Second, with the advances in optical communications technology and especially in WDM technology, bandwidth is no longer a performance bottleneck of a multiprocessor system, rather, it is the control and communication latency that will be the limiting factors. Last but not the least, a recent technological advance in telecommunications as well as computer communications is the introduction of *Asynchronous Transfer Mode* (ATM), which requires the establishment/release of *virtual channel connections* (VCCs) and supports connection-oriented services required by certain applications.

Both the LM and PM approaches discussed in this paper are *variations* of circuit-switching. More specifically, when each physical link in a network is *time-multiplexed* with K virtual links, it means that on each link, the infinite time horizon is divided into intervals of K time slots. This interval is referred to as a *frame* and K is called the frame size, or *multiplexing degree*. Accordingly, a connection consists of a set of time slots, one on each link that forms a path from a source to a destination. For a *given link*, if time slot i $(0 \leq i \leq K-1)$ of a frame is used to establish a connection, then for the duration of the connection, the same time slot of the *next frame* is also used for that connection. However, different time slots may be used on *different links*.

Common to both approaches studied in this paper is that a connection needs to be established before a source processor can send a message to a destination processor. Connections can be established under either centralized or distributed control as in [18, 13].

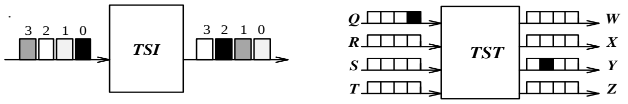
Once a connection is established during one in every K time slots on each link, the source processor divides its message into several fixed sized data packets whose length is equivalent to the duration of a time slot. These packets are sent one after another in every K time slots. That is, one packet is sent in every *frame*. Clearly, global synchronization is required for bookkeeping information on the usage of the network resources, for sending and receiving data packets at both ends of a connection, and for switching from one (virtual) link to another (virtual) link along the connection.

In the following sections, we give detailed descriptions and examples of how a connection is established using the conventional *Link Multiplexing* (LM) and the proposed *Path Multiplexing* (PM) approaches.

3 Link Multiplexing (LM)

One way to establish a connection in a direct network multiplexed with degree K is the so-called *Link Multiplexing* (LM) approach. More specifically, a connection may be established over a set of *independent* time slots along a path. This, however, calls for the use of *Time Slot Interchangers* (TSIs) in order to transfer the information carried over the same connection from one time slot to another along the path.

A TSI capable of interchanging K time slots can be viewed as a "black box" having one input and one output, each being multiplexed with K virtual channels, or equivalently, carrying frames of K time slots each. In Figure 2(a), K is assumed to be 4 and thus an input and output frame have 4 time slots each. The TSI *switches* time slots 0, 1, 2 and 3 of the input frame to time slots 2, 0, 3, 1 of the output frame, respectively. By setting the TSI properly, any time slot of the input frame can be switched to any *idle* time slot of the output frame (i.e. any time slot to which no other time slots of the input frame has been switched).



(a). A Time Slot Interchanger (TSI). (b). A Time-Space-Time (TST) switch.

Figure 2: TSI and TST switches

A more generalized form of a TSI is the so-called *Time-Space-Time* (TST) switch which has $S \geq 1$ inputs and outputs. When $S = 1$, a TST switch becomes a TSI. When $S > 1$, a TST switch may be viewed as an *augmented* $S \times S$ switch with each input and output being multiplexed with degree K ($K > 1$). By setting a TST properly, any time slot of any input can be switched to any idle time slot of any output. Figure 2(b) shows a TST with $S = K = 4$, in which time slot 0 of input Q is switched to time slot 2 of output Y.

We now illustrate the *LM* approach using an example. As shown in Figure 3, each switch in a network is required to be a TST switch. Assume that a connection from a source to a destination needs to be established on two physical links X and Y. From the viewpoint of switch α (that can interconnect links X and Y), the message may arrive during any time slot i on X and then leave during any time slot j on Y, where $1 \leq i, j \leq K$, as long as these two time slots are available (that is, they are not being used by any other connections). This is because switch α can interchange time slots. Therefore, when establishing the connection using LM, time slot i and j will be selected *independently* on link X and Y, respectively.

