Deadlock-Free Routing in an Optical Interconnect for High-Speed Wormhole Routing Networks

Prasasth Palnati*, Mario Gerla†
405 Hilgard Ave, Computer Science Department,
University of California, Los Angeles, CA 90024
{palnati,gerla}@cs.ucla.edu
Ph: (310) 825-1888

Emilio Leonardi
Department of Electronics
Politecnico di Torino, Torino 10129
leonardi@polito.it
Fax: (310) UCLA CSD

Abstract

The Supercomputer SuperNet (SSN) is a two-level hierarchical high-speed network. The lower level is a High Speed Electronic mesh fabric; the higher level is a WDM optical backbone network interconnecting the high-speed fabrics distributed across a campus or metropolitan area. The salient characteristics of this architecture are the use of wormhole routing and backpressure hop-by-hop flow control mechanism. Because of these features, deadlocks are possible in SSN. In this paper, we address the issue of deadlock-free routing which is an essential prerequisite for the proper operation of SSN. To this end, we first present a deadlock free routing scheme for the WDM backbone which is implemented with a shufflenet multihop virtual topology. We use the notion of virtual channels to obtain mappings of virtual channels to physical channels such that deadlock-free routing is achieved for any (p, k) shufflenet (uni and bidirectional). Then, we compare the virtual channels scheme with the more conventional Up/Down deadlock free routing scheme for the bidirectional shufflenet and show that the former yields much better performance. Finally, we address the problem of deadlock prevention across the entire network (i.e., lower level fabric as well as the optical backbone) and develop an integrated solution combining different schemes best suited for the different levels.

Keywords: Routing, Deadlock prevention, wormhole routing, virtual channels, optical WDM network.

*Corresponding author.

†This work was supported by the USDOD ARPA/CSTO under Contract DABT63-93-C-0055. The Distributed Supercomputer Supernet – A Multi Service Optical Intelligent Network.

1 Introduction

Optical networks have the potential to provide much higher aggregate throughputs than high speed electronic networks by employing dense Wavelength Division Multiplexing (WDM) [4] to access the vast bandwidth offered by the optical fiber. Electronic networks, however, provide a much more cost effective means to interconnect local hosts upto the Gbps range. Furthermore, they possess the intelligence required for network control and management. A two-level hierarchical network architecture that combines the throughput advantages of optical interconnection with the low cost and intelligence of electronic switches becomes a natural candidate for exploration. Such an architecture has been defined and discussed in [18] and [17] for the Supercomputer SuperNet project (see Fig. 1). The lower level of this architecture consists of host computers connected to crossbar switches - the interconnections are by point-to-point full-duplex links - forming a High Speed Electronic Local Area Network (electronic LAN). Several such electronic LANs (distributed across a campus or a metropolitan area) are interconnected by an optical backbone network (with a single passive optical star physical topology) using Wavelength Division Multiplexing (WDM).

Some of the salient features of the high speed electronic LANs include:

- *Wormhole routing:* a form of cut-through routing (the unit of information transfer is a worm) [16],
- *Source routing:* the worm header carries the complete route,
- *Backpressure flow control:* implemented on a hop-by-hop basis.

An Optical Channel Interface (OCI) acts as an interface between the electronic LAN and the optical star network. SSN provides support for both circuit-switched and packet-switched traffic for high performance Supercomputers which are geographically distributed across a metropolitan area (about 30Km radius). Support for circuit-switching is provided by dedicating wavelengths between a pair of OCIs. Packet-switching support can either be provided by single hop networks [22] or by multihop networks [23]. For the SSN architecture, packet-switched traffic is actually worm-switched traffic. The unit of transfer between hosts (message) is a worm composed of flow control digits (called flits). We shall assume a flit equals a byte. Worm lengths can range from a few tens of bytes to thousands of bytes. In this context, a few protocols for worm-switching in single hop networks were presented and analyzed in [7]. Another approach to worm-switching is the use of a multihop virtual topology (like a Shufflenet [1, 12], de Bruijn graph, Manhattan Street Network etc).

With wormhole routing, a worm is blocked when it finds its path occupied by another worm. When blocked, part of the worm is buffered in a buffer called the slack buffer. By employing the backpressure flow control mechanism, the blocked worm will hold a path (which may extend all the way to the source host), and may thus block other worms. It is clear that deadlocks may occur in this situation. Fortunately, efficient techniques for electronic LAN deadlock prevention exist as discussed later.

One of the goals of SSN is the extension of the wormhole routing, source routing and backpressure flow control properties of the electronic LANs to the entire campus or metropolitan network in a transparent fashion. Transparency here refers to the property that the protocols used in the high speed electronic LANs are identical to the ones employed in the optical star network and thus the optical network (in particular the multiple hops for worm-switching) appears to the host stations as a single link joining two electronic LANs. As a consequence of these goals, the optical links are bidirectional (to facilitate backpressure hop-by-hop flow control). Thus, a new kind of virtual multihop topology called bidirectional shufflenet topology is employed. Properties of such a virtual topology are presented in [24, 19].

An important requirement for SSN is deadlock prevention. Indeed, this property must be guaranteed for the entire two-level SSN system comprising electronic LANs and WDM backbone. Without deadlock prevention, there is no guarantee of reliable delivery since worms will be lost when a deadlock occurs. The wormhole routing operation and the particular nature of the electronic switches make the conventional packet-switching deadlock prevention schemes not cost effective or even infeasible. Thus, novel deadlock prevention schemes must be designed.

In this paper, we address the issue of providing deadlock-free source routes for the two-level SSN architecture. We do this by first addressing the issue in the case of the traditional shufflenet architecture in isolation. Then we look at the issue in the context of the bidirectional shufflenet multihop virtual topology which can potentially be employed in SSN because of its natural support for hop-by-hop backpressure flow control. Next we present protocols for serving virtual channels and compare their performance via simulation for the bidirectional shufflenet topology. Finally, we extend our scope to include the complete network (electronic LANs and the WDM backbone).

In section 2, we present a brief overview of the SSN network architecture. Section 3 summarizes the previous work done in the area of deadlocks in computer networks. In section 4, we present the scheme for achieving deadlock-free routing for the shufflenet multihop virtual topology followed by the schemes for the bidirectional shufflenet multihop virtual topology in section 5. Section 6 presents protocols for serving virtual channels and compares their performance via simulation for the bidirectional shufflenet. Section 7 addresses the issue of providing deadlock-free routing in the SSN architecture. Finally, we present our conclusions in section 8.

2 Network Architecture

The two-level hierarchical network architecture that we consider for this paper is shown in Fig. 1. At the lower level, the host stations interface to the high speed electronic LAN. Using full-duplex point-to-point links, the crossbar switches are interconnected to form the electronic LAN. For SSN these crossbar switches are provided by $Myricom, Inc.^{©}$ At the higher level, we have an optical star network that interconnects several electronic LANs. The interface between the two different types of networks is provided by the Optical Channel Interface (OCI).

2.1 High Speed Electronic LAN

Host stations belonging to the electronic LAN are connected to the crossbar switches using full-duplex point-to-point links. An arbitrary topology can be created by interconnecting various crossbar switchs as desired. The salient features of the high speed electronic LAN are:

1. **Wormhole routing**: The traffic (both packet-switched and circuit-switched) in the network is sent from one host to another as worms (messages). Each switch forwards the worm without buffering it. This means that the worm could stretch across several links and switches at the same time. The worm threads its way through the routers until it is switched to the destination host. If at any intermediate switch, the required output port for the worm is not available, the worm is blocked. Upon blocking a worm keeps the resources allocated to it so far. When the output port becomes available the worm continues its journey towards the destination. Since there is a possibility of deadlock in this scenario a deadlock avoidance approach, namely, Up/Down routing (as in Autonet [26]) is used.

