Multiprocessor Interconnection Networks Using Partitioned Optical Passive Star (POPS) Topologies and Distributed Control

Donald M. Chiarulli
Dept. of Computer Science

Steven P. Levitan
Dept. of Electrical Engineering

Rami P. Melhem
Dept. of Computer Science

James P. Teza
Dept. of Electrical Engineering

Greg Gravenstreter
Dept. of Computer Science

University of Pittsburgh, Pittsburgh PA, 15260

Abstract

This paper presents a scalable electro-optical interconnection network architecture which is suitable for tightly coupled multiprocessors. The architecture is called a Partitioned Optical Passive Star (POPS). It is a type of multiple passive star topology in which only constant and symmetric coupler fanouts are used and in which exactly one coupler is traversed on any path through the network. Control is based on the state sequence routing paradigm which multiplexes the network between a small set of control states and defines a control operation to be a transformation of those states.

These networks have highly scalable characteristics for optical power budget, resource count, and message latency. Optical power is uniformly distributed and the size of the system is not hard limited by the power budget. Resource complexity grows with asymptotic complexity O(n) for the couplers, O($n\sqrt{n}$) for transceivers, and O($\sqrt{n}$ log(n)) for control. In this paper, we present a static analysis and a simulation of dynamic performance which demonstrates the ability of a POPS design to support 1024 nodes using current device and coupler technology.

1.0 Introduction

Among the most attractive features of reconfigurable electro-optical interconnection networks is the ability to trade off optical channel bandwidth for the wiring complexity of an equivalent electronic implementation. A variety of multiplexing methods exist to support this capability. In evaluating these designs, both the traditional measures of resource complexity and message latency must be considered as well as two additional constraints which are specific to electro-optical networks. The first constraint is the balanced distribution of optical power throughout the system. The second is the efficient control and routing of messages through the network.

This researcch has been supported in part by a grant from the Air Force Office of Scientific Research under contract numbers 89-0469 and F49620-93-1-0023DEF

Optical power constraints can limit the size of a system to the number of receiver sites that can be illuminated by a single transmitter. Often these receivers must be illuminated in unpredictable patterns and in the presence of significant losses in the transmission medium. Control efficiency is a measure of the amount of data transferred per control operation. Since in an electro-optic network the bandwidth of the electronic control circuits is typically limited, control bandwidth can become a bottleneck unless each increase in optical bandwidth is accompanied by a corresponding increase in control efficiency.

In this paper, we present an architecture which is a synthesis of a topological solution to the resource complexity and power distribution problems, and a control paradigm which exploits locality characteristics in the message traffic to amortize control overhead over multiple messages. The topology is based on a multiple passive star organization which we call a Partitioned Optical Passive Star or POPS network. The control solution is based on the state sequence routing paradigm which was originally proposed by the authors in [CLMQ,CLMQ93].

Our presentation is organized as follows. Section 2 describes the background and previous research which motivates this effort. Section 3 introduces the POPS network topology, and Section 4 describes state sequence routing. Section 5 characterizes network performance by a static analysis of the service time for random message sets. Section 6 outlines the distributed implementation of state sequence routing, and Section 7 presents simulation data on dynamic performance. Section 8 outlines our conclusions and directions for future research.

2.0 Background and Motivation

Multiple passive star networks are attractive for multiple access single-hop interconnection networks because they offer a maximum connectivity with a constant power budget[Bir93], and are simple, relatively low cost yet robust structures[BLM93]. Using multiple passive stars, it is possible to design completely reconfigurable networks without the use of active photonic switches [GLZ91]. Historically, network configurations using passive star components have been implemented using one of three multiplexing technologies[Muk92]: wavelength division multiplexing (WDM), time division multiplexing (TDM) or code division multiplexing (CDM). In general, to be a completely non-blocking network the number of available slots (wavelength, time or code) must be greater than or equal to the total number of nodes. If not, then contention resolution must be incorporated into the control.

WDM technologies allow non-blocking networks to be constructed using passive star and broadcast-and-select architectures with large numbers of wavelength slots relative to the number of nodes. However, application of WDM technology is hampered by the need to use closely spaced wavelengths to obtain the required number of slots when the network is scaled to more than a trivial number of processors. This technology is currently capable of wavelength separations on the order of 1 nm yielding a total on the order of 100 slots given a typical fiber transmission window with constraints on the total power of all wavelengths[Bra90]. In addition, it is necessary to use rapidly tunable laser sources, rapidly tunable filter receivers, or both, to obtain sufficient interconnectivity and throughput. Only if both tunable receivers and tunable transmitters are used is it possible to use a number of wavelengths less than the number of interconnected nodes. Currently the tuning speed for tunable lasers is on the order of several nanoseconds[KVG+90] and for tunable filters is on the order of several microseconds[SJ91]. Conflicts in WDM systems consist of wavelength conflicts in which two or more packets are routed using the same wavelength, and destination conflicts in which a packet may be routed on a wavelength to which a receiving node is not tuned. These conflicts can be resolved with varying success using any of several distributed control protocols [Meh90]. However, the need to use a control channel may place a limit on the network capacity imposed by the bandwidth of the control channel [Dow91].

CDM requires data bits be encoded with respect to the address of the receiver using a sequence of bits at a much higher rate than the data bit rate[PSS86, SWH90]. This requires high speed signal processing be performed at both the transmitting and receiving nodes in order to obtain usable data rates. Several implementations using optical processing have been proposed[CM92, HS92]. In addition, a set of orthogonal codes is required to uniquely address each receiver node. This limits the scale of the network to the length of the code and to the speed of the signal processing used for the encoding and decoding.

TDM systems assign fixed time slots to specific paths in order to obtain complete connectivity. For burst type traffic, both throughput and latency are relatively poor since the utilization of the slots is low and the wait time between paths is fixed. In many implementations, throughput may be sacrificed in favor of simple control. Other designs have implemented adaptive scheduling to increase channel utilization and improve performance[CL91].

In considering the combined issues of topology and control, our investigation has specifically focused on the application of optical technology which can be implemented currently or in the near term. Our designs are constrained by the following parameters.