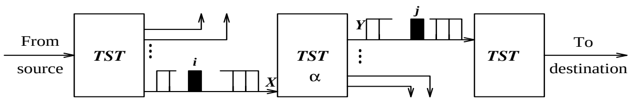


Figure 3: Establishing a connection with LM

A TSI (or TST) can be implemented either electronically or optically. An electronic implementation of a 4×4 TSI is shown in Figure 4. An Optical-to-Electronic (O/E) conversion circuit (i.e. receiver) at the input side converts the incoming optical signals to electronic ones. The converted signals from time slots 0, 1, 2 and 3 are then stored in input buffers 0, 1, 2 and 3, respectively. The 4×4 crossbar switch between the input buffers and output buffers is then set such that the content of a given input buffer is copied to a desired output buffer (e.g. Fig. 4 is an implementation of the TSI in Fig. 2 (a)). Finally, an Electronic-to-Optical (E/O) conversion circuit (transmitter) transmits the content of the output buffers 0, 1, 2 and 3 in that order.

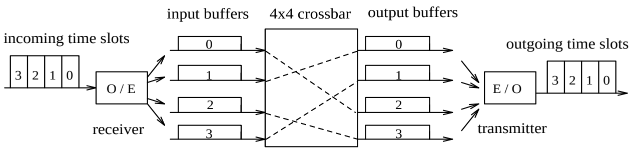


Figure 4: An electronic implementation of a TSI

One of the drawbacks of an electronic implementation of a TSI (or TST) is the need for the expensive high speed receiving and transmitting circuits inside each switch, which also lead to its nontransparency to bit-rate. For a direct network consisting of $S \times S$ TSTs, in addition to one pair of receiving and transmitting circuits that is needed at the processor-to-network (or processor-to-switch) interface, another $S - 1$ pairs are needed inside each switch, one for each input-ouput link. Since these circuits are designed for a specific receiving and transmitting speed (say a 1Gb/s), one will have to replace all of them for *higher* bit-rate transmissions (say at 2 Gb/s). Another drawback that is of special interest to us in this paper is the delay introduced in the process of interchanging time slots, which is at least as long as K time slots.

A TSI (or TST) can also be implemented *optically* using fiber-loop delay lines and Lithium Niobate directional couplers [19, 20, 21]. Such an optical implementation eliminates the need for E/O and O/E conversions for switching and therefore can provide *bit-rate* *transparency*. However, optical TSTs are not without their drawbacks. For one thing, they, like electronic ones, will introduce delays as a part of propagation latency of a network, in addition to control and hardware complexity. Moreover, optical TSTs will introduce additional crosstalks and attenuations of optical signals. While all these undesirable effects may not significantly affect long-distance telecommunications applications, and certainly do not undermine the advantages of optical TSTs over their electronic implementations, these undesirable effects could be detrimental to the performance of optically interconnected multiprocessor systems. In the next section, we describe an approach that can establish *all-optical* connections without using TSTs.

4 Path Multiplexing (PM)

Before we describe the proposed approach in details, we first review a connection paradigm, called *Reconfiguration with TDM* (RTDM), which was previously developed for indirect networks [13, 14].

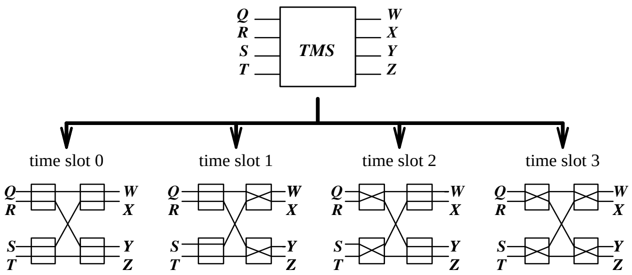


Figure 5: Connecting inputs to outputs in TMS

Using the RTDM, an indirect network, such as a crossbar, behaves like a *Time-Multiplexed Switch* (TMS). More specifically, the network repeatedly goes through a sequence of configurations, one for each time slot. Figure 5 illustrates a possible sequence of four (4) configurations that a 4×4 crossbar switch may go through. Like the TST switch shown in Figure 2(b), each input and output is multiplexed with degree $K = 4$. However, unlike that TST, a time slot i of any input link is switched to the same time slot i of an output link, as shown in the Figure. Such a TMS does not have the capability of interchanging time slots as a TST switch does. It is important to note that, as a result, the output, which the signals arriving at time t at a given input will go to, depends solely on the *switch setting* at time t.

Just as shown in the example, for an $S \times S$ switch, all possible S^2 connections can be established by multiplexing the switch with degree $K = S$ [22, 23]. However, this may result in an unnecessarily long latency because a given input-output connection is established once every S time slots. *RTDM* provides efficient multiprocessor communications through either *static* [13] or *dynamic* [14] reconfiguration which reduces the multiplexing degree K by establishing only those required connections. Note that, one can easily adapt K in a TMS according to application requirements [13, 14], whereas in an optical TST, K is fixed by its architecture.

