

Deflection Routing in Slotted Self-Routing Networks With Arbitrary Topology

Chun-Yin Li, *Member, IEEE*, P. K. A. Wai, *Senior Member, IEEE*, Xiao Chun Yuan, and Victor O. K. Li, *Fellow, IEEE*

Abstract—A deflection routing scheme for small to medium size future all-optical networks with arbitrary topologies is proposed. The proposed scheme assumes only single-bit all-optical processing and no buffers. The primary output selection and the alternate output choices by a packet at each node are encoded in the packet header in order to reduce the signal processing requirement. Additional features such as priority and time-to-live fields have also been defined. The performance of the deflection routing scheme is studied using the AT&T North America OC-48 optical fiber network topology.

Index Terms—All-optical networks, deflection routing, packet switching.

I. INTRODUCTION

THE advantages of all-optical packet-switched networks in which both the data and control signals remain in the optical domain from the source node to the destination node are well-known. All-optical processing of the address headers, however, is difficult to realize because of the limited optical signal processing power available and the lack of efficient optical buffers. Although simple optical logic gates such as AND, OR, NOR, and NAND that can perform bit-by-bit processing as well as high-speed all-optical multiplexing and demultiplexing of optical signals are demonstrated optically, complex optical signal processing is not available yet [1]–[3]. Fiber delay line based buffers have fixed delay [4]. Once sent into a delay line, the signals cannot be accessed until they emerge from the fibers. Unless there is significant technology breakthrough, the routing schemes for all-optical packet-switched networks in the near future must therefore require minimal processing and buffering.

With these constraints in mind, we proposed a self-routing address scheme that requires only on-the-fly bitwise processing [5]. Thus, the address headers can be processed using available bitwise all-optical logic gates. Recently, Wai *et al.* implemented the proposed self-routing address in a $1 \times N$ all-optical packet switch with all-optical header processing and packet

switching at 10 Gb/s [6]. Despite its simplicity, the proposed address scheme has a number of advantages. It can be applied to networks with arbitrary topology. The paths between any two nodes can be chosen arbitrarily. The proposed address scheme used fixed routing but since the scheme allows multiple paths to be defined between two nodes, the multiple paths can be used in alternate routing schemes in case of network failure or network congestion. The proposed address scheme can also be adapted to a hierarchical address structure. The major drawback of the address scheme is the length of the address. In its basic form, the length of the address equals the sum of the number of outputs of all the nodes in a network. Applications of the proposed address scheme are therefore limited to small to medium size networks like the current backbone networks. For example, there are 27 nodes in the AT&T IP backbone network [7] and 31 nodes in the European research backbone network GEANT [8]. The average node output degree in these networks is about 3. In these cases, an address will be about 100 bits long.

The proposed self-routing address scheme does not deal with the packet contention problem. Packet contention occurs when two or more packets request the same output at a node. In traditional packet-switched networks, a node stores the conflicting packets temporarily in buffers so that all packets are optimally routed over the shortest path. In other words, both complex signal processing and buffering are used to resolve the conflicts. Obviously such a strategy violates the constraints set forth in the development of the self-routing address scheme of [5]. Whether a packet routing scheme can be implemented all-optically in the near future depends on the detail design of the scheme and the optical signal processing power available at the time. The use of buffers, however, can be avoided using deflection routing [9]. Deflection routing also reduces the optical signal processing required because buffer management is not required. In this paper, we describe how deflection routing can be applied to the self-routing address scheme we proposed in [5]. We improve the deflection routing performance by encoding the deflection output choices in the packet address header. Similar to [5], we trade the processing requirement with the length of the address. The increase in the length of the address depends on the average node output degree and the implementation detail. Thus, the deflection routing address scheme would still be applicable to small and medium size all-optical networks.

The paper is organized as follows. In Section II, we give a brief review of the proposed self-routing scheme. Section III discusses how deflection routing can be implemented in the new address scheme. Section IV describes how other features such as priority classes and time-to-live can be implemented in the proposed address scheme. Even though the packet routing

Manuscript received January 2, 2003; revised February 24, 2004. This work was supported in part by the Areas of Excellence Scheme established under the University Grants Committee of the Hong Kong Special Administrative Region, China (Project no. AoE/E-01/99). Additional support was provided by a grant from The Hong Kong Polytechnic University (Project Number A-PD69).