- A totally connected network is necessary and performance goals require this connectivity be achieved in a single-hop. Therefore, a direct connection between any two nodes must not require a forwarding by another node.
- Power distribution must be uniform, and the power budget for any path must not directly limit the size of the network (number of nodes).
- Designs must be implementable with “off the shelf” components such as, fixed wavelength transmitters and receivers, and couplers of nominal fanin and fanout.

Our solution for meeting these constraints uses a multiple passive star topology with control implemented in the time domain. Multiple passive star couplers can be configured to provide the connectivity required by the first constraint within the power budget limitations imposed by the second. By controlling the network in the time domain, reconfiguration switching can be implemented electronically by a selection operation at the transmit and receive sites. This provides for faster switching times than are currently available in either WDM or spatial electro-optic switches.

3.0 Network Topology

In this section, we describe the topology of POPS networks as shown in Figure 1. POPS networks use a multiple passive star topology in which the interconnected nodes are partitioned into groups such that each group shares common inputs or common outputs from among a set of passive star couplers. POPS networks are distinguished from other types of multiple passive star topologies [Bir93,GLZ91] as follows. First, all couplers have equal fanout and are symmetric in the degree of fanin and fanout. Second, the nodes are completely connected with couplers arranged in parallel and without hierarchical interconnections. Thus, a path exists between every pair of nodes and each path traverses exactly one coupler.

All POPS networks are characterized by the parameter triple *(n,d,r)*. The first parameter, *n*, is the number of nodes. In Figure 1, each of the nodes to the left and to the right of the couplers is actually the transmit and receive logic of the same node. The second parameter, *d*, is the partition size. This parameter sets the size of each group

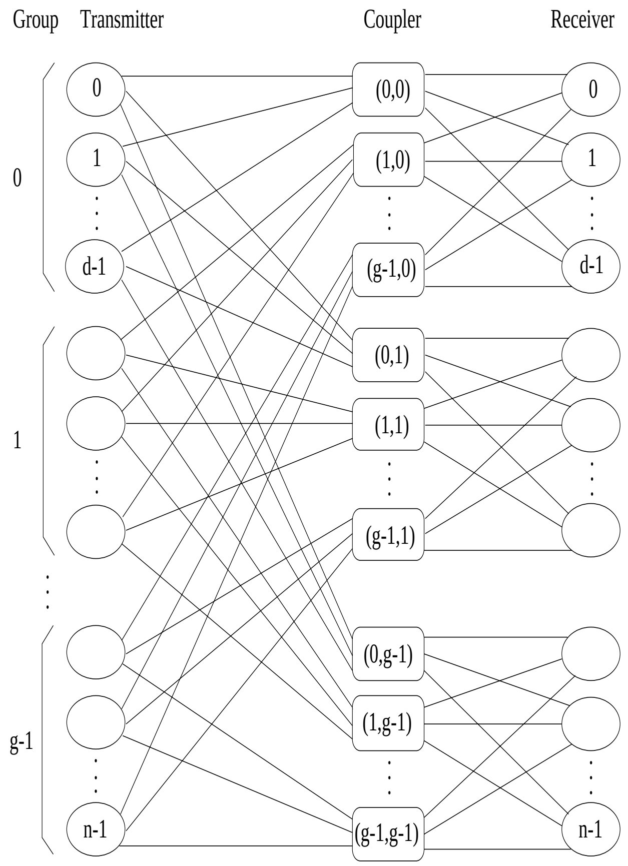

Figure 1: POPS Network Topology

and the fanin/fanout of the couplers. The third parameter, *r*, characterizes the redundancy of the network. Redundancy refers is the number of paths available between any pair of nodes. These paths may be independently switchable or may be parallel paths for fault tolerance or bit parallelism. The implications of redundancy in POPS networks is beyond the scope of this paper. We will restrict ourselves to networks with a redundancy of one.

It is useful to additionally define a fourth parameter, $g \equiv n/d$, which represents the number of groups into which the nodes have been partitioned. Each of the couplers in Figure 1 is identified by a double *(i,j)*, where, *i* is the group number of the nodes which share the input side of the coupler and *j* is the group number of the nodes which share the output side of the coupler. A POPS network is constructed by appropriately connecting couplers for all possible values $(0 \leq i < g, 0 \leq j < g)$. Each node is connected to the inputs of *g* couplers and is capable of independently transmitting a message into any one. Similarly, each node has *g* receivers connected respectively to the output side of *g* couplers and may independently receive a message from any one of the couplers on its receive side. Switching and configuration of the network is accomplished by selecting the appropriate output and input channels at each node. Thus, a message which traverses a coupler *(i,j)* is routed by a pair of selection operations, one at the transmitter and one at the receiver. The transmitter selects the coupler corresponding to the receiving node group, *j*, and the receiver selects the coupler corresponding to the node group, *i*, of the transmitter.

From this discussion, it is clear that the size of any POPS network is the product of two parameters, $n=dg$. The choice of these two parameters determines the number and distribution of resources in the network. Specifically, the size of each partition, *d*, determines the fanout requirement for each coupler. The number of groups, *g*, determines the number of transmitter/receiver channels per node as well as the total number of couplers in the system.

The complexity growth of the network under scaling also depends upon the choices made for *d* and *g*. We refer to a specific relationship between these parameters as a scaling rule. A scaling rule determines how additional nodes are allocated between groups when the network in increased in size. The fixed-*g* rule keeps the number of groups constant and increases the size of the network by increasing *d*. Under a fixed-*g* rule, network resource counts for couplers remains constant. Only the degree of fanout for the couplers is increased. The total number of transmitters and receivers in the network increases linearly with *n*. Conversely, the network can be scaled using a fixed-*d* rule. Under this rule, the node count is increased by adding additional groups and keeping the size of each group constant. In this case, coupler fanout remains constant and the total number of couplers increases as g^2. The total number of transmitters and receivers in the network increases as *ng*. The total capacity of the network, also increases as g^2. Clearly, the advantage of POPS networks is in their ability to scale up using a trade off between these two parameters. Consider another scaling rule, the square-root rule, in which *d* and *g* are not considered independently, but instead are increased according to the relationship, $d=c\sqrt{n}$, where *c* is a proportionality constant. Under this rule, the total couplers increase linearly in proportion to *n*, and the number of transmitter/receiver channels increase proportionately to $n\sqrt{n}$.