The proposed *Path Multiplexing* (or PM) approach extends the principle of RTDM by establishing a connection in a direct network consisting of TMS switches, each of which is multiplexed with degree K.

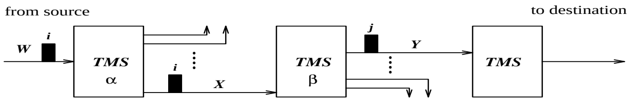


Figure 6: Establishing a connection with PM

Figure 6 illustrates the idea of the PM approach. Let us assume that a connection needs to be established along links W, X and Y, and that switch α can be set to interconnect link W with link X during time slot i only ($0 \leq i < K$). Consider first the case in which the propagation delay on link X is *negligible* when compared to the duration of a time slot. Since any message carried on the connection will arrive at switch β during time slot i, we have to be able to set switch β so that link X is interconnected with link Y during time slot i also. (in the figure, this means $j = i$). If the link propagation delay on X is *not* negligible (say, equal to L_P time slots), then the connection can be established only if during time slot $j = (i + L_P)\ mod\ K$, switch β can be set so that link X is interconnected with link Y. In either case, the time slot to be used on link Y (which is j) is dependent on the time slot used on link X (which is i). In general, when using the PM approach, the time slots on the links along the path are selected in such a way that packets can flow through the links without being buffered, or in other words, no time slot interchanging is required.

The need to find such a set of time slots in the network may result in a longer connection-establishment latency in PM than in LM. On the other hand, since PM uses the TMS switches that do not introduce delays while LM uses the TSTs switches that do, the overall communication latency in the PM approach may be smaller than that in the LM approach. In the next section, we first examine the components of communication latency and then compare the PM and the LM approaches via analysis and simulations.

5 Comparing PM with LM

Since TST switches have higher control and hardware complexity than TMS switches, it is more expensive to adopt the LM approach especially if it uses electronic TSTs, than the proposed PM approach. In addition, when it comes to choosing an appropriate multiplexing degree for an application, the PM approach is much more flexible than the LM approach especially if the latter uses optical TSTs. Complexity and flexibility aside, what we will focus on in this comparative study, however, is the performance of these two approaches in terms of the communication latency they will result in.

5.1 Communication Latency

In this section, we first examine the components of communication latency. In the following discussions, we will use the duration of a time slot as a *unit* of time. In a TDM environment, it is common to let the duration of a time slot equal to the transmission time of a packet plus some *guard bands* for synchronization purposes. For example, if the transmission rate is 1 *Gb/s* and a packet has 512 bits, then it will take 512 *ns* to transmit the packet. Assuming that the guard bands of 50 *ns* are used, then a time slot would be 562 *ns*. Note that, many factors including the transmission technology as well as applications will determine the optimal value of the duration of a time slot.

In circuit-switching, the communication time includes both message-transmission time and *communication latency*. If the message is divided into m packets, then the time required for transmitting the message into the network is Km time slots since each of these packets is transmitted in every K time slots (i.e. one frame).

The communication latency, on the other hand, is the sum of two component values: *connection-establishment* latency and *network-propagation* latency. The connection-establishment latency, denoted by L_E, is the time required for the connection to be established in the network. The *network-propagation* latency, denoted by L_N, is the time for a signal to traverse the path from a source to a destination. To summarize, the communication latency, denoted by L, can be calculated as

$$L = L_E + L_N \qquad (1)$$

Note that, L_N is further composed of two parts: *link propagation* latency and *switching* latency. The first part, the link propagation latency, denoted by L_P, is defined to be the time for the signal to traverse an external link. This latency is often *negligible* in a multiprocessor system (relative to the duration of a time slot). For instance, since processors are usually only a few feet away, the link propagation latency is only a few nanoseconds while a time slot may be several hundred nanoseconds. The second part, the switching latency, denoted by L_S, is defined to be the time for a message to go through a switch. Assuming that the average distance between a source and a destination is H hops, the total switching latency along the path, excluding the source or destination point, is $(H-1) \times L_S$. Hence, by neglecting the link propagation latency L_P, Eq. 1 can be modified to

$$L = L_E + (H-1) \times L_S \qquad (2)$$

Note that, if PM is used, then L_S is also negligible. However, as discussed in Sec 3, a TST switch capable of interchanging time slots introduces a delay, which may be K time slots. Thus, L_S is *non-negligible* if LM is used.