2. **Source routing**: The header of each worm has a source route specified in it. The source route consists of the sequence of switch port numbers through which the worm must travel to get to its destination. It is assumed that the network has one or more Address Consultants which specify the source route upon request by the source station. Each crossbar switch is a complete crossbar without internal blocking. The worm is switched towards the output port specified in the header, then this port specification is removed from the header; thus, at the next node, the head of the worm has the appropriate output port specification. In order to avoid deadlocks in SSN, the source routes provided by the Address Consultant(s) must be deadlock free. We address this issue later.

3. **Backpressure flow control**: The input rate at a switch port may exceed the rate at which the worm leaves the switch if either the worm is blocked due to contention for an output port or when there is a speed mismatch in the rate at which the stations can receive worms. For

this purpose, STOP and GO signals are used on the full-duplex links to exercise flow control on a link by link basis. Using this backpressure mechanism, the entire worm is frozen from the point of contention to the point of generation. This causes the resources allocated to the worm, namely, buffers at intermediate mesh routers, to be 'frozen' as well. Thus it is possible to have a circular wait causing deadlocks.

For the SSN project, the high speed electronic LANs are the Myrinet[1] LANs provided by $Myricom, Inc.^{TM}$ Myrinet LANs can sustain speeds of up to 640 Mbps over dual twisted pair cables of length up to 75 ft (25m). Myrinet LANs have at least one Address Consultant (Route Manager) that provides source routes upon request. Myrinet LANs use the features discussed earlier to sustain such high speeds.

2.2 Optical Network

The optical network has a broadcast-and-select star topology (using a passive star coupler). By employing WDM, the optical network can provide circuit switched service by dedicating a wavelength between two OCIs. The packet switched service can be provided by configuring the network either as a single hop network or as a multihop network. The single hop network requires the use of tunable transmitters or receivers whereas the multihop network can be implemented by using only fixed tuned transmitters and receivers. In this paper, we focus on the multihop virtual topologies (shufflenet and bidirectional shufflenet) for providing packet-switching support.

For the SSN project, any virtual multihop topology can be configured on the optical star. For this paper we consider two virtual multihop topologies - traditional shufflenet[1, 6, 12, 14, 15] and the bidirectional shufflenet[24, 19]. An example 8-node shufflenet and the corresponding bidirectional shufflenet are shown in Fig. 2.

2.3 Optical Channel Interface

The Optical Channel Interface (OCI) acts as an interface between the electronic LAN and the Optical network. The OCI-LAN interface supports the backpressure form of flow control used in the LAN. When a portion of a worm that reaches the OCI finds that the buffers at the OCI are full, the backpressure form of flow control is used to stop the arrival of the rest of the worm until the buffers at the OCI are released. Each OCI is equipped with a fixed number of transmitters and receivers (tunable or fixed-tuned). The OCI also has processing power and thus, it can implement fairly sophisticated protocols unlike the Myrinet switches.

[1] Myrinet is a trademark of $Myricom, Inc.^{TM}$

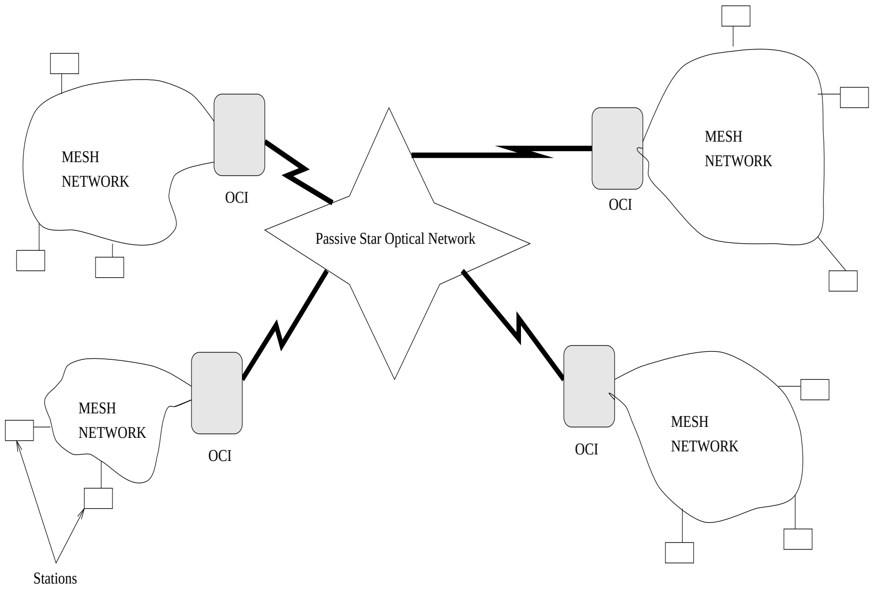

Figure 1: The two-level hierarchical network architecture of the Supercomputer SuperNet. One level is the High Speed Electronic LAN, the other is the Optical Star network.

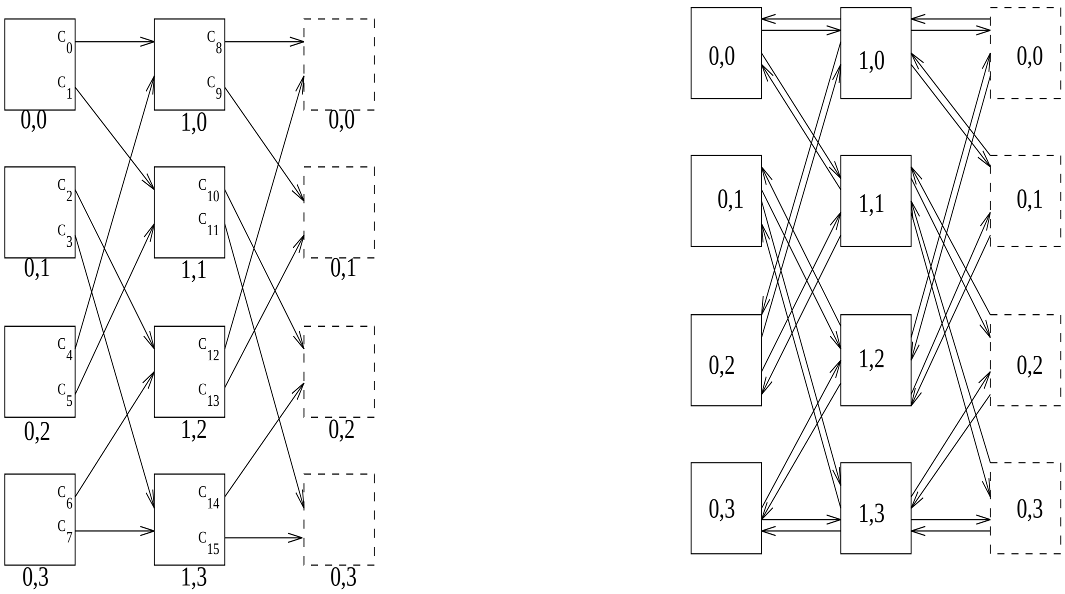

Figure 2: An 8-node shufflenet and an 8-node bi-directional shufflenet multihop topology. The first and third column represent the same nodes.

3 Deadlock-Free Wormhole Routing

The issue of deadlocks in computer networks (and more generally in computer systems) has been well studied. Basically, a deadlock is said to occur in a computer network when a state is reached such that no progress can be made and the network will stay in such a state until some 'external' action is taken. Progress for computer networks is the continuation of transport of messages toward the destination. An alternative definition is that a deadlock occurs when resources are allocated in such a fashion that there is a cycle (involving at least two processes) among processes which are waiting for more (already allocated) resources to complete their task and such a cycle cannot be broken without external intervention. The theory of deadlocks is presented in a seminal work by Holt [13].