C.-Y. Li and P. K. A. Wai are with the Photonics Research Center and Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong SAR, China (e-mail: enli@polyu.edu.hk; enwai@polyu.edu.hk).

X. C. Yuan and V. O. K. Li are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong SAR, China (e-mail: xcyuan@eee.hku.hk; vli@eee.hku.hk).

Digital Object Identifier 10.1109/JSAC.2004.833852

is significantly simplified, realization of the proposed deflection routing using single-bit logic gates without buffers remain a challenge. However, we are confident that the proposed deflection routing address algorithm can be implemented all-optically in the near future considering that implementation of the proposed self-routing address has already been demonstrated in [6] and it is already possible to cascade the single-bit logic gates to perform relatively complex logic operations such as the all-optical generation of a pseudoerror signal from a 10-Gb/s signal demonstrated in [10]. In Section V, we describe an original design of a 2×2 all-optical deflection routing node using the all-optical crossbar switches demonstrated by Glesk *et al.* [11]. We note that existing performance models cannot be applied directly to analyze the proposed deflection routing networks because the network topologies are arbitrary, the traffic distribution is nonuniform, and the nodes can have degrees of 3 or more. In Section VI, we describe an analytical model to study the performance of the deflection routing algorithm. By grouping packets with the same preferred output irrespective of their destinations, we are able to significantly reduce the computational complexity in determining the deflection probabilities in each node. In Section VII, we study the performance of the deflection routing algorithm using the AT&T North America OC-48 optical fiber network topology as an example [7]. We conclude in Section VIII.

II. SELF-ROUTING ADDRESS SCHEME

In the proposed self-routing address scheme [5], each output of all the nodes in a network is associated with a bit in the address header. The header processing unit of a node processes the address of an input packet bit-by-bit and set the packet routing unit to route the packet to the output with its corresponding address bit set to **1**. The paths between any two nodes are fixed. The address of a node encodes a unique path from any other node to the node itself. The paths contained in each address must satisfy the following condition.

Condition 1: If the paths from two different nodes to the same destination node meet at an intermediate node, the subsequent ports and nodes used by the two paths must be the same.

The basic self-routing addresses of the nodes in a network can be constructed as follows. We consider a network made up of N nodes and L links. All links are assumed to be bidirectional. Each node is arbitrarily labeled from 1 to N . The links connecting to each node are also arbitrarily labeled from 1 to K_i where K_i is the number of links connected to the i th node. We have $\sum_{i=1}^N K_i = 2L$.

The address of a node contains H bits, where $H = 2L$. Each address is divided into N fields. Each field corresponds to one node in the network. The i th field of an address contains K_i bits. The i th address field of node i is set to zero. For the j th address field of node i , $j \neq i$, one and only one of the K_j bits, say the x th bit, is set to **1**. The other bits at the j th address field are set to zeros. A nonzero entry at the x th bit of the j th address field means that node j will forward a packet with such an address to the x th output.

When a node receives a packet, it only processes the address field corresponding to the node itself. A node recognizes that a

packet has arrived at the destination if the corresponding address field is all zeros. Otherwise, it forwards the packet to the local output as specified. In a node address, there is a total of $(N - 1)\mathbf{1}$ bits out of the H bits. Bits in an address field are set to **1** depending on the paths defined.

The self-routing address is a set of routing instructions for the intermediate nodes. A destination node has to periodically collect the network topology information and broadcast the updated self-routing addresses to other nodes [5]. An outdated self-routing address would cause packets to be misrouted. Hence, a time-to-live field similarly to the one described in Section IV should be included in the packet header. Since how to ensure the validity of the routing information is a common problem in all packet-switched networks, we assume that the problem has been either handled by upper layer protocols or that the network is rather static.

III. DEFLECTION ROUTING

In deflection routing, typically a node determines the optimal paths from itself to the destination nodes of the arriving packets. Packets requesting the same outputs are in conflict and they are prioritized according to some deflection criteria, for example age or distance-to-destination. The packets with higher priorities are routed optimally to the shortest paths while those with lower priorities are deflected to outputs that may lead to longer path lengths. Thus, a node must have the capability to determine the optimal paths of arriving packets and compare the deflection criteria of conflicting packets. The information required to determine the optimal path and deflection criterion of a packet can be stored either in the nodes or in the packet header. The former approach requires all-optical memory in the node and the latter requires long address headers. While all-optical implementation of both approaches are in principle possible, neither of them is very practical.