If we let the average *connection latency* in PM and LM be $L_E(PM)$ and $L_E(LM)$ respectively, and also denote the *communication latency* in the two approaches by $L(PM)$ and $L(LM)$, respectively, we have

$$L(PM) = L_E(PM) \qquad (3)$$

$$L(LM) = L_E(LM) + (H-1) \times K \qquad (4)$$

Throughout our comparative study of the LM and PM approaches, we will focus on the value of $L(PM)$ as defined in Eq. 3 and the improvement ratio, I, which is defined as follow :

$$I = \frac{L(LM) - L(PM)}{L(LM)} \times 100\% \qquad (5)$$

Hence, the larger the I is, the better PM becomes when compared to LM.

Our comparative study of the PM and LM approaches involves two parts: analytical work and simulation. In both cases, a direct network whose links are time-multiplexed with degree K is studied and the communication latency resulted from LM and PM is evaluated. Common assumptions to both analyses and simulation that simplify our evaluation processes have been made. These assumptions include

1. a *shortest path* is used to establish a given connection (even though alternate paths may exist).
2. a rejected request is resubmitted after t time slots ($t \geq K$).

3. the destination of each message is randomly chosen among all the processors.

Additional assumptions will be stated in the following two sections that describe analytical and simulation results, respectively.

5.2 Probability Analysis

We first calculate the probability that a given connection can be established along a path, which will then be used to calculate L_E. In order to simplify the analysis, it is further assumed that

4. the network is *symmetric* and consists of $S \times S$ switches. An example is a k-ary n-cube, where $k, n \geq 2$.
5. the network is fully loaded. That is, during each time slot, every input of a switch needs to be connected to an output of that switch.

We first consider the case in which there is no time-multiplexing and hence, a switch can be treated as an $S \times S$ crossbar. Accordingly, if multiple inputs compete for a common output link (say X), the probability that a given input will successfully be interconnected to X is

$$P_a = 1 - (1 - 1/S)^S \tag{6}$$

which is approximately 0.67 when $S \geq 5$. In the following discussions, P_a will be referred to as the probability that output X is *available* to a given input.

We will use the notation $PA[H, K]$ (the success probability) or $\overline{PA}[H, K]$ (the failure probability which equals $1 - PA[H, K]$) to denote the probability that a given connection *can* or *cannot* be established between a source and a destination, which are H hops away when the multiplexing degree is K. In a trivial case where $K = 1$ (which is equivalent to having no time-multiplexing), the LM and the proposed PM approach are not much different and will result in the same $PA[H, 1]$. More specifically, since a connection can be established only if all the H external links along the path are available, we have

$$PA_{LM}[H, 1] = PA_{PM}[H, 1] = {P_a}^H \tag{7}$$

When the multiplexing degree $K > 1$, PM and LM approach will result in different success (and failure) probabilities. Consider first the case in which PM is used. We may begin with the probability that *none* of the K time slots is available on *all* the H links. Based on Eq. 7, this probability is

$$\overline{PA}_{PM}[H, K] = (1 - PA_{PM}[H, 1])^K = (1 - {P_a}^H)^K$$

Since a connection can be established as long as there is one out of K time slots such that the same time slot is available along all the H links, we have

$$\begin{aligned} PA_{PM}[H, K] &= 1 - \overline{PA}_{PM}[H, K] \\ &= 1 - (1 - {P_a}^H)^K \end{aligned} \tag{8}$$

We now consider the case in which LM is used. For *a given link*, the probability that *none* of the K time slots is available is $(1 - P_a)^K$, thus the probability that at least one time slot is available on that given link is $1 - (1 - P_a)^K$. Since a connection can be established as long as on each of the H links along the path, one of the K time slots is available, we have

$$PA_{LM}[H, K] = (1 - (1 - P_a)^K)^H \tag{9}$$

where P_a is about 0.67 as given by Eq. 6.

Assuming that p (which is either PA_{PM} or PA_{LM}) is the success probability and t is the resubmission interval after each failure, then a connection request will experience a latency of 1 with probability p, a latency of $(t + 1)$ with probability $p(1 - p)$, and a latency of $(2t + 1)$ with probability $p(1 - p)^2$, etc. Therefore,

$$\begin{aligned} L_E &= p + p(1 - p)(t + 1) + p(1 - p)^2(2t + 1) + \ldots \\ &= p \times \sum_{i=0}^{\infty} (1 - p)^i (i \times t + 1) \end{aligned} \tag{10}$$

which results in

$$L_E = 1 + \frac{(1 - p)t}{p} \tag{11}$$

Therefore, we may calculate the connection latency $L_E(PM)$ and $L_E(LM)$ by

$$\begin{aligned} L_E(PM) &= 1 + \frac{(1 - PA_{PM}[H, K])t}{PA_{PM}[H, K])} \\ L_E(LM) &= 1 + \frac{(1 - PA_{LM}[H, K])t}{PA_{LM}[H, K]} \end{aligned} \tag{12}$$

where $PA_{PM}[H, K]$ and $PA_{LM}[H, K]$ can be substituted using Eqs. 8 and 9, respectively. Finally, we can determine the communication latency $L(PM)$ and $L(LM)$ according to Eq. 3 and Eq. 4, respectively, as well as the improvement ratio I according to Eq. 5.