Given these characteristics and current technological limits on coupler fanout [Opt93], it is currently feasible to construct POPS networks on the order of a few thousands of processors. For example, using a coupler fanout of 64, a 1024 node system would have 16 groups, 16 transmitter and receiver channels per node and 256 couplers. Further, if one were to adopt integrated optical [OOOK92] or free space coupling[Hin89] in lieu of fused fiber, even higher fanouts and larger scale systems are possible. This domain is potentially very interesting for two reasons. First, as optical bandwidths continue to increase more rapidly than the data bandwidth per processor, high fanout coupling is more desirable. Second, in free space the essential component for which we quote complexity is spatial bandwidth rather than physical devices. How such a system would perform is the subject of later sections of this paper. We turn now to a discussion of the control system.

4.0 The State Sequence Control Paradigm

Our system for control of a POPS network is based on the paradigm of state sequence routing [CLMQ,-CLMQ93]. The goal of state sequence routing is to decouple the bandwidth of the optical channels in an electro-optically switched network from the bandwidth of the electronic control system which routes each message. In other words, state sequence routing provides a mechanism where network throughput need no longer be directly linked to the bandwidth of the control and routing system.

Consider a POPS network interconnecting a set N of n nodes. Within this network, an end-to-end connection between any two nodes is referred to as a path, p_{io}, such that $i,o \in N$, and $p_{io} \in P$, where the set P represents the complete connection space of the n^2 possible paths. If the network is partitioned into g groups, then the network can establish at any one time up to g^2 of the paths in the set P. This is accomplished by programming each of the source and destination nodes to respectively select specific output and input channels. That set of channel selections is collectively referred to as the network *state* and the corresponding set of paths implemented is referred to as a *mapping*.

Because individual pairs of paths may block over contention for a specific coupler or destination, not all sets of paths may constitute a mapping. Let M be the set of all mappings for a specific topology. If T is any arbitrary set of paths which corresponds to the current traffic, then it is always possible to partition T into subsets, t_i, such that each of the subsets is a mapping, $t_i \in M$, and $T = t_1 \cup t_2 \cup t_3 \cup .... \cup t_k$. In other words, there exists a sequence of mappings of length k which contains all paths in the current traffic and a corresponding sequence of states which implements those mappings.

In the simplest implementations of state sequence routing, the set of paths T is known apriori because it represents the static embedding of a fixed computational structure. For example, state sequences can be derived for a set of paths which represent the links in a mesh, a tree, or a cube connection structure. This sequence is repetitively applied to the switches in the interconnection network. A node may transmit whenever it detects that the network state implements a mapping which contains the desired path. If no such path exists in the current mapping, the node must wait. Further, a state sequence can be devised to implement all of n^2 paths in a completely connected network[Tho91]. In this case, state sequence routing is identical to time multiplexing. Unfortunately, the latencies inherent in a fully time multiplexed implementation, as well as for most other static embeddings, is prohibitive. This is because although the computational structure is regular, and the corresponding paths are repetitively available, the channel utilization is low and a large percentage of the available bandwidth is wasted.

In addition to the latency argument, another problem with static sequences is that for the majority of computing problems the computational structure cannot be known apriori. The current traffic set T changes dynamically as the computation progresses. If a state sequence is applied repetitively to the switching elements in the network, the control problem is a matter of transforming the state sequence to track the dynamic changes in the traffic. The essential point is that the control unit needs only to respond to the changes in the traffic and is no longer required to respond to individual messages. Since communication patterns in a multiprocessor environment tend to exhibit locality characteristics[Joh92], the rate at which changes occur in the traffic is significantly lower than the message generation rate. Thus, our stated goal of decoupling the control bandwidth from the throughput is achieved.

There are several methods of implementing dynamic sequence transformation. They can generally be classified by two characteristics, those which use fixed or variable sequence lengths, and those which allow or do not allow preemption of paths in the current sequence. For this paper, we focus on a method which uses fixed length sequences and in which path preemption is allowed. Under these assumptions, the selection of the path which must be preempted is the essential function of the control algorithm.

To understand this algorithm, it is useful to examine the correlation between this form of state sequence routing and the virtual memory page replacement problem. A virtual memory system attempts to emulate the functionality of a large address space by using a small amount of physical memory. Similarly, we are emulating the n^2 paths of a fully connected network using a small number of physical channels. By this analogy, a path is analogous to a memory location and a mapping is analogous to a page frame. The state sequence, which is a set of mappings including all of the currently available paths, is analogous to physical memory, which is a set of page frames containing all of the currently accessible memory locations.

In a virtual memory system, when a memory fault occurs, the memory controller arranges to replace one of the pages in physical memory with a page containing the required location. In state sequence routing, a sequence fault occurs when none of the mappings in the current state sequence contain the required path. In response to a sequence fault, a controller must replace one of the mappings in the sequence with a mapping which includes the required path. In both cases, the most significant control problem is selecting the entity which must be preempted to make room for the change. Optimally, the algorithm should choose the entity which will be used at the most distant future time.

Since future behavior cannot be precisely predicted, an optimal algorithm cannot be devised. However, the literature on operating systems contains numerous algorithms of varying complexity for selecting a candidate for replacement[SPG91]. A central focus of our current research is the application of these algorithms to state sequence routing. Although the replacement algorithm is important, the latency of any message, whether in a fault state or not, is also tied to the choice of sequence length. In the next section we turn to this issue.

5.0 Static performance analysis

In a POPS network using fixed length/preemptive state sequence control, the average latency between the generation of a message and the launching of the message is given by

$$L_{avg} = \left(\frac{kp}{2}\right) h + \left(f + \frac{3kp}{2}\right) (h-1)$$

where k is the sequence length, p is the period of a sequence step, h is the fault probability, and f is the service time for a sequence fault. In other words, messages for which a path exists in the sequence must on average wait for one half of the sequence period, kp. When a sequence fault occurs, a message must wait for one sequence period to determine that the path is not in the sequence, plus one fault service time, plus, on average, one half of the sequence period for the newly inserted mapping to be presented to the network. As a rule of thumb, the maximum fault probability that can be tolerated is related to the fault service time. However, in this case there is a substantial contribution to the latency of faulting messages from the *3kp/2* term. In fact, for many implementations the fault service time will be significantly less than the sequence period. Therefor, average message latency is critically dependent on the ability to efficiently deliver large volumes of traffic using a minimal sequence length.