In applying deflection routing to the proposed self-routing scheme for networks with arbitrary topology, it is important to simplify the signal processing requirement of the deflection algorithm as much as possible so that it can be implemented using available optical logic devices. Recall that the optimal output choice of a packet is already encoded in the proposed self-routing address headers; therefore, a node does not need to determine the optimal paths for the arriving packets. We do not require the node to determine the alternate output choices for a packet in the event of deflection. The alternate output choices are encoded in the address header of the packet instead. To further simplify the signal processing requirement, a node will not sort the packets in conflicts either. In case of conflicts in the first output choice, one of the conflicting packets is selected at random and allowed to use its preferred output. The rest of the packets will try to use their respective second choices of outputs. Further conflicts in the second choice of outputs are resolved in the same way as that of the first choice. Successive choices of outputs can be arranged such that each gives the best alternative path, for example the next shortest path, when compared to the previous deflection choices. The delay due to deflections can therefore be minimized.

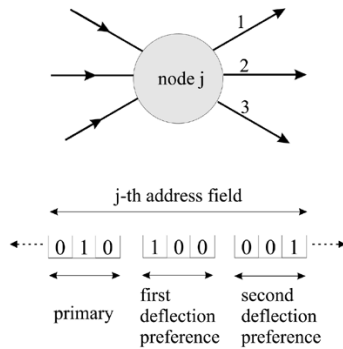


Fig. 1. Example of the address field of a 3×3 node with full deflection preference fields added.

Alternate output choices of a packet can be implemented in the proposed self-routing addresses by adding deflection preference fields after each address field. The deflection preference fields have the same number of bits as the corresponding address field. Similar to the original address field, each bit position in the deflection preference fields is associated with an output of the node. Each deflection preference field identifies an output that is different from all the previous choices of outputs by the packet. For a node with k outputs, there can be at most $k - 1$ deflection preference fields. The number of deflection preference fields used at each node need not be the same. Fig. 1 shows an example of the address field corresponding to node j in a network. Node j has three outputs, which are labeled as shown in Fig. 1. Full deflection preferences are specified. The packet has output 2 as its primary choice and outputs 1 and 3 as its first and second deflection preferences, respectively.

We assume that the network is slotted so that packets at all inputs arrive at each node simultaneously. We also assume that each node has the same number of input and outputs. A node assigns its outputs to the arriving packets in successive rounds in accordance to the order of the output choices indicated in the packet's deflection preference fields. In the first round of output assignment, the node considers the primary address fields of arriving packets. The node assigns the uncontended outputs to the packets requesting them. The node assigns each of the contended outputs to one of the contending packets at random. In the second round of output assignment, the node processes the first deflection preference fields of all the packets that have not been assigned outputs in the first round. These are the packets that lose in their respective contentions. If the output indicated at the first deflection preference field of a packet has already been assigned in the previous round, the packet will not be assigned an output in this round. If more than one of the remaining packets request the same output in their first deflection preference field, the contended outputs are assigned to one of the contending packets at random. The uncontended outputs are assigned to the packets requesting them.

If some packets have not been assigned an output after the first two rounds, the node will start the third round of output assignment by processing the second deflection preference fields of these packets. The procedure will be repeated until all the packets are assigned an output or the deflection preference fields are exhausted. In the latter case, the packets that have not been assigned an output will be assigned to the remaining available

outputs randomly. This case occurs only when the number of deflection preference fields is less than $k - 1$. Once all the packets are assigned a unique output, the packets are routed accordingly.

The address length is given by $\sum_{i=1}^N K_i^2$ if full deflection preferences are given at each node. The address length will be rather long if the number of nodes or the number of outputs per node is large. Thus, the proposed deflection routing algorithm is only suitable for networks with small number of nodes and average number outputs per node such as in backbone networks. For example, in the AT&T IP backbone network [7], the length of the address using full deflection preferences is only 229 bits long. The length of the address can be reduced if the optical signal processing technology advance beyond the single-bit logic gates. Network simulations show that the improvement in network performance with the number of deflection choices decreases rapidly when the number deflection preferences increases. Therefore, a small number of deflection preferences is sufficient in practice. For large networks, the address length can be further reduced with the use of hierarchical addresses [5].

IV. ADDITIONAL FEATURES

Besides deflection preferences, additional features such as packet priority and time-to-live can be implemented in the proposed self-routing address scheme using bitwise optical logic gates.