Assuming $K = 4$ and $t = 4$, Table 1 shows some of the results obtained through the above probability analysis. From Table 1, one can see that in all cases, PM results in reduced communication latency when compared to LM. Note that because the difference in

L_N between the two approaches is *linear* to H (see Eqs.3 and 4), whereas the difference in L_E between the two approaches is *exponential* (see Eqs. 8, 9 and 12), the improvement ratio decreases with H.

H	2	4	6	8
PA_{PM}	0.9095	0.5983	0.3205	0.1563
PA_{LM}	0.9769	0.9545	0.9325	0.9110
$L(PM)$	1.39	3.68	9.48	21.60
$L(LM)$	5.09	13.19	21.29	29.39
I	74.5%	72.0%	55.5%	26.5%

Table 1: Analytic Results

5.3 Simulation Study

In our comparative study, simulations have also been conducted to evaluate the LM and PM approaches. In addition to the common assumptions (1 - 3) made above, the simulation model is based on the following set of assumptions:

$4'$. the network is a two dimensional $N \times N$ mesh.
$5'$. each PE has a message buffer of size b. During each time slot, a PE will generate r messages ($r < 1$) unless the message buffer is full.
$6'$. each message is composed of m packets.

The simulation program is written in C using the libraries from the YACSIM [24]. Statistical measures on the connection latency resulted from PM and LM are collected with a confidence level no less than 90% and a confidence interval no larger than 0.1. In addition, measures on the average distance *in hops* between a source and a destination of all the connections established is also collected. This would be the value of H in Eq. 4, which, together with the multiplexing degree K, can be used to determine $L(LM)$ and therefore the improvement ratio I. This H, for example, is about 6.6 and 13.3, respectively, in the 10×10 and 20×20 meshes simulated.

The simulation results presented in Figures 7 through 11 are obtained by limiting the message rate r not to exceed a certain point, beyond which the network will be saturated as indicated by a steady network throughput. In practice, the actual traffic applied to the network would (and should) not exceed a point where the network is saturated. Our simulation results have indicated that within the range of parameter values we considered, the PM approach will always result in lower communication latency than the LM approach.

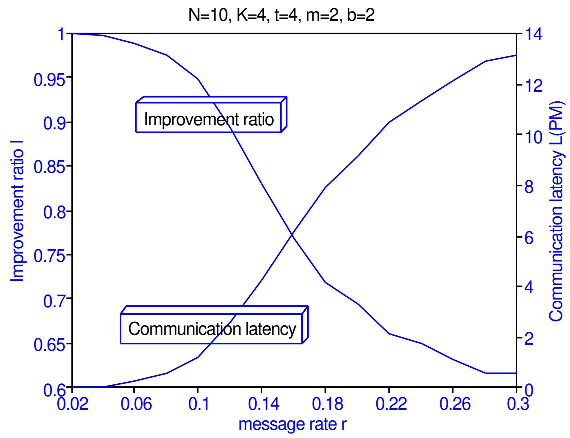


Figure 7: Simulation results for different message rates

In Figure 7, both improvement ratio I and communication latency in PM, $L(PM)$, are shown. These results are obtained using the following parameter values: the network size $N = 10$; the multiplexing degree $K = 4$; resubmission interval $t = 4$ time slots; each message contains $m = 2$ packets and each processor has a message buffer of size $b = 2$. Our simulation results indicated that the network is near saturation when message rate r reaches 0.3. At this point, the improvement ratio and communication latency in PM approach 60% and 13 time slots, respectively. At a low message rate (e.g. $r = 0.02$), the improvement ratio is almost 100%. This occurs because at a low message rate, $L(PM)$ is near 0 while $L(LM)$ is dominated by the switching latency, which is about 22 time slots. We also observe that the improvement ratio decreases with message rate. This is because while the difference in the network propagation delay L_N between PM and LM remains almost constant (at about 22 time slots), the difference in the success probability between PM and LM, and hence the difference in the connection-establishment latency L_E between the two approaches, increases with the network traffic load.