The purpose of this section is to demonstrate this characteristic for POPS networks. We do so by static analysis of random traffic sets, *T,* to determine the number of steps in a sequence which would deliver the entire set without repetition. In the first analysis, a (1024,128,1) POPS network was considered. This network consisted of 1024 nodes in 8 groups, and used a total of 64 couplers. Ten thousand traffic sets, each consisting of 512 messages, which corresponds to 50% active processors, were generated. The results shown in Figure 2(a) are the average percentage of each traffic set which was transmitted at each step. Figure 2(b) shows the cumulative percentage of messages delivered on average after each step. From the latter result, we see that over 94% of the messages were delivered after only 10 steps and 100% of the messages were delivered in 22 steps. Further, Figure 2(a) shows that in the first five steps, between 12% and 12.5% of the message set of 512 messages were transmitted per step. This corresponds to a channel utilization of nearly 100% for the initial steps in the sequence.

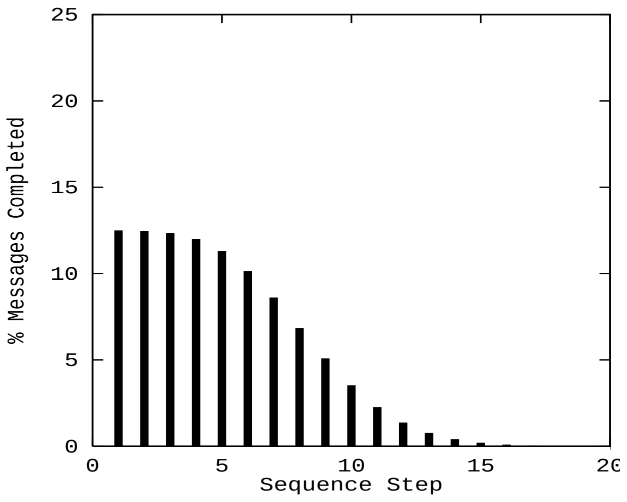

Figure 2(a):%Messages Completed vs. Sequence Step

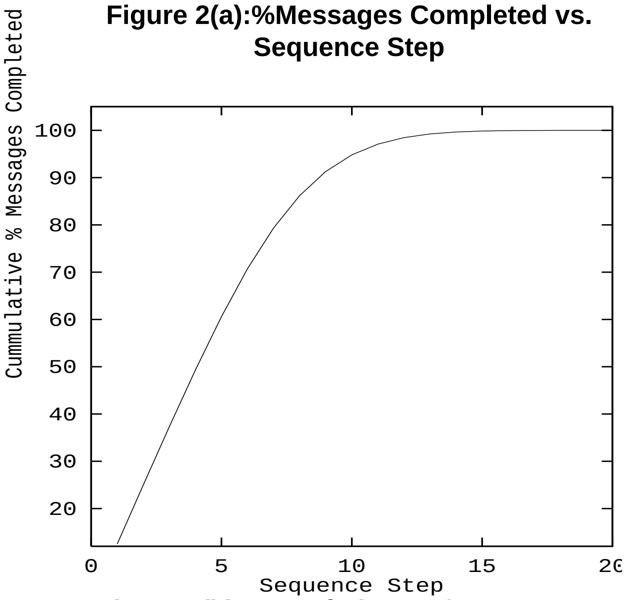

Figure 2(b): Cumulative% of Messages Completed vs. Sequence Step

In a second analysis, we considered several different network configurations ranging in size from n=64 nodes to n=1024 nodes, with group sizes given by the scaling rule $d=4\sqrt{n}$. By adopting this scaling rule, each increase in the number of nodes was accompanied by a linear increase in the total number of couplers, and an increase in the number of transmitter and receiver channels proportional to $n\sqrt{n}$. For each configuration, an analysis was made of the maximum number of steps required to deliver a random traffic set with size corresponding to 50% active processors. Figure 3(a) is a plot of the probability that a random traffic set can be delivered by a specific length sequence without repetition. Each curve corresponds to one system size over the range of the analysis. The scaling characteristics are summarized in Figure 3(b). This data shows that the average sequence length does not significantly increase even when the system is scaled over two orders of magnitude.