For packet-switched computer networks, two kinds of approaches are employed to achieve deadlock-free routing. The first approach is restricting the routes that a packet may take towards the destination. Finding and using routes in the network in such a fashion that deadlocks are avoided works at the cost of non-optimal routes (in terms of number of hops). Such an approach was adopted in the Autonet [26, 27]. In such networks, a spanning tree is constructed (based upon the connectivity in the network) and then all hosts use this tree to determine routes to all other hosts. Routes consist of zero or more 'up' moves (towards the root) followed by zero or more 'down' moves (away from the root). Deadlock freedom of the routes is guaranteed by the fact that a spanning tree is used and all 'up' moves are made before 'down' moves towards the destination. In general, the paths obtained subject to such Up/Down routing are not the shortest paths that are possible in the network.

Another approach to avoiding deadlocks consists of defining a partial order on the buffers (hierarchy of packet buffer pools) at each node [8, 9, 10, 21, 28, 29]. Most of these schemes avoid deadlocks by defining a partial order on the buffers and allocating them on various criteria like number of hops traveled, the route which the packet is traveling etc. Further, priorities in the order of service are defined among the buffers. For packet-switching it is easy to implement such prioritised service.

Some mesh networks like the Manhattan Street Network [3, 20] use deflection routing along with a random choice of inputs for service to avoid deadlocks. However, the delay in this case is unbounded.

The concept of wormhole routing differs from store-and-forward routing in the way a message is relayed toward the destination. In store-and-forward networks, the message is received at an intermediate node[2] (or switch) completely and then it is sent to the next node on a path to the

[2]We use node and switch interchangeably throughout the paper.

destination. However, in wormhole routing, the intermediate node forwards the incoming worm to the required output port (if available) without storing the entire worm first. Thus, it is possible for a worm in a wormhole routing network to span several nodes and links at any time (creating something like a virtual circuit for the duration of the worm transfer). If at an intermediate switch the desired output port is not available, the worm is said to be blocked and it is buffered until the small slack buffer associated with the incoming port is filled. Then by employing the backpressure flow control mechanism, the worm is frozen in place until the required output port becomes available.

Of the two basic approaches described above, it is possible to use the approach of restricting routes (Up/Down routing) for wormhole routing networks without any modifications.

In principle it is possible to use the hierarchy of buffer pools for deadlock prevention in the wormhole routing context. However, for this concept to work, the switches need to be capable of interleaving worms. The need to interleave worms arises due to the requirement that there be some priority in the order of service of the buffers – this means that if a worm in a lower priority buffer gets blocked, then a worm in a higher priority buffer must be served leading to interleaving of worms.

An approach similar to the hierarchy of buffer pools, called virtual channels, combines restricted routing and buffer partition to achieve deadlock free routing in the wormhole routing context. This was proposed in [5].

Virtual channels are groups of channels that share a physical channel. Each virtual channel has its own buffer. Thus, the buffer space is partitioned according to the number of virtual channels. Physical channels that belong to a cycle in the network are decomposed into a group of virtual channels (see Fig. 3) and then routing is restricted (only in the sense that the proper virtual channel must be selected when a physical link must be crossed) such that the virtual channels are traversed in a particular order, thereby eliminating cycles. Virtual channels capture the virtual-circuit like behavior of wormhole routing while providing near optimal buffer partitioning (in terms of number of virtual channels). In [5], the authors provide mappings of virtual channels to physical channels for obtaining deadlock-free routes for three topologies – k-ary n-cubes, cube-connected cycles and for shuffle-exchange networks. We extend their work by providing the mappings for the traditional shufflenet and the bidirectional shufflenet.

The concept of virtual channels implies a protocol for serving the virtual channels that are mapped onto the same physical link. This issue has not been discussed in previous literature. The multiplexing of several virtual channels on the same physical channel requires that the switch have the ability to interleave worms. Unfortunately, Myrinet switches do not currently support interleaving of worms. Thus, this scheme cannot be implemented in the high speed electronic LAN of SSN. However, the OCI has intelligence and we can use this intelligence to interleave worms while serving virtual channels in the WDM optical backbone. The protocols that do this are presented and evaluated later. The interleaving of worms is transparent to the upper layer protocols and to the electronic LAN.

3.1 Principles of Deadlock-Free Routing using Virtual Channels

In order to introduce the notation, we reproduce below some of the definitions from [5].

Definition 1: An *interconnection network* I is a strongly connected directed graph, $I = G(N, C)$. The vertices of the graph, N, represent the set of nodes. The edges of the graph, C, represent the set of communication channels. Associated with each channel, c_i, is a queue (the slack buffer).

Definition 2: A *routing function* $R : C \times N \rightarrow C$ maps the current channel c_c and the destination node n_d to the next channel c_n on the route from c_c to n_d, $R(c_c, n_d) = c_n$.

Definition 3: A *channel dependency graph* D for a given interconnection network I and routing function R, is a directed graph, $D = G(C, E)$. The vertices of D are the channels of I. The edges of D are the pairs of channels connected by R:

$$E = \{(c_i, c_j) | R(c_i, n) = c_j \ for \ some \ n \in N\}.$$

Dally and Seitz then prove a theorem which is the basis for design of deadlock-free routing algorithms for wormhole routing interconnection networks. We state the theorem next.

Theorem: A routing function R for an interconnection network I is deadlock free iff there are no cycles in the channel dependency graph D.

In light of the above theorem, deadlock-free routing can be achieved by breaking the cycles in the channel dependency graph. A cycle in a channel dependency graph may be broken by splitting each physical channel into a group of virtual channels such that by eliminating one virtual channel we construct a channel dependency graph which has no cycles. In the next two sections we apply this technique to show how to do deadlock-free routing in the shufflenet multihop topology and the bidirectional shufflenet multihop topology.

It is easy to see that Up/Down routing satisfies the above theorem since there are no cycles in the channel dependency graph for a network where it is used. However, Up/Down routing results, in general, in larger average path lengths than routing using virtual channels. To illustrate this, we show in Fig. 3 a 10-node ring topology along with the spanning tree (a chain) used for Up/Down routing and the routing made possible by using virtual channels. The average hop length in the spanning tree (UP/DOWN routing) is 3.67 while for the virtual channel routing scheme, it is 2.5.

Also, it can be seen that links near the middle of the chain will be more heavily loaded than other links under uniform traffic conditions. This is not so for the scheme employing virtual channels.

4 Deadlock-free Routing for Shufflenet

A shufflenet with parameters p and k has $\aleph = kp^k$ nodes. It has k columns of p^k nodes each. Each node has p transmitters and receivers. The shufflenet multihop topology has several cycles of varying length. For example, in the 8-node example shown in Fig. 2, $(0,0) \rightarrow (1,0) \rightarrow (0,0)$ and $(0,0) \rightarrow (1,1) \rightarrow (0,2) \rightarrow (1,0) \rightarrow (0,0)$ are cycles. Thus, unrestricted routing could result in deadlocks. We define some notation and then present the routing function R followed by a proof of its correctness.

We define the following notation. The k columns are numbered 0 to $k-1$ and a wrap around occurs when a node in column $k-1$ sends a message to a node in column 0. The p^k nodes in a single column are numbered 0 to $p^k - 1$. Thus each of the n nodes in the shufflenet is uniquely identified by a tuple (c, r) where $c \in [0, k-1]$ and $r \in [0, p^k - 1]$ (see Fig. 2). A link (which is a wavelength in an optical network) between any two nodes is called a (physical) channel. Channels are partitioned into two sets: those that connect any node in column $k-1$ to any node in column 0 are called Wraparound channels; all other channels are called non-wraparound channels. The total number of physical channels in a shufflenet is kp^{k+1}. We number the channels from 0 to $kp^{k+1} - 1$ and denote them by c_j where j is the channel number (see Fig. 2).