In Figure 8, the effect of buffer size, b, on both improvement ratio and communication latency in PM is shown. It can be seen that in all cases, the improvement ratio is above 50%, although for a given message rate, the improvement ratio decreases with b. This is consistent with the previous observation because a larger value of b translates into a heavier traffic load for the same message rate. We have also studied the effect of various message lenghth m, and as expected, improvement ratio decreases when m increases, which results in an effectively heavier traffic

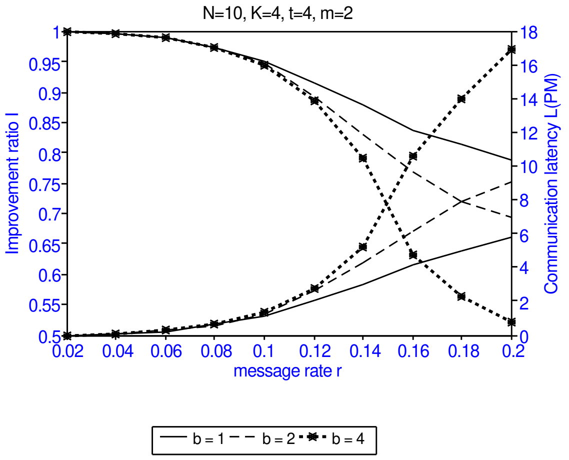


Figure 8: Effect of buffer size b

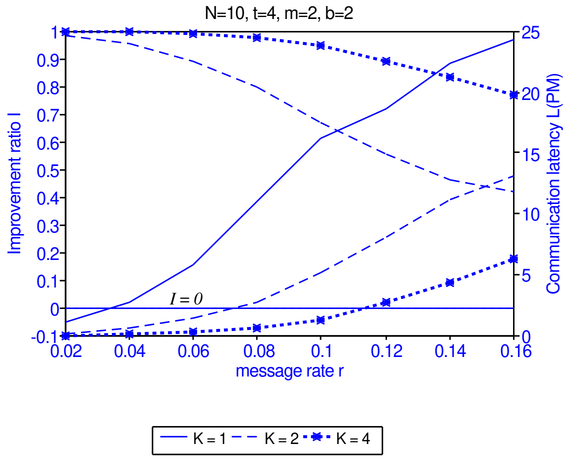


Figure 10: Effect of multiplexing degree K

load for a given message rate. Note that, when total *communication time* is considered, a larger m results in a lower amortized communication latency because of increased message-transmission time.

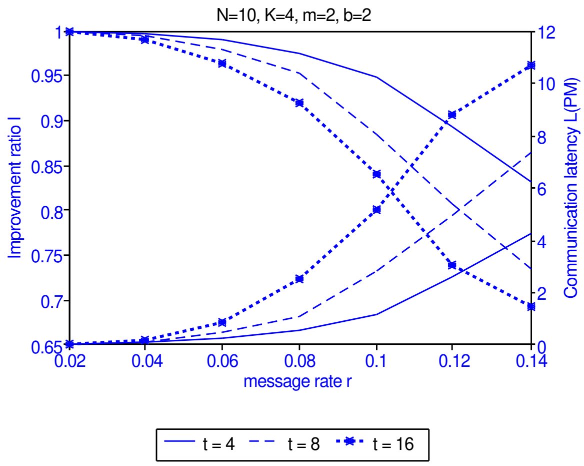


Figure 9: Effect of resubmission interval t

Figure 9 shows that the improvement ratio decreases with t, although it is about 70% at $r = 0.14$ even for the case $t = 16$. This is due to the fact that one-time failure probability in establishing a connection is higher in PM than in LM (see Eqs. 8 and 9) and thus, a larger t penalizes PM more than LM. Put in another way, a larger t magnifies the disadvantage of PM when compared to LM in establishing a connection (see Eq. 12).

Figure 10 shows that the improvement ratio increases with K. When $K = 1$, PM is the same as LM, resulting in $I = 0$ (the horizontal line). Our simulation results have indicated that the network can tolerate a heavier traffic load with a larger K. In this regard, a larger K helps PM as much as LM in reducing the connection-establishment latency L_E. However, a larger K means a larger switching latency and hence a larger network latency in LM, thus resulting in a larger improvement ratio I.

In order to investigate the effect of network size N and *decouple* it from the effect of traffic load, we define *normalized* message rate R to be such that the average number of messages generated in the *network* during each time slot is $R \times N$. Because each message will need to travel an average distance on the order of N from its source to its destination, and the number of links in the network is of order N^2, using the normalized message rate would create a *per link* load of $c \times R$, where c is a constant. Note that, given R, $r = R/N$.