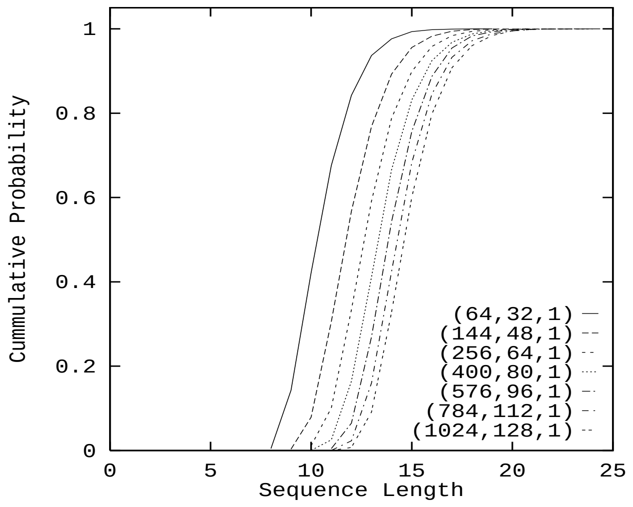

Figure 3(a): Complete Set Transmission Probability vs. Sequence Length

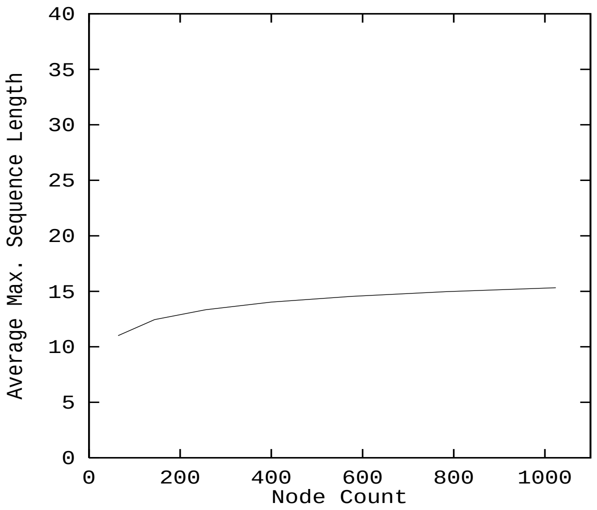

Figure 3(b): Complete Transmission Sequence Length vs. Node Count

These results show that the number of steps required to deliver a specific static traffic set is relatively short and that this is consistently the case even for large scale systems when the square root scaling rule is observed. For dynamic traffic, an underlying assumption is that changes in the traffic set will occur slowly due to locality. Our repetitive application of a fixed length sequence is based on that assumption. From the latency expression above, we determined that the performance of a dynamic system is critically dependant on the choice of sequence length, k. The static results show that sequences with small k can deliver substantial percentages of the current traffic set. With dynamic changes, transformations of the sequence are also required at a rate given by the sequence fault probability, h. The minimum value of h is determined by the locality in the traffic. However, the actual value may be greater, depending on contention for slots within the sequence as paths are preempted by the transformation algorithm. Such preemptions are control overhead and should be kept minimal. The rate at which they occur depends on the sequence length and on how nearly the transformation algorithm can achieve optimal replacement. The static results suggest that small values of k will provide sufficient slots to deliver the messages with minimal contention. Thus, fault probability as a function of k can be expected to decline sharply as k is increased to approach the values suggested by the static analysis. As h declines, our rule of thumb relating the service time, f, to fault probability allows us in turn to implement sophisticated replacement algorithms. The simulation data for the performance of a dynamic system presented in Section 7 confirms this relationship.

6.0 Distributed State Sequence Control

In the previous sections, we have analyzed POPS networks topologically and in terms of message latency. We turn now to the implementation of state sequence control. Within state sequence control, we identify two functionalities which the control system must provide to the network. The first is state sequence generation. This part of the control system keeps track of the current state sequence and transmits in turn each state word in the sequence to the control circuitry in the nodes. The sequence is transmitted repetitively and message traffic enters the network when the current state word corresponds to a mapping which contains the path required. The second functionality is state sequence transformation. This part of the control system is responsible for monitoring the nodes for the occurrence of a sequence fault. When a sequence fault occurs, the sequence transformer modifies the state sequence to provide the request path.

In this implementation, both the state sequence generation and state sequence transformation functions are distributed to designated nodes within the network. The problem is partitioned in much the same way that the POPS topology partitions the nodes. Each group of nodes in the POPS topology has within it a designated control node. Since each state word corresponds to a mapping, the state sequence generation function is partitioned such that the control node for each group generates a portion of the state word corresponding to any path which originates in that group. Similarly, state sequence transformation is partitioned such that the control node in each group services sequence faults for paths originating in that group.

A side effect of this partitioning of state sequence generation is that contention among the transmitters of a group for the coupler inputs, and contention among the coupler outputs for the receiver channels at each node must be resolved independently. For this reason, the state sequence generation function is additionally divided into a two phase control pipeline. The first phase of the control pipeline resolves contention for the couplers. The second phase resolves contention for the receiving nodes. One advantage to this system is that it also allows the use of the second phase to communicate sequence faults from each node to its respective controller.

Consider the (12,4,1) POPS network shown in Figure 4. It consists of 12 nodes numbered 0 to 11, and three groups of 4 nodes numbered 0, 1, and 2. The group number of any node is the node number modulo 4. Nodes 0, 4 and 8 are designated as control nodes for groups 0, 1, and 2, respectively. As in the previous example, couplers are identified by the double *(i,j)* where *i* indicates the upstream group whose transmitters fanin to the coupler and *j* the downstream group whose receivers are fanned out from the coupler. Thus, any path which originates from a node *s* and ends at node *d* will pass through coupler *(i=s mod 4,j=d mod 4).*

In the first phase of the control pipeline, the control node for the *i*-th group must generate a state word consisting of a set of fields $T_{ij}R_{ij}$ for all possible node groups *j*, where T_{ij} is the node designated to transmit into coupler *(i,j)*, and R_{ij} is the node designated to receive the output

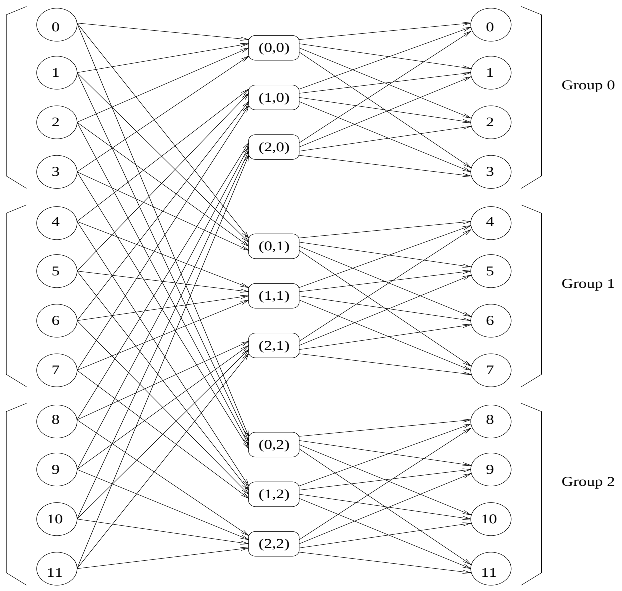

Figure 4: (12,4,1) POPS Network

from coupler *(i,j)*. As shown in Figure 5, in phase one the control nodes simultaneously transmit this word into all couplers connected to the group. Each node in the network simultaneously receives and buffers the control word from all controllers in the network.