Now, we map 3 virtual channels per physical channel in the following fashion. The 3 virtual channels associated with a physical channel c_j are denoted as c_{0j}, c_{1j} and c_{2j}, where $j \in [0, kp^{k+1} - 1]$. The (slack) buffer associated with each physical channel at a node is divided into 3 parts - one for each virtual channel. We eliminate the c_{0j} virtual channels for the Wraparound channels to remove cycles in the channel dependency graph. The new messages (generated by an external source (host)) entering a node are placed in the queue associated with virtual channel 0. The routing function $R : C \times N \rightarrow C$ then becomes:

$$R(c_{ij}, n_d) = \begin{cases} c_{0n}, & i = 0 \text{ and } c_n \text{ is not a wraparound channel} \\ c_{1n}, & i = 0 \text{ and } c_n \text{ is a wraparound channel} \\ c_{1n}, & i = 1 \text{ and } c_n \text{ is not a wraparound channel} \\ c_{2n}, & i = 1 \text{ and } c_n \text{ is a wraparound channel} \\ c_{2n}, & i = 2 \end{cases}$$

Theorem: The routing function R specified achieves deadlock-free routing in the shufflenet multihop topology when wormhole routing is used.

Proof: The routing function R specifies a partial ordering on the channels according to their subscripts: $c_{0j} < c_{1j} < c_{2j}, \forall j \in [0, kp^{k+1} - 1]$. Thus, there cannot be a cycle in the channel dependency graph. Hence, this routing function is deadlock-free.

The reason for splitting each physical channel into 3 virtual channels is the following: the diameter of the traditional shufflenet is $2k - 1$. This means that at most 2 wraparound channels can be encountered by any message. Since we have virtual channels with subscripts 1 and 2 that span $2k$ hops, it is possible to route all messages correctly to their destination irrespective of their node of origin. An example of the routing trees that result from using the routing function R is shown in Fig. 4. We show the routing tree for the node $(0, 0)$ of the eight node shufflenet shown earlier in Fig. 2, and it is easy to see that a message entering at this node (from external source) can be routed to any other node without encountering any buffer at any node twice.

5 Deadlock-free Routing for Bidirectional Shufflenet

A bidirectional shufflenet is a shufflenet which has full-duplex links[24, 19]. An example 8-node bidirectional shufflenet is shown in Fig. 2. As is evident, a bidirectional shufflenet uses twice as many receivers and transmitters as the shufflenet. A bidirectional shufflenet with parameters p and k has $\aleph = kp^k$ nodes. It has k columns of p^k nodes each. However, it has twice as many channels as the shufflenet. The number of physical channels in the bidirectional shufflenet is $2kp^{k+1}$. The diameter of bidirectional shufflenet is only $\lfloor \frac{3k-1}{2} \rfloor$ because of the ability to send traffic in either direction (forward or backward) on a link. Like the shufflenet, this virtual multihop topology also has cycles of various lengths. Hence, deadlocks are possible. We present two approaches to achieving deadlock-free routing for the bidirectional shufflenet. The first approach uses Up/Down routing to get deadlock-free routes while the second approach uses virtual channels to provide deadlock-free shortest routes for the bidirectional shufflenet topology.

5.1 Up/Down Deadlock-Free Routing

As we have bidirectional links between nodes in the bidirectional shufflenet topology, we can construct a spanning tree the links of which are bidirectional. This idea is similar to the one used in Autonet [26, 27]. The spanning tree can be constructed by any node. Once this node (the root of the spanning tree) has constructed the spanning tree[3], it sends a copy to all the other nodes in the bidirectional multihop topology. All nodes then use this spanning tree to decide the routes that they must use to get to desired destinations. The routing is done in an Up/Down fashion. All

[3]Actually, the spanning tree can be constructed in a distributed fashion too, as in the IEEE 802.1 spanning tree bridge algorithm[2, 11, 25].

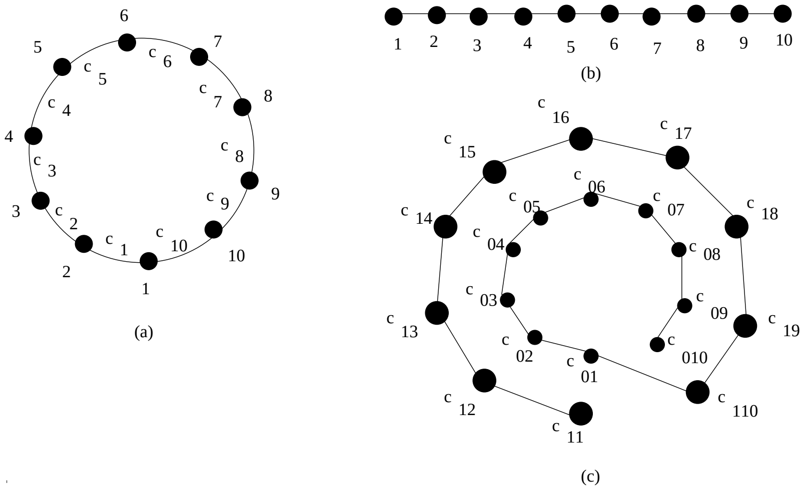

Figure 3: (a) A 10-node ring topology. (b) The spanning tree for Up/Down routing. (c) The virtual channel mapping for deadlock-free routing.

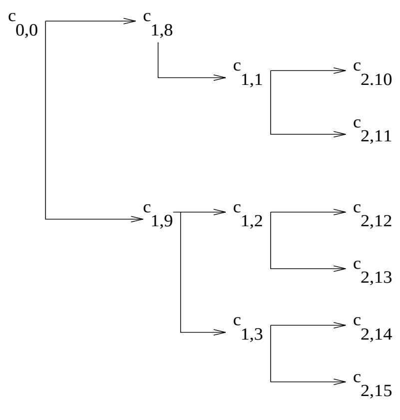

Figure 4: An example of a routing tree derived by applying the routing function R to node $(0, 0)$ of the eight node shufflenet shown earlier.

upward movements (toward the root) take place before any downward movements (away from the root). As we use a spanning tree derived from a strongly connected graph, each node can reach any other node; also by following the Up/Down scheme, no cycles are possible and thus no deadlocks are possible. Though a solution to the problem of deadlocks, this scheme suffers from the problem that routes between any two nodes need not always be the shortest possible. Further, this scheme may cause non-uniform loading of some links. The scheme using virtual channels presented next overcomes these deficiencies.

5.2 Deadlock-free Shortest Paths using Virtual Channels

Shortest paths in bidirectional shufflenet are of the type f, b, fb, bf, fbf and bfb, where f denotes one or more (a sequence of) hops in the forward direction and b denotes one or more hops in the backward direction [19]. Also it is possible to obtain shortest paths in which the total number of hops in a sequence of forward (backward) hops does not exceed k. All these patterns are substrings of fbfb. Now we present a mapping of virtual channels to physical channels that allows such routing to take place in a deadlock-free manner.