Our simulation results have indicated that although the absolute value of the communication latency in both PM and LM increases with N, their difference also increases due to the increase in the average distance between a source and a destination H, which is linear to N. Therefore, as can be seen from Figure 11, with the same normalized message rate R, the improvement ratio in networks of different sizes is about the same. It is worth noting that because of the finite buffer size b, having the same R in networks of different sizes can not guarantee that the *actual* load is also the same.

To summarize, our simulations have shown that PM is effective in reducing the communication latency in a wide range of practical situations. The proposed PM approach applies especially well, as indicated by

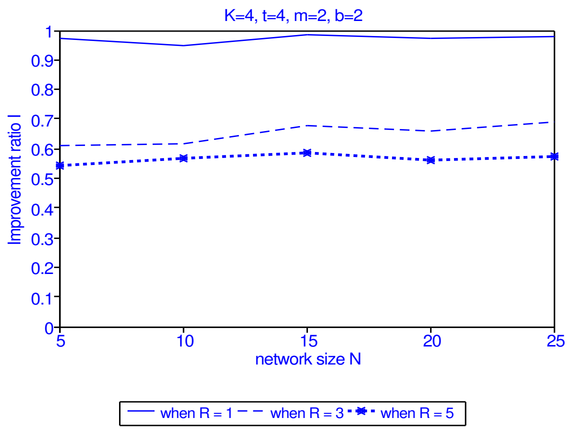


Figure 11: Effect of network size N

a high improvement ratio, to networks having a large multiplexing degree and a reasonable traffic load.

6 Concluding Remarks

With tremendously high bandwidth available in optics, it becomes increasingly important to reduce the communication latency which ultimately limits the performance of the multiprocessor systems that utilize optical interconnects. In this paper, we have proposed an approach called *Path Multiplexing* (PM) as an alternative to the conventional *Link Multiplexing* (LM) approach for establishing connections in networks where *Time-Division Multiplexing* (TDM) techniques are used to created virtual channels. We studied the issues related to the implementation of both approaches and compared the two in terms of the resulting communication latency through analysis and simulations.

The conventional LM approach uses switches that are capable of interchanging time slots while the PM approach does not. While the LM approach may result in lower connection-establishment latency due to its flexibility in selecting time slots for establishing a connection, both hardware and control complexity of the network is high. In addition, because interchanging time slots introduce additional delays, the proposed PM approach may reduce the *overall* communication latency.

We note that there are two circuit-switching based *wavelength routing* approaches in a *Wavelength Division Multiplexed* (WDM) system that are analogous to the PM and LM approach. One approach, being analogous to the PM approach, establishes a connection using a single wavelength [25, 26] and the other, being analogous to the LM approach, establishes a connection using multiple wavelengths and thus requires wavelength converters [27]. The results on blocking probabilities of the two time-domain approaches we have obtained agree in principle with those of the two analogous wavelength-domain approaches [26, 27, 28]. However, it is worth noting that the issues regarding communication latency, which relate to, but differ from, those regarding blocking probability, have not been discussed in the WDM domain. As a future work, we will study how a scheme that combines distributed time slot assignment [13] and adaptive routing algorithms will affect the communication latency of the PM and LM approaches. In addition, the effect of non-negligible link propagation latency which is present in distributed systems such as LANs and WANs will also be taken into considerations in the comparative study. The results obtained from these studies shall shed a light on the communication latency issues related to wavelength routing under distributed control in WDM systems.

Acknowledgements

This work is supported in part by a grant from NSF under contract number MIP-9409864 to the first author. The authors would like to thank an anonymous reviewer whose detailed comments helped improve the quality of this paper.

References

[1] M. Feldman, S. Esener, C. Guest, and S. Lee, "Comparison between optical and electrical interconnects based on power and speed considerations," *Applied Optics*, vol. 29, pp. 1742–1751, May 1990.

[2] J. Goodman, F. Leonberger, S. Kung, and R. Athale, "Optical interconnections for VLSI systems," *IEEE Proc.*, vol. 72, pp. 850–866, 1984.

[3] P. Lalwaney, L. Zenou, A. Ganz, and I. Koren, "Optical interconnects for multiprocessors: cost peformance analysis," in *Proceedings of the Symposium on Frontiers of Massively Parallel Computation*, pp. 278–285, Oct. 1992.

[4] R. Floren et al, "Optical interconnects in the Touchstone supercomputer program," *SPIE Proc. Intergrated optoelectronics for communication and processing*, vol. 1582, pp. 46–54, Oct. 1991.