A. Priority Class

There are extensive discussion of priority schemes in the literature [12]. Here, it suffices to say that we can add a priority field that contains one or more priority bits to the address header. Depending on the relative importance between the deflection preferences of a packet and its priority class, one can define two types of packet priority scheme: strong priority and weak priority. In the strong priority scheme, packets in higher priority classes are always assigned outputs before packets of lower priority classes irrespective of their deflection preferences. In the weak priority scheme, deflection preferences will be considered first. In the strong priority scheme, the priority bit or bits will be added to the front of the address header while in weak priority scheme the priority bits will be added to each address fields in the header.

Weak priority can be used to implement the *don't care* routing preference in networks with regular topologies such as Manhattan street networks (MSN) [13]. In such networks, packets at a node that have two paths with the same distance to the destination are assigned a lower priority. We found that a significant performance improvement can be obtained [13]. Moreover, one can implement a form of near-to-destination contention resolution scheme by assigning higher priority to the address fields corresponding to those nodes which are near the destination.

B. Time-to-Live

Livelock may occur in the proposed deflection routing scheme if contention resolution is carried out deterministically. It is, therefore, important for a network to remove packets that wander in the network for a long time. A time-to-live field can be implemented in the proposed self-routing address as follows.

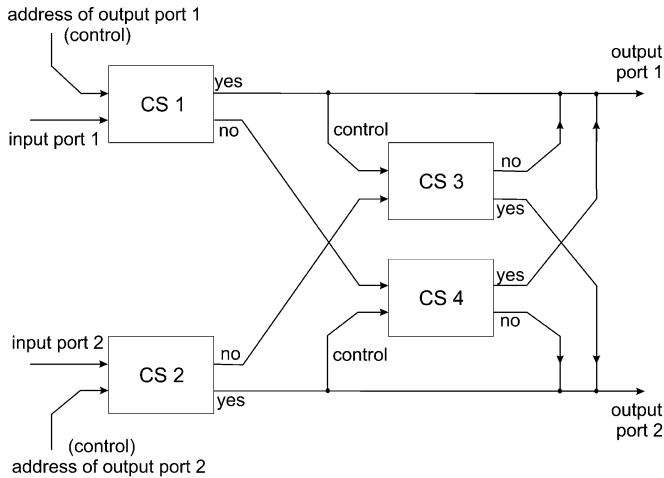


Fig. 2. Schematic of an all-optical 2×2 deflection routing node.

The nodes of the network are synchronized. Time is divided into cyclically indexed frames. Each frame contains an equal number of packet slot time. Each node keeps track of the current frame time. The time-to-live field contains m bits, where m is the number of time frames. Each bit identifies one time frame. When a node generates a packet header, the node determines the maximum time that the packet can stay in the network in unit of time frames. If the current time index is i and the packet's time-to-live is j time frames, then the i th to $(i + j)$ th bits in the time-to-live field is set to one. All other bits are set to zero. The time-to-live field is added to the beginning of the packet header.

When a node receives a packet, it checks the bit position in the time-to-live field corresponding to the current time index. If the bit position contains a one, the node continues to process the address field corresponds to the node. If the bit position contains a zero, the node will drop the packet.

V. ALL-OPTICAL IMPLEMENTATION

Even in such a simplified form, all-optical realization of the proposed deflection routing algorithm using only single-bit logic gates without buffers is still difficult. The main problem is to determine when contentions occur and how to resolve the contentions. The complexity of the problem increases with the number of input/output (I/O) ports in a node. However, we do not think that the problem is insurmountable with the advance in technology and for networks with small number of I/O ports per node. As an illustration, we will describe an original design of a 2×2 all-optical node implementing the proposed deflection routing algorithm. The design used four 2×2 all-optical crossbar switches invented by Glesk *et al.* [11]. No optical buffer is needed. Contention resolution is achieved by assigning a preferred output to an input. When two packets contend for the same output, the packet in the preferred input will act as a control signal to deflect the other packet in one of the crossbar switches. Thus, no higher level control nor optical buffer is necessary.