We adopt notation for the bidirectional shufflenet similar to the one defined for the unidirectional shufflenet. The k columns are numbered 0 to $k-1$ with a wrap around in the forward direction occurring when a node in column $k-1$ sends a message to a node in column 0. Similarly a wrap around in the backward direction occurs when a node in column 0 sends a message to a node in column $k-1$. As before each of the n nodes in the topology is uniquely identified by a tuple (c, r) where $c \in [0, k-1]$ and $r \in [0, p^k - 1]$. For the bidirectional shufflenet we define Forward Wraparound channels to be those that connect any node in column $k-1$ to any node in column 0; those channels that connect any node in column 0 to any node in $k-1$ are called Backward Wraparound channels. All other channels are called non-wraparound channels. The total number of physical channels in a bidirectional shufflenet is $2kp^{k+1}$. We number the channels 0 to $kp^{k+1}-1$ in the forward direction and 0 to $kp^{k+1}-1$ in the backward direction. Thus there are two channels c_{fj} and c_{bj} for each channel number j. The subscripts f and b refer to the forward and backward direction respectively.

As the total number of hops in a sequence in any one direction does not exceed k, Wraparound channels (Backward and Forward) can be encountered at most once. Thus, we associate 4 virtual channels with each physical channel (2 for each possible sequence of hops in that direction[4]) in the following fashion. The 4 virtual channels associated with a physical channel c_{fj} are denoted as c_{0fj}, c_{1fj}, c_{2fj} and c_{3fj}. The 4 virtual channels associated with a physical channel c_{bj} are denoted

[4]For example, fbf has two sequences of hops in the forward direction, each of which can be at most k hops.

as c_{0bj}, c_{1bj}, c_{2bj} and c_{3bj}. In both cases, $j \in [0, kp^{k+1}-1]$. Virtual channels numbered 0 and 1 are used for the first sequence of hops in that direction as per **fbfb** (see Fig. 5). For the second sequence of hops in the same direction, we use the virtual channels numbered 2 and 3. Thus, the slack buffer associated with each physical channel at a node is divided into 4 parts (one per virtual channel). We eliminate the c_{0fc} and c_{0bc} virtual channels for the Forward Wraparound channels and the Backward Wraparound channels respectively in order to eliminate cycles from the channel dependency graph. The new messages entering a node are placed in the c_{0fc} channel if the path is of type **fb** or **fbf**. If the path is of the type **f**, then the new message is placed in channel c_{2fc}. If the path is of the type **b**, then the new message is placed in channel c_{2bc}. Otherwise, it is placed in the c_{0bc} channel if the path is of the type **bf** or **bfb**. We assume that this information (path type) is known at the time the new message becomes available at the node. The routing function $R : C \times N \rightarrow C$ then becomes:

$$R(c_{ifj}, n_d) \quad = \quad \begin{cases} c_{0fn}, & i = 0 \text{ and } c_{fn} \text{ is not a forward wraparound channel} \\ c_{1fn}, & i = 0 \text{ and } c_{fn} \text{ is a forward wraparound channel} \\ c_{1fn}, & i = 1 \\ c_{0bn}, & i = 0,1 \text{ (beginning of a } \mathsf{b} \text{ sequence) and} \\ & \qquad c_{bn} \text{ is not a backward wraparound channel} \\ c_{1bn}, & i = 0,1 \text{ (beginning of a } \mathsf{b} \text{ sequence) and} \\ & \qquad c_{bn} \text{ is a backward wraparound channel} \\ c_{2fn}, & i = 2 \text{ and } c_{fn} \text{ is not a forward wraparound channel} \\ c_{3fn}, & i = 2 \text{ and } c_{fn} \text{ is a forward wraparound channel} \\ c_{3fn}, & i = 3 \\ c_{2bn}, & i = 2,3 \text{ (beginning of a } \mathsf{b} \text{ sequence) and} \\ & \qquad c_{bn} \text{ is not a backward wraparound channel} \\ c_{3bn}, & i = 2,3 \text{ (beginning of a } \mathsf{b} \text{ sequence) and} \\ & \qquad c_{bn} \text{ is a backward wraparound channel} \end{cases}$$

$$R(c_{ibj}, n_d) = \begin{cases} c_{0bn}, & i = 0 \text{ and } c_{bn} \text{ is not a backward wraparound channel} \\ c_{1bn}, & i = 0 \text{ and } c_{bn} \text{ is a backward wraparound channel} \\ c_{1bn}, & i = 1 \\ c_{2fn}, & i = 0,1 \text{ (beginning of a } \mathsf{f} \text{ sequence) and} \\ & \quad c_{fn} \text{ is not a forward wraparound channel} \\ c_{3fn}, & i = 0,1 \text{ (beginning of a } \mathsf{f} \text{ sequence) and} \\ & \quad c_{fn} \text{ is a forward wraparound channel} \\ c_{2bn}, & i = 2 \text{ and } c_{bn} \text{ is not a backward wraparound channel} \\ c_{3bn}, & i = 2 \text{ and } c_{bn} \text{ is a backward wraparound channel} \\ c_{3bn}, & i = 3 \end{cases}$$

The proof that R is a deadlock-free routing function is similar to the one given earlier and is omitted. Instead, a diagram showing how the virtual channels are connected in a graph is shown in Fig. 5.

6 Protocols for Serving Virtual Channels

The use of virtual channel mappings implies the existence of protocols for serving such channels at each node. These protocols are required in order to map several virtual channels on a single physical channel. The virtual channels can be served in a Time Division Multiplexed fashion, or some kind of (preemptive or non-preemptive) priority scheme. We present 3 protocols for serving virtual channels and compare them via simulation for the bidirectional shufflenet.

6.1 Protocols

The following protocols can be used to serve virtual channels.

a. **Fixed Assignment (FA)**: In this protocol, the virtual channels are each served in round-robin manner for a fixed duration of time called a slot (like TDM). This approach guarantees that each worm will be served once in V slots, if V virtual channels share a single physical channel. However, the latency experienced by each worm is quite large (the larger the worm, the higher the delay) because the worm gets access to the channel for only $\frac{1}{V}$ slots.

b. **Preemptive Priority (PP)**: In this protocol, we give priority to worms using higher numbered virtual channels since they are closer to the destination. The priority is given in a preemptive resume fashion. If a worm using virtual channel numbered x is being served and there is no arrival of worms on virtual channels numbered higher than x, then this worm is served until either the worm has been completely transmitted or the worm is blocked. In either case the next lower virtual channel with a worm is served. While the worm on virtual channel x is being served, if a worm arrives on a higher numbered virtual channel, then the newly arrived worm preempts the one being served. Transmission of the worm using virtual channel x will resume when there are no more worms to serve in virtual channels numbered higher than x.

c. **Non-preemptive Priority (NP)**: This protocol is similar to the Preemptive Priority protocol described above. The difference is that if a worm on virtual channel x is being served and a worm arrives at a higher numbered virtual channel, the new worm does not preempt the worm being served. When the worm being served has been completely transmitted or when it gets blocked, we pick the highest numbered virtual channel which has a worm waiting to be transmitted for service.

6.2 Simulation Model

We chose a 24-node ($p = 2, k = 3$) bidirectional shufflenet topology. Wormhole routing was assumed. Worm lengths were geometric with average worm lengths being 2000, 4000 and 40000 bytes. This was done to study the effect of small worms and somewhat longer worms. The arrival of worms at a host was modeled as a poisson process with average interarrival time being a function of average worm length and offered load. Hop-by-hop backpressure flow control was assumed for the bidirectional shufflenet – hence, there was no worm loss.

The transmission speed of the transmitters was assumed to be 1 Gbps. The time to transmit one byte (i.e. $8ns$) was assumed to be our unit of time. The optical link lengths were assumed to be 1 mile (1.6km) long. The resulting one-way propagation delay was a 1000 time units (i.e. $8\mu s$). The queues at the input (slack buffers) to the nodes were 5000 bytes – two and a half times the round-trip propagation delay. The throughput is normalized (i.e. throughput per node). The delays are calculated on a worm basis.