[5] B. Kahle, E. Parish, T. Lane, and J. Quam, "Optical interconnects for interprocessor communications in the Connection Machine," in *IEEE Conference on Computer Design*, Oct. 1989.

[6] W. Dally and C. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Transactions on Computers*, vol. 36, pp. 547–553, May 1987.

[7] C. Jesshope, P. Miller, and J. Yantchev, "High performance communications in processor networks," in *16th International Symposium on Computer Architecture*, pp. 150–157, 1988.

[8] L. Ni and P. McKinley, "A survey of routing techniques in wormhole networks," Tech. Rep. MSU-CPS-ACS-46, Michigan State University, Department of Computer Science, Oct. 1991.

[9] Z. Guo et al., "Array processors with pipelined optical busses," *J. of Parallel and Distributed Computing*, vol. 12, no. 3, pp. 269–282, 1991.

[10] H. Jordan, "Time multiplexed optical computers," in *Proceedings of Supercomputing*, pp. 370–378, Nov. 1991.

[11] R. Melhem, D. Chiarulli, and S. Levitan, "Space multiplexing of waveguides in optically interconnected multiprocessor systems," *The Computer Journal*, vol. 32, no. 4, pp. 362–369, 1989.

[12] C. Qiao and R. Melhem, "Time-division optical communications in multiprocessor arrays," *IEEE Transactions on Computers*, vol. 42, pp. 577–590, May 1993.

[13] C. Qiao and R. Melhem, "Reconfiguration with time-division multiplexed MINs for multiprocessor communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 5, no. 4, pp. 337–352, 1994.

[14] C. Qiao, R. Melhem, D. Chiarulli, and S. Levitan, "Dynamic reconfiguration of optically interconnected networks with time division multiplexing," *Journal of Parallel and Distributed Computing*, vol. 22, pp. 268–278, Aug. 1994.

[15] P. Dowd, K. Bogineni, K.A.Aly, and J. Perreault, "Hierarchical scalable photonic architectures for high performance processor interconnection," *IEEE Transactions on Computers*, vol. 42, pp. 1105–1120, Sept. 1993.

[16] A. Ganz and Y. Gao, "A time-wavelength assignment algorithm for a WDM star network," in *Proc. of IEEE INFOCOM*, pp. 2144–2150, 1992.

[17] T. Feng, "A survey of interconnection networks," *Computer*, vol. 14, pp. 12–27, Dec. 1981.

[18] S. Nugent, "The iPSC/2 direct-connect communications technology," in *Proceedings of the 3rd Conference on Hypercube Concurrent Computers and Applications*, vol. 1, Jan. 1988.

[19] D. Hunter and D. Smith, "New architectures for optical TDM switching," *IEEE Journal of Lightwave Technology*, vol. 11, pp. 495–511, Mar. 1993.

[20] S. Ramanan and H. Jordan, "Serial array shuffle-exchange architecture for universal permutation of time slots," in *SPIE Proc., Digital Optical Computing II*, vol. 1215, pp. 330–342, Jan. 1990.

[21] R. Thompson, "Architectures with improved signal-to-noise ratio in photonic systems with fiber-loop delay lines," *IEEE Journal on Selected Areas in Communications*, vol. 6, Aug. 1988.

[22] H. Shing and L. Ni, "A conflict-free memory design for multiprocessor," in *Proceedings of Supercomputing*, pp. 46–55, Nov. 1991.

[23] R. Thompson, R. Anderson, J. Camlet, and P. Giordano, "Experimental modular switching system with a time-multiplexed photonic center stage," in *Photonic Switching, OSA Technical Digest*, pp. 212–218, Mar. 1989.

[24] J. Jump, *YACSIM reference manual, version 2.1.* Department of Electrical and Computer Engineering, Rice University, Mar. 1993.

[25] K. Bala, T. Stern, and K. Bala, "Algorithms for routing in a linear lightwave network," in *Proceedings of the IEEE Infocom*, pp. 1–9, 1991.

[26] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: an approach to high-bandwidth optical wans," *IEEE Transactions on Communications*, vol. 40, pp. 1171–1182, July 1992.

[27] K. Lee and V. O. Li, "A wavelength-convertible optical network," *IEEE Journal of Lightwave Technology*, vol. 11, no. 5/6, pp. 962–970, 1993.

[28] R. Ramaswami and K. Sivarajan, "Optimal routing and wavelength assignment in all-optical networks," in *Proceedings of IEEE Infocom*, vol. 2, pp. 970–979, June 1994.