Fig. 2 shows a schematic of the proposed a 2×2 deflection routing node. The deflection routing node is made up of four 2×2 all-optical crossbar switches (CS) which are constructed

from two terahertz optical asymmetric demultiplexers (TOADs) [14], [15]. The first TOAD acts as an ultrafast all-optical routing controller and the second TOAD acts as an ultrafast all-optically controlled routing switch. The CSs and the outputs of the node are labeled as shown in Fig. 2. One of the inputs of the crossbar switch serves as the control and the other input serves as the signal. The packet is transmitted to the signal port. If the address field corresponding to the node in the packet header matches the address in the control signal, the packet will exit the switch through the port marked "yes," otherwise, the packet will exit through the port marked "no." Recall that the address field corresponding to a node in the address header of a packet in effect contains the local address of the output used by the packet at the node. The addresses of outputs 1 and 2 are used as control signals for crossbar switches 1 and 2, respectively [16].

There are three possible scenarios in the operations of the node. First, only one packet arrives. Assume that the packet arrives at input 1. If the address field of the packet corresponding to the node matches the address of output 1, the packet will be routed to the "yes" port and then onto output 1. The packet header is also used as the control signal for CS 3. Since no signal is input to CS 2, no packet exits CS 4. If the address field of the packet does not match the address of output 1, the packet will be sent to the "no" port of CS 1 and enters the input signal port of CS 4. Since no signal comes from the control port of CS 4, the packet will exit through the "no" output of CS 4 and then to output 2.

The second scenario is when two packets arrive and they are intended for different outputs. There is no contention and each packet will be routed to their intended output as described above. Finally, contention occurs when two packets arrive simultaneously and both are intended for the same output. For example, assume that packets A and B intended for output 1 arrive at inputs 1 and 2, respectively. Packet A exits the "yes" output of CS 1 and continues to output 1. Part of the header of packet A also split off and used as control signal to CS 3. Packet B exits the "no" output of CS 2 and enters the signal port of CS 3. Since the address field of packet B matches the address field of packet A which is used as the control signal in CS 3, packet B exits CS 3 at the "yes" port and exits the node through output 2. In other words, packet B is "deflected" by packet A in CS 3.

The above operations assume that all 2×2 crossbar switches are synchronized. It also requires that packets at all inputs arrive at each node simultaneously. The propagation delays between nodes should therefore be multiples of slot time. This can be achieved by lengthening some links, thus increasing the propagation delay. The tuneable fiber delay lines at each input provide further compensation for the propagation delay variances so that the transit packets are aligned and synchronized to the operations in node [17]–[19]. These packet synchronization problems as well as the packet contention problem are the fundamental problems for all slotted optical packet-switched networks [20]. In this paper, we assume that all synchronization problems have been solved with one of the existing approaches [17]–[20].

In the implementation shown in Fig. 2, packets arriving at input i have priority in the use of output i , $i = 1, 2$. It is well-

known that deterministic contention resolution algorithms will lead to *livelock* in deflection routing networks. The problem can be eliminated by dropping packets that exceed the time-to-live limit. Fig. 2 illustrates that it is possible to implement the proposed deflection routing algorithm all-optically. Although the hardware complexity of the design rapidly increases with the node degree K , nodes with small K s, like 3 and 4, can be still constructed using the 2×2 crossbar switches.

VI. PERFORMANCE ANALYSIS

Analytical models in general reduce the required computations in network performance evaluation. However, the computations will become intractable if we directly apply existing performance models to analyze the proposed deflection routing networks because the networks topology are arbitrary, the traffic distribution is nonuniform and the nodes can have degree of 3 or more. We therefore have to modify the existing models to improve the computational efficiency. In Section VI-A, we review typical models of deflection routing network and their computational requirement. We describe our proposed model in Section VI-B. The system throughput and delay are given in Section VI-C. We then discuss the computational requirement of the proposed model in Section VI-D. The performance model presented here assumes that there is no deflection preference field, i.e., packets that lose in their respective contentions will be assigned to the remaining outputs at random. Although implementation of the no deflection preference case with single-bit logic gates is still a challenge, this case can serve as a benchmark for the simulation results of other choices in the number of deflection preferences. The network performance for the cases of one and full deflection preferences will be studied by computer simulation.