6.3 Performance Comparison

Throughputs achieved by employing these protocols as well as by using Up/Down (UD) deadlock-free routing for the 24-node bidirectional shufflenet are shown in Figs. 6, 7 and 8. For reference we also plot the results when unrestricted shortest path routing (UR) (potential deadlocks) was used.

It can be seen that unrestricted routing leads to deadlocks while the other protocols do not exhibit this behavior. Of the deadlock-free protocols, Fixed Assignment (FA) protocol for serving virtual channels gives the worst performance. This is due to the fact that a worm gets served only

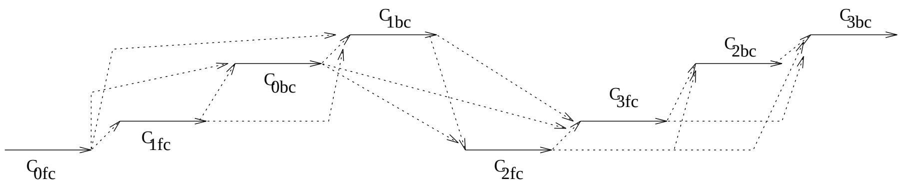

Figure 5: A graph showing the possible sequences of virtual channels encountered for the shortest path deadlock-free routing in a bidirectional shufflenet. The solid arrows represent one or more sequences of hops along virtual channels with the same numbers, the dashed arrows show possible changes in virtual channel numbers. It is easy to see that there are no cycles in this graph.

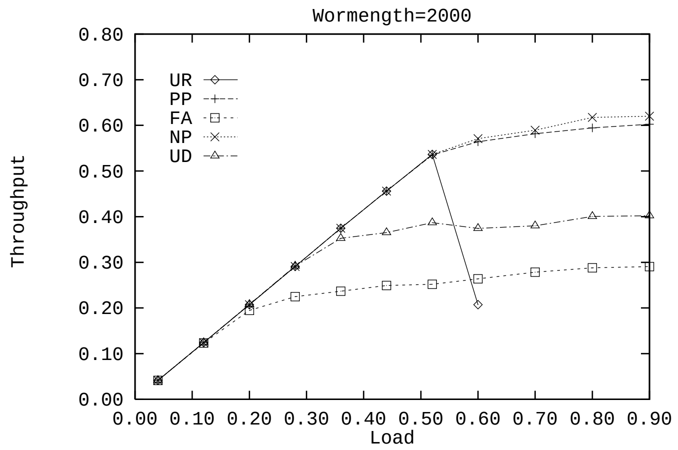

Figure 6: Throughput versus Offered Load for the 24 node ($p = 2$) bidirectional shufflenet. Average worm length, l, is 2000 bytes. The curves compare Fixed Assignment (FA), Preemptive Priority (PP), Nonpreemptive Priority (NP) protocols for virtual channels with Up/Down routing (UD) and unrestricted routing (UR) protocols.

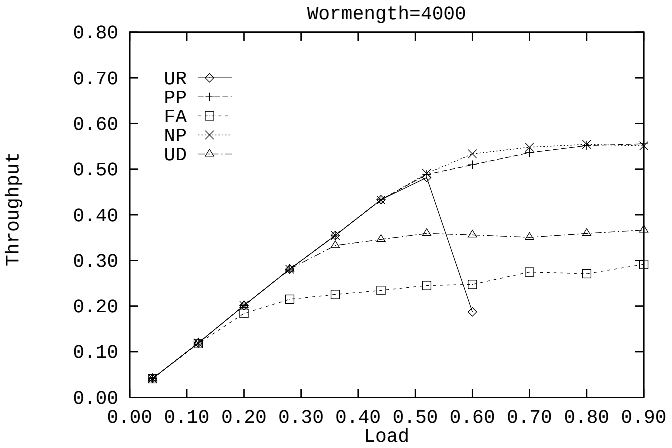

Figure 7: Throughput versus Offered Load for the 24 node ($p = 2$) bidirectional shufflenet. Average worm length, l, is 4000 bytes. The curves compare Fixed Assignment (FA), Preemptive Priority (PP), Nonpreemptive Priority (NP) protocols for virtual channels with Up/Down routing (UD) and unrestricted routing (UR) protocols.

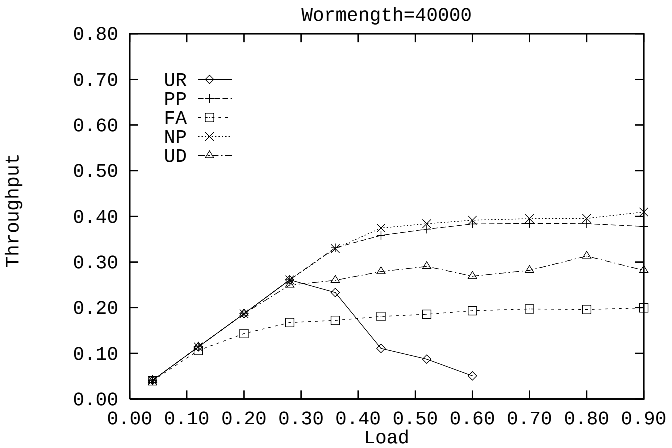

Figure 8: Throughput versus Offered Load for the 24 node ($p = 2$) bidirectional shufflenet. Average worm length, l, is 40000 bytes. The curves compare Fixed Assignment (FA), Preemptive Priority (PP), Nonpreemptive Priority (NP) protocols for virtual channels with Up/Down routing (UD) and unrestricted routing (UR) protocols.

$\frac{1}{V}$ of the time, where V is the number of virtual channels per physical channel, and some slots may go unused. Up/Down routing (UD) does not fare much better. This could be due to a combination of two factors – some links may be more heavily loaded than others and shortest paths are not available for all traffic. Both Preemptive Priority (PP) and Nonpreemptive Priority (NP) perform similarly and consistently give the highest throughput. Also, the throughput is the same as for unrestricted shortest path routing till deadlocks occur when unrestricted routing is used. Thus, no throughput performance is lost by employing the PP and NP protocols. It can also be seen that the maximum throughput achieved reduces with worm length. A point to note is that though deadlock-freedom is achieved, using virtual channels involves some costs – increased buffer space and some complexity in protocol.

See Figs. 9, 10 and 11 for delay results. As expected the deadlock-free protocols FA and UD perform much worse than the PP and NP protocols. The higher delay for FA protocol occurs due to the TDM like characteristic of the protocol. The higher delay of the Up/Down routing scheme is due to non-uniform loading of links and not being able to use shortest paths at all times.

The interesting result here is that the Non-preemptive protocol performs slightly better than the Preemptive protocol. This can be explained as follows: when a worm is preempted, it may be the case that the preempted worm has already reached the destination; thus, the virtual channels assigned to the worm will be drained by the destination leaving empty virtual channels allocated to a worm that is blocked by preemption. These virtual channels cannot be used by other worms until the preempted worm is served again. Since the 24 node bidirectional shufflenet has only 3 columns, these 'holes' may not be very large thus showing up as only minor differences on the graphs.

7 Deadlock-Free Routing in SSN

The Supercomputer SuperNet (SSN) has a two-level hierarchical architecture – high speed electronic mesh networks and a passive star coupler WDM optical network – as described earlier. For the optical network, we have just described how to achieve deadlock-free routing for the shufflenet multihop virtual topology – both unidirectional and bidirectional. For the high speed electronic network, deadlock-free routing can be achieved by employing the technique of the Up/Down routing on a spanning tree of the network (as was used in Autonet [26]). Having deadlock-free routing schemes in the different parts of the network, however, does not guarantee deadlock freedom for the network as a whole. An example is shown in Fig. 12. In this section, we assume that the optical backbone of SSN has the bidirectional shufflenet multihop virtual topology.