At the end of the first control phase, each node is aware of all paths in the current state word. Based on the current node activity, it may elect to retain the path in the sequence for the second phase or to modify the sequence to reflect its current activity. For example, a node may discover a path in the sequence originating from itself, but it may not have a message in its output buffer. Similarly, a receiver may discover a path but not have an input buffer available, or it may discover that multiple paths end at its receivers. Based on these modifications each node, *i*, generates one field, N_i, of the modified phase two state word. N_i encodes the combination of the current transmit and

Transmit Condition	Description	Receive Condition	Description
Idle	Transmit buffer empty or no path in mapping.	Idle	No path in mapping terminating at node
Transmit *n*	Message to destination *n*, mapping unmodified	Busy	Path in mapping but no input buffers available.
Fault *n*	Sequence fault for dest. *n*, sent in lieu of idle	Receive *g*	Path in mapping from group g, mapping unmodified

Table 1: Transmit and Receive Node Conditions for Phase 2

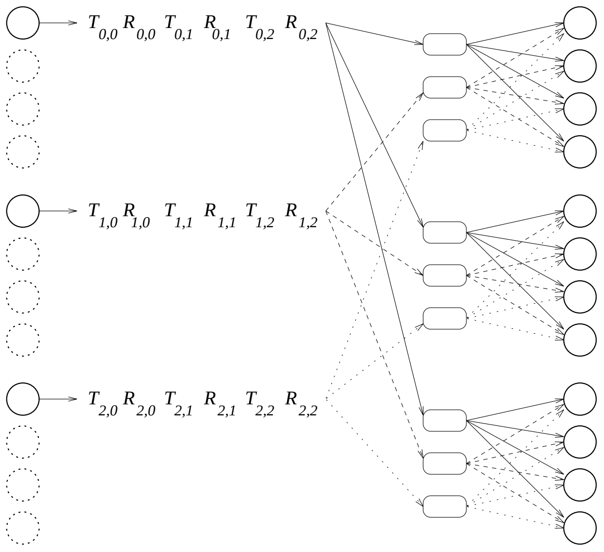
$T_{0,0}$ $R_{0,0}$ $T_{0,1}$ $R_{0,1}$ $T_{0,2}$ $R_{0,2}$
$T_{1,0}$ $R_{1,0}$ $T_{1,1}$ $R_{1,1}$ $T_{1,2}$ $R_{1,2}$
$T_{2,0}$ $R_{2,0}$ $T_{2,1}$ $R_{2,1}$ $T_{2,2}$ $R_{2,2}$

Figure 5: Control Pipeline Phase 1

receive activity for the node. Possible values for each are shown in Table 1. As in phase one, each node transmits its portion of the modified state word simultaneously on all of its output channels. Within a group, each node field is positioned in the output word based on the node number offset within the group. All other bits in the output word of each node are zero. As shown in Figure 6, one segment of the phase two state word is produced for each group by an optical OR operation within the couplers connected to that group. Combined in this fashion, all of the phase two state word segments are simultaneously received and buffered by each node.

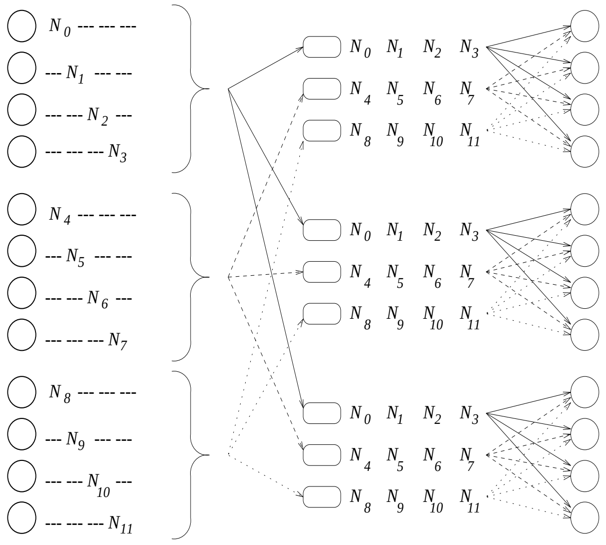

Figure 6: Control Pipeline Phase 2

Once the modified state sequence is received by the nodes, each node which has data in its output buffer examines the modified sequence fields for the source and destination to determine if the required path exists. The path exists if the node states for both the source and destination nodes indicate transmit and receive activity corresponding to the required path. When this condition is met, the message is transmitted. Since the modified state word also indicates fault conditions, each of the control nodes examines the segment generated within its own group for any node state which indicates a sequence fault. If so, the sequence fault service algorithm selects a location in the phase one state sequence into which the faulting path is placed.

An important issue relative to performance and scalability is the encoding method for information in the control pipeline. For stage one, $T_{i,j}$ and $R_{i,j}$ encode a node offset within the group. Thus, the size of each field of a state word segment in phase one is O($log_2(d)$) bits long and each segment is O($g\ log_2(d)$) bits. In phase two, each of the N_i fields must encode the combined transmit and receive activities of a node. From Table 1, there are a total of $2n+1$ transmit states and $g+2$ receive states. Therefore, the field size required to encode each node activity is O($log_2(ng)$) and the total length of each modified state word segment is O($d\ log_2(ng)$) bits.

The encoding complexity of the control word is important because it represents the amount of information which must traverse the network within each sequence step. In the latency equation in section 5, the period of a sequence step is represented by *p* and contributes to the latency in the relationship as the sequence length. The exact rate of growth in *p*, relative to the size of the network, is implementation dependent. However, if we assume that the control phases are overlapped, the maximum number of bits per segment are in phase two. Thus, for systems which are scaled using the square root scaling rule, *p* grows as O($\sqrt{n}\ log_2(n\sqrt{n})$).

7.0 Simulation Results

The analysis in Section 5 demonstrated that for static traffic sets, it was possible to deliver the entire message set in a relatively small number of steps. It was also shown that the number of steps required was essentially constant when systems were scaled using the square root scaling rule. For fixed length sequences and dynamic traffic sets, this result is useful to guide us in our selection of *k*. In this section, we extend this result by a simulation study to characterize the fault probability and latency in terms of sequence length for a POPS network under dynamic control. In a second study, we select a fixed value for *k* and characterize the latency of the network when scaled using the three scaling rules: fixed-*d*, fixed-*g*, and square root.

The simulation model is a POPS network with system size, partition size, sequence length, and traffic profile, set by parameters. The topology is the one shown in Figure 1 and the control system is based on the description in Section 6. The simulator is event driven with its timebase defined to be one step in the state sequence. Thus, each tick includes one step in each phase of the control pipeline and one set of message transmissions and receptions. The propagation latency of the fiber in the network is assumed to be two ticks, and this latency is applied to both control segments and messages. Receive and transmit portions of the state word are skewed to take this latency into account. Each node is assumed to have one output buffer and one input buffer for messages. Thus, only one message at a time is available for output from a transmitting node and each receiving node is considered busy for two ticks after a message is received to allow time for the node to consume the message. Message latency is defined to be an interval starting when a message is placed in the output buffer of the source node and ending when the message arrives in the input buffer of the destination node. Sequence faults cause path preemptions in the sequence according to an NUR replacement algorithm with the additional restriction that a new path is not eligible for replacement until it is used at least once.