Several alternatives for having deadlock free routing across the network exist. One way is to use Up/Down routing on a single spanning tree for the whole network. In this approach, either one

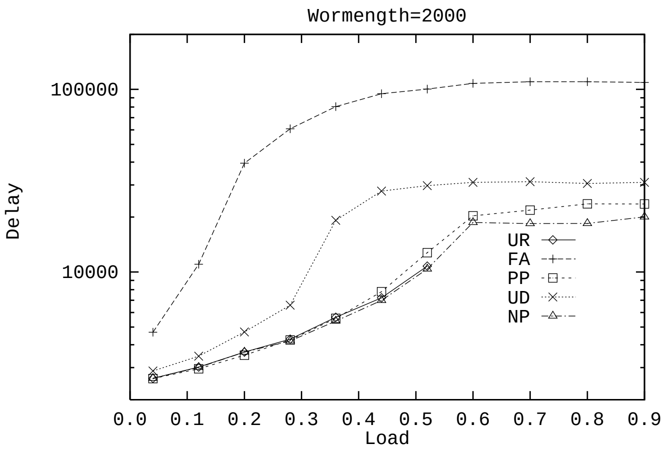

Figure 9: Delay versus Offered Load for the 24 node ($p = 2$) bidirectional shufflenet. Average worm length, l, is 2000 bytes. The curves compare Fixed Assignment (FA), Preemptive Priority (PP), Nonpreemptive Priority (NP) protocols for virtual channels with Up/Down routing (UD) and unrestricted routing (UR) protocols.

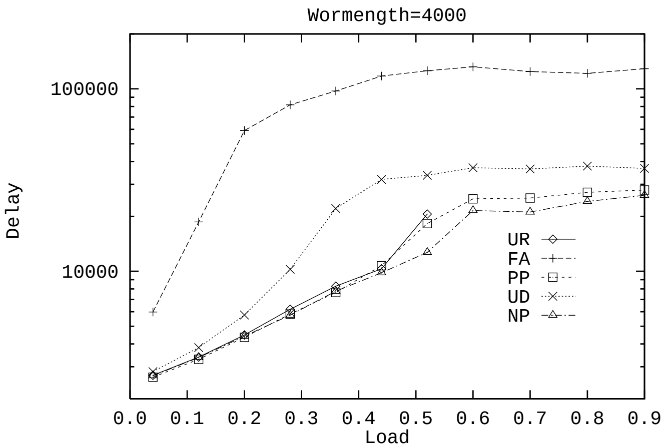

Figure 10: Delay versus Offered Load for the 24 Node ($P = 2$) bidirectional shufflenet. Average worm length, L, is 4000 bytes. The curves compare Fixed Assignment (FA), Preemptive Priority (PP), Nonpreemptive Priority (NP) protocols for virtual channels with Up/Down routing (UD) and unrestricted routing (UR) protocols.

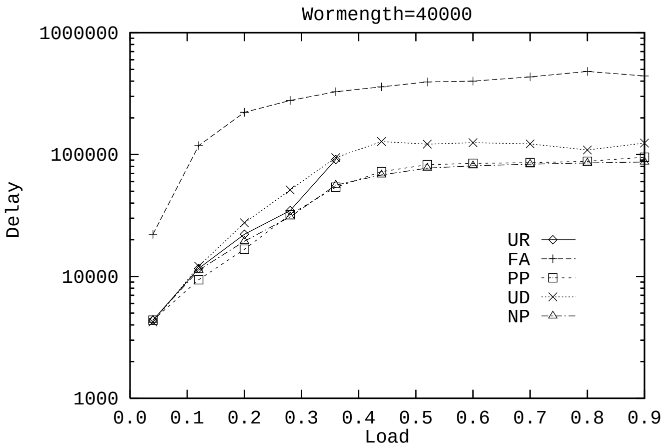

Figure 11: Delay versus Offered Load for the 24 node ($p = 2$) bidirectional shufflenet. Average worm length, l, is 40000 bytes. The curves compare Fixed Assignment (FA), Preemptive Priority (PP), Nonpreemptive Priority (NP) protocols for virtual channels with Up/Down routing (UD) and unrestricted routing (UR) protocols.

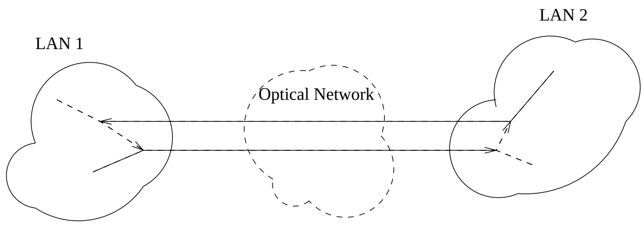

Figure 12: An example showing that even if each part of the network is deadlock-free, it does not follow that the whole network is deadlock-free. The solid lines represent a worm destined to a remote LAN that are in a remote LAN; the dashed lines represent local/remote traffic still in the origin LAN. The arrows indicate where blocking occurs.

of the OCIs or one of the stations on the net can behave as the root of the spanning tree. This alternative works correctly since we have bidirectional links in SSN. However, this approach has the usual drawback of not providing the shortest paths at all times and non-uniform loading of some links.

The virtual channel mapping method that we have used for the multihop virtual topologies, can also be used for the high speed electronic mesh network. We assume that one of the several mappings presented in [5] can be used to provide deadlock free routing in the electronic high speed LAN. In other words, we extend the virtual channel mapping method across the whole network. Namely, we achieve deadlock-free routing in the electronic LANs by dedicating separate sets of virtual channels to origin and remote LAN traffic[5]. Thus, origin LAN traffic does not interfere with remote LAN traffic. Note that traffic to a remote destination still travels on the local virtual channels in the origin LAN. Only in a remote LAN (or, any LAN other than the origin) does the worm travel along remote virtual channels. Thus, there is no possibility of circular wait (again we have a partial order on the virtual channels) and hence the situation shown earlier in Fig. 12 cannot occur. Thus, deadlock free routing across the whole SSN network can be achieved. Unfortunately, Myrinet LANs currently do not support virtual channels with separate slack buffers for each virtual channel.

A third solution consists of requiring that the OCI in each electronic LAN (subnet) be the root of the spanning tree used to define the Up/Down links for that subnet. Then, deadlock prevention can be achieved for the subnet with Up/Down routing as discussed earlier. Bidirectional shufflenet deadlock prevention can be achieved with any of the schemes discussed in section 5. For deadlock-free routing across the whole network, it suffices to require that remote traffic (i.e. a connection between two distinct subnets) be routed Up in the origin LAN and Down in the destination LAN. Of course, choosing the OCI as the root of the spanning tree for the subnet is key to this solution. Deadlock prevention is evident from inspection of Fig. 12, where conflict between local and remote worms could not exist with this new rule. Note that this deadlock free routing scheme can be easily extended to the case in which an electronic LAN is connected to the optical backbone via multiple OCIs. In this case, one of the OCIs is elected as the root of the spanning tree (using, for example, a distributed procedure similar to the IEEE 802.1 spanning tree bridge algorithm). Each OCI can then handle remote traffic to and from sources and destinations in its subtree. Of course, the 'root' OCI could handle all remote traffic by itself. However, hosts will be assigned to different OCIs using appropriate criteria (e.g. shortest distance) in order to balance the traffic across the

[5]Here, traffic originating within a LAN destined to a host in the same LAN is called local traffic. If the destination is a host on a remote LAN then the traffic is called remote traffic. If the worm is traveling through the LAN to which the origin host belongs then it is said to be in the origin LAN otherwise it is said to be in a remote LAN.

electronic LAN, OCI and the bidirectional shufflenet.