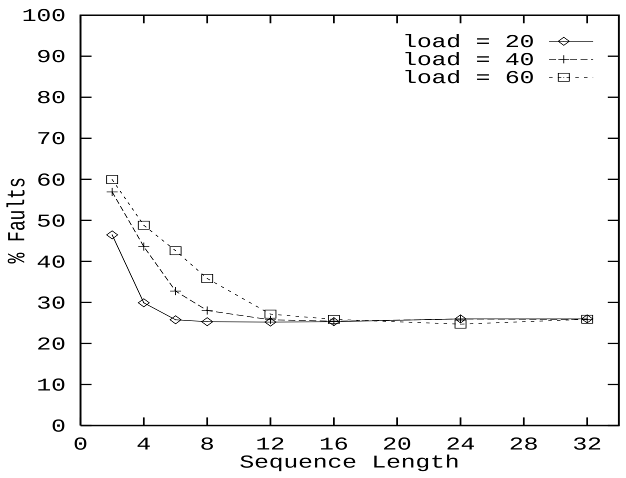

Figure 7(a): Fault Rate vs. Sequence Length

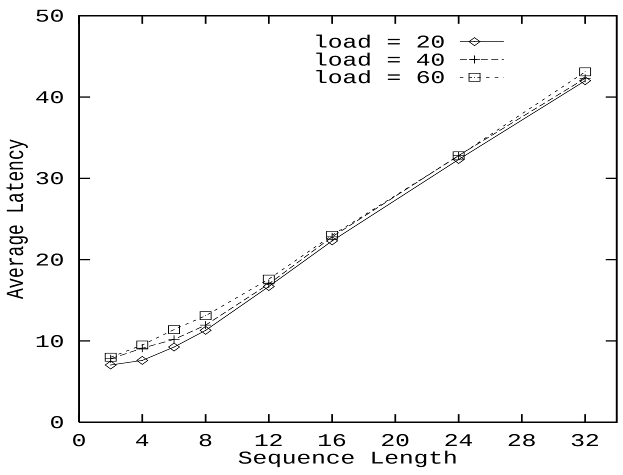

Figure 7(b): Latency vs. Sequence Length

Message traffic for each simulation run consisted of bursts of messages where all messages in each burst shared a common source and destination. Traffic was characterized by three values: burst interval, burst length, and burst rate. Burst interval is the time between end of a burst and the initiation of a new burst, burst length is the number of messages in a burst, and burst rate is the time between

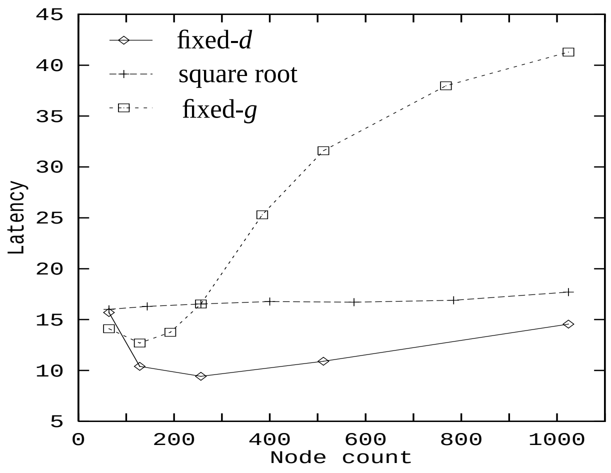

Figure 8: Latency vs. Node Count

initiation of individual messages within a burst. Each of these values was generated at random for each burst at each node during the course of the run, based on parameters specifying the average and range of each value. The destination node was also generated at random for each burst at each node. Based on these parameters, we calculate two statistics regarding the traffic for each run. Load average is the average number of new messages generated per tick as a percentage of the network capacity per tick. Message locality is the ratio of the number of bursts to the number of messages transmitted by a node. This is also the inverse of the average burst length.In the first of our simulation experiments, we selected a (512,64,1) network. Simulation runs were made for sequence lengths ranging from 2 to 32. Figure 7(a) shows the percentage of sequence faults vs. sequence length for load averages of approximately 20, 40, and 60 percent with a message locality of 75 percent. Figure 7(b) shows the message latency versus sequence length for the same runs. The results in Figure 7(a) show that excess faults caused by premature path preemptions decline sharply as the sequence length approaches the value suggested by the results in section 5. Beyond this value, the fault rate approaches a minimum of approximately 25 percent which corresponds to the message locality of the traffic. The message latency, shown in Figure 7(b), rises slowly for short sequence lengths as the overhead from excess faults is reduced. For longer sequences, the latency increases linearly with sequence length.

The second simulation experiment was designed to determine the effect on latency when a system scales in the number of nodes. Figure 8 shows these results for system sizes ranging from 64 to 1024 nodes, all using a sequence length of 12. The three curves shown are latency versus node counts under three different scaling rules. The first rule, shown by the solid line, is the fixed-*d* rule in which the system is scaled by increasing the number of groups while keeping the size of each group constant, in this case 32 nodes. This corresponds to quadratically increasing the network capacity since the number of couplers increases as g^2. Thus, the results show an initial sharp decline in latency as the additional couplers are utilized by the nodes. However, while this scaling rule decreases contention for the couplers, there is a corresponding increase in contention at the receiving nodes. This effect begins to dominate at system sizes above a few hundred nodes. Beyond this point, overall message latency slowly increases in proportion to the node count. The dotted line in Figure 8 shows the fixed-*g* scaling rule. Each system uses exactly 4 groups and system size is scaled by increasing the group size. In this result, there is an initial decrease in latency from 64 to 128 nodes as utilization increases. Beyond that point, sequence faults from contention for the couplers cause latency to increase sharply towards a limit corresponding to saturation of the output buffers in the transmitting nodes. The third case, shown by the dashed line, is the square root scaling rule. This corresponds to linearly increasing coupler count and channel capacity as the system is scaled. Under this rule, the static analysis predicted fixed latency over scaling. The simulation results confirm this prediction, showing only a marginal increase in message latency over the range of the experiment. Thus we have verified the linear scaling with constant latency characteristic of the network design.