One important advantage of the last solution (over the previous two) is that it allows the independent choice of the most cost effective deadlock free routing scheme for each level of the SSN architecture.

8 Conclusions

Deadlocks in wormhole routing networks occur when a state is reached such that no messages can be forwarded towards their destination and without external intervention such a state will persist forever. Wormhole routing networks differ from traditional store-and-forward networks in that two or more worms cannot be interleaved. Once resources (slack buffers) are allocated to a worm, they can be released only when the tail of the first worm is received. There is blocking in such networks due to non-availability of resources (buffers). Hence, deadlocks occur.

We have built upon previous work by [5] and provided virtual channel mappings for the shufflenet and the bidirectional shufflenet which are provably deadlock-free. For the bidirectional shufflenet architecture, we presented two schemes by which deadlock-free routing is possible. The Up/Down routing scheme is the simplest one but has certain drawbacks (not necessarily providing shortest paths and leading to non-uniform loads). The second scheme based on virtual channels provides deadlock-free shortest path routing.

We have presented and compared 3 protocols for serving virtual channels via simulation on the 24-node bidirectional shufflenet. The Non-preemptive Priority and Preemptive Priority protocols perform favorably. The Fixed Assignment protocol and the Up/Down routing schemes do not perform well.

In addition, we have addressed the issue of providing deadlock-free routing in networks with a two-level architecture (as in SSN) – a passive star coupler WDM optical backbone network connecting together several high speed electronic mesh networks – and have shown how to resolve this issue. An interesting conclusion of our study is that deadlock freedom over the entire network can be achieved even while employing different deadlock prevention schemes at the different levels of the network architecture.

References

[1] A. S. Acampora. A Multichannel Multihop Local Lightwave Network. In *Proceedings of GLOBECOM 87*, pages 1459–1467, November 1987.

[2] F. Backes. Transparent Bridges for Interconnection of IEEE 802 LANs. *IEEE Network*, 2(1):5–9, January 1988.

[3] F. Borgonovo and E. Cadorin. Locally-Optimal Deflection Routing in the Bidirectional Manhattan Network. In *Proceedings of INFOCOM 90*, pages 458–464, June 1990.

[4] C. A. Brackett. Dense Wavelength Division Multiplexing Networks: Principles and Applications. *IEEE Journal on Selected Areas in Communications*, 8:948–964, August 1990.

[5] W. J. Dally and C. L. Seitz. Deadlock-Free Message Routing in Multiprocessor Interconnection Networks. *IEEE Transactions on Computers*, C-36(5):547–553, May 1987.

[6] M. Eisenberg and N. Mehravari. On the Performance of Multihop, Multichannel Lightwave Networks and Non-uniform Traffic. *IEEE Journal on Selected Areas in Communications*, 6(7):1063–1078, August 1988.

[7] M. Gerla, B. Kannan, and P. Palnati. Protocols for an Optical Star Interconnect for High Speed Mesh Networks. In *Proceedings of INFOCOM 95*, pages 146–153, April 1995.

[8] M. Gerla and L. Kleinrock. Flow Control: A Comparative Survey. *IEEE Transactions on Communication*, 28(4):553–574, April 1980.

[9] I. S. Gopal. Prevention of Store-and-Forward Deadlock in Computer Networks. *IEEE Transactions on Communications*, COM-33(12):1258–1264, December 1985.

[10] K. D. Gunther. Prevention of Deadlocks in Packet-switched Data Transport Systems. *IEEE Transactions on Communications*, COM-29(10):512–524, October 1981.

[11] J. Hart. Extending the IEEE 802.1 MAC Bridge Standard to Remote Bridges. *IEEE Network*, 2(1):10–15, January 1988.

[12] M. Hluchyj and M. Karol. Shufflenet: An Application of Generalized Perfect Shuffles to Multihop Lightwave Networks. *IEEE/OSA Journal of Lightwave Technology*, 9(10):1386–1397, October 1993.

[13] R. C. Holt. Some Deadlock Properties of Computer Systems. *Computing Surveys*, 4(3):179–196, September 1972.

[14] M. Karol and S. Shaikh. A Simple Adaptive Routing Scheme for Shufflenet Multihop Lightwave Networks. In *Proceedings of GLOBECOM 88*, pages 1640–1647, 1988.

[15] M. Karol and S. Shaikh. A Simple Adaptive Routing Scheme for Congestion Control in Shufflenet Multihop Lightwave Networks. *IEEE Journal on Selected Areas in Communications*, 9(7):1040–1051, September 1991.

[16] P. Kermani and L. Kleinrock. Virtual Cut-through: A New Computer Communication Switching Technique. *Computer Networks*, 3(3):267–286, September 1979.

[17] L. Kleinrock, N. Bambos, J. Cong, E. Gafni, M. Gerla, and L. Bergman. Optimic: A Scalable Distributed, All-Optical Terabit Network. *Journal of High Speed Networks.* To appear in Special Issue on All-Optical Networks.

[18] L. Kleinrock, M. Gerla, N. Bambos, J. Cong, E. Gafni, L. Bergman, J. Bannister, et al. The Supercomputer Supernet Testbed: A WDM Based Supercomputer Interconnect. *joint issue of IEEE JSAC and IEEE/OSA JLWT*, 1995. To Appear in Special Issue on Multiple Wavelength Optical Technologies and Networks.

[19] E. Leonardi, P. Palnati, and M. Gerla. Analysis of the Bidirectional Shufflenet Multihop Topology. Submitted for publication.

[20] N. F. Maxemchuk. Routing in the Manhattan Street Network. *IEEE Transactions on Comunications*, COM-35(5):503–512, May 1987.

[21] P. M. Merlin and P. J. Schweitzer. Deadlock Avoidance in Store-and-Forward Networks - I: Store-and-Forward Deadlock. *IEEE Transactions on Communications*, COM-28(3):345–354, March 1981.

[22] B. Mukherjee. WDM-based Local Lightwave Networks Part I: Single-Hop Systems. *IEEE Network*, 6(3):12–27, May 1992.

[23] B. Mukherjee. WDM-based Local Lightwave Networks Part II: Multi-Hop Systems. *IEEE Network*, 6(4):20–32, July 1992.

[24] P. Palnati, E. Leonardi, and M. Gerla. Bidirectional Shufflenet: A Multihop Topology for Backpressure Flow Control. In *Proceedings of* 4^{th} *International Conference on Computer Communications and Networks*, pages 74–81, September 1995.

[25] R. Perlman. An Algorithm for Distributed Computation of a Spanning Tree in an Extended LAN. In *Proceedings of 9th Data Communication Symposium*, September 1985.

[26] T. L. Rodeheffer. Experience with Autonet. *Computer Networks and ISDN Systems*, 25(6):623–629, January 1993.

[27] M. D. Schroeder, A. D. Birrell, M. Burrows, H. Murray, R. M. Needham, T. L. Rodeheffer, E. H. Satterthwaite, and C. P. Thacker. Autonet: A High-Speed, Self-Configuring Local Area Network Using Point-to-Point Links. *IEEE Journal on Selected Areas in Communications*, 9(8):1318–1335, October 1991.

[28] S. Toueg. Deadlock- and Livelock-free Packet Switched Networks. In *Proc. ACM Symposium on Theory of Computing*, pages 94–99, April 1980.

[29] S. Toueg and J. D. Ullman. Deadlock-free Packet Switched Networks. In *Proc. ACM Symposium on Theory of Computing*, pages 89–98, April 1979.