8.0 Conclusions and Future Research

In conclusion, we have presented in this paper an optical interconnection network which combines the topological advantages of a multiple passive star configuration with the control paradigm of state sequence routing. Table 2 is a summary of the network parameters and their complexity growth under the square root scaling rule. The significant features of the network include linear scaling in couplers, sublinear scaling in power budget and coupler fanout, and $n\sqrt{n}$ scaling in transceiver count. We have shown by static analysis and by simulation that latency is nearly constant under scaling in terms of the control sequence steps. The amount of control information per sequence step increases sublinearly under scaling

Our studies to date have used only synthetic traffic loads. We are currently constructing a small prototype to verify these results for actual multiprocessor traffic. In addition, we are actively investigating the appropriate choice of algorithms for sequence fault handling in order to characterize the trade-off between complexity and performance in this aspect of the design..

Resource	Notation	Growth
node	n	n
nodes/group	d	$\sqrt{n}$
groups	g	$\sqrt{n}$
couplers	g^2	n
coupler fanout	d	$\sqrt{n}$
transceivers	nd	$n\sqrt{n}$
latency	(kp/2)(1-h) + (f+3kp/2)h	kp
control bits	d log(ng)	$\sqrt{n}\ log(ng)$
power	d	$\sqrt{n}$

Table 1: POPS Network Parameters and Scaling Characteristics

9.0 References

[Bir93] Y. Birk. Power-optimal layout of passive, single-hop, fiber-optic interconnection whose capacity increases with the number of stations. In *INFOCOM '93: 12th Joint Conference of the Computer and Communications Societies*. IEEE, 1993.

[BLM93] Y. Birk, N. Linial, and R. Meshulam. On the uniform-traffic capacity of single-hop interconnections employing shared directional multichannels. *IEEE Transactions on Information Theory*, 39(1):186–191, January 1993.

[Bra90] C. A. Brackett. Dense wavelength division multiplexing networks: Principles and applications. *IEEE Journal on Selected Areas in Communications*, 8(6):948–964, August 1990.

[CL91] W. T. Chen and H. J. Liu. An adaptive scheduling algorithm for TDM systems.

In *INFOCOM '91*. IEEE, 1991.

[CLMQ] D. M. Chiarulli, S. P. Levitan, R. G. Melhem, and C. Qiao. Locality based control algorithms for reconfigurable optical interconnection networks. *Applied Optics*, to appear.

[CLMQ93] D. M. Chiarulli, S. P. Levitan, R. G. Melhem and and C. Qiao. Bandwidth as a virtual resource in reconfigurable optical interconnections. In *Optical Computing Digest, 1993*, volume 7, pages 299–302, Washington, D. C, 1993. Optical Society of America.

[CM92] Y. L. Chang and M. E. Marhic. Fiber-optic ladder networks for inverse decoding coherent CDMA. *Journal of Lightwave Technology*, 10(12):1952–1962, December 1992.

[Dow91] P. W. Dowd. Random access protocols for high-speed interprocessor communication based on an optical passive star topology. *Journal of Lightwave Technology*, 9(6):799–808, June 1991.

[GLZ91] A. Ganz, B. Li, and L. Zenou. Reconfigurability of multi-star based lightwave LANs. In *GLOBECOM '91*, New York, N. Y., 1991. IEEE.

[Hin89] H. S. Hinton. Relationship between the hardware required for photonic switching and optical computing based on free-space digital optics. In *OSA Proceedings on Photonic Switching*, pages 184–191, Salt Lake City, Utah, March 1-3 1989. Optical Society of America.

[HS92] A. S. Holmes and R. R. A. Syms. All-optical CDMA using quasi-prime codes. *Journal of Lightwave Technology*, 10(2):279–286, February 1992.

[Joh92] K. L. Johnson. The impact of communication locality on large-scale multiprocessor performance. *Computer Architecture News*, 20(2):392–402, 1992.

[KVG+90] H. Kobrinski, M. P. Vecchi, M. S. Goodman, E. L. Goldstein, T. E. Chapuran, J. M. Cooper, M. Tur, C. E. Zaha, and S. G. Menocal. Fast wavelength-switching of laser transmitters and amplifiers. *IEEE Journal on Selected Areas in Communication*, 8(6):1190–1201, August 1990.

[Meh90] N. Mehravari. Performance and protocol improvements for very high speed optical fiber local area networks using passive star topology. *Journal of Lightwave Technology*, 8(4):520–530, April 1990.

[Muk92] B. Mukherjee. WDM-based local lightwave networks, part 1: Single-hop systems. *IEEE Network*, pages 12–27, May 1992.

[OOOK92] K. Okamoto, H. Okazaki, Y. Ohmori, and K. Kato. Fabrication of large scale integrated-optic n x n star couplers. *IEEE Photonics Technology Letters*, 4(9):1032–1035, September 1992.

[Opt93] Gould Fiber Optics. *Gould Product Catalog*. 1993.

[PSS86] P. R. Prucnal, M. A. Santoro, and S. K. Sehgal. Ultrafast all-optical synchronous multiple access fiber networks. *IEEE Selected Areas in Communication*, 4(9):1484–1493, December 1986.

[SJ91] D. A. Smith and J. J. Johnson. Switching speed of integrated acoustically-tunable optical filter. *Electronics Letters*, 27(23):2102–2103, November 7 1991.

[SPG91] A. Silberschatz, J. Peterson, and P. Galvin. *Operating System Concepts, 3rd edition*. Addison Wesley, 1991.

[SWH90] J. A. Salehi, A. M. Weiner, and J. P. Heritage. Coherent ultrashort light pulse code division multiple access communication systems. *Journal of Light Wave Technology*, 8(3):478–491, March 1990.

[Tho91] R. A. Thompson. The dilated slipped banyan switching architecture for use in an all-optical local-area network. *Journal of Lightwave Technology*, 9(12):1780–1787, December 1991.