A. Background

In networks with regular topology and uniform traffic such as MSN [21] and ShuffleNet [22], we can model the deflection routing performance using only a few typical nodes because of the similarity of traffic distribution in the nodes [23]–[26]. Computational complexity, therefore, does not rapidly increase with network size [23], [24], the degree of the nodes [25], or the number of priorities of the packets [26]. In networks with arbitrary topology, traffic distributions can be different from node to node. We cannot evaluate the network performance before we determine the global traffic distribution. Thus, the appropriate performance modeling approach for networks with arbitrary topology is similar to that of networks with nonuniform traffic [27]–[30]. The model in [27] is for MSNs with packet age contention policy. It is extended for arbitrary topology networks in [28], which requires to solve at least $\mathcal{O}(N^2\mathcal{K}D)$ equations each with at least $\mathcal{O}((ND)^{\mathcal{K}-1})$ terms, where N , \mathcal{K} , and D are number of nodes, average node degree, and diameter of the network, respectively. The model in [29] is for arbitrary topology networks with degree 2 nodes only, and is extended for buffered networks in [30]. If we follow the same approach in [29] to derive a model for arbitrary networks, we need to solve $\mathcal{O}(N^2\mathcal{K}^2)$ equations each with $\mathcal{O}(N^{\mathcal{K}-1})$ terms. It has large computational savings compared to that of [28] but will still become intractable

with large N and \mathcal{K} . We need a performance model that requires much smaller number of computations.

B. Performance Model

We assume a slotted network. The numbers of inputs and outputs of a node are equal. Packets arriving at inputs of nodes are independent of each other and of the new packets generated at the nodes. The offered load, $t_a(v)$, to a node a for destination v is the probability of generating a new packet with destination v at node a in a slot time. Since at most one new packet is generated at a node per slot time, $\sum_{v \neq a} t_a(v) \leq 1$. Different nodes can have different offered loads, i.e., nonuniform traffic loading.

Transit packets have priority over new packets in output contentions. A K -input node can receive as many as K packets per time slot. We define $u_{a,h}(v)$ as the probability of finding a packet destined for v at the h th output of node a . Similarly, $\ell_{b,h}(v)$ is the probability of finding a packet destined for v at the h th input of node b . We have $\ell_{b,k}(v) = u_{a,h}(v)$ if the h th output of node a is connected to the k th input of node b . The probability $u_{a,h}(v)$ in general is written as a function of $\ell_{b,k}(v)$'s and system loading. We can solve $u_{a,h}(v)$ by iterations but the computations will become intractable in networks with large N and \mathcal{K} [27]–[29].

We observe that packets with the same preferred output will have the same deflection probability irrespective of their destinations. Thus instead of calculating $u_{a,h}(v)$, we compute $U_a(k, h)$ the probability of finding a packet with preferred output k at output h of node a . The probability $U_a(k, h)$ can also be written as a function of $\ell_{b,k}(v)$'s and system loading. We shall show in Section VI-D that the computational requirement to calculate $U_a(k, h)$ is much smaller than that of $u_{a,h}(v)$. Hence, we solve $U_a(k, h)$ and compute $u_{a,h}(v)$ from them. We define the following for node a with degree K_a , i.e., with K_a inputs and K_a outputs:

- β_a is the probability of no new packets generated at node a ;
- $\alpha_{a,i} = 1 - \sum_{b \neq a} \ell_{a,i}(b)$ is the probability of no transit packets arriving at input i ;
- $\mathbf{C}_a(k)$ is the set of destinations of packets that request output k at node a ;
- $A_a(k) = \sum_{b \in \mathbf{C}_a(k)} t_a(b)$ is the total offered load for packets that request output k ;
- $L_{a,i}(k) = \sum_{b \in \mathbf{C}_a(k)} \ell_{a,i}(b)$ is the probability that a packet arriving at input i requests output k ,

We also define $\tilde{L}_{a,i}(k) = 1 - \alpha_{a,i} - L_{a,i}(k)$ as the probability that a transit packet arriving at input i does not prefer output k . The probability of finding a packet with preferred output k at output h depends on the following two independent events: 1) whether the packet is a new or a transit packet and 2) whether the preferred output k of the packet is the same as output h which is under consideration. We can solve $U_a(k, h)$ and then $u_{a,h}(v)$ once we determine the probabilities of the four combinations of these two events. For clarity, we will drop the subscript a in the following discussion.

1) *The Packet is a New Packet and $k = h$* : $A(h)$ is the probability that there is a new packet with h as the preferred output. The packet can reach output h only if there is no transit packets

requiring output h . The probability of finding a new packet at its preferred output h , $\chi(k = h, h)$, is given by

$$\chi(k = h, h) = A(h) \sum_{i=0}^{K-1} \left[\sum_{\substack{\{x_1, \dots, x_i\} \\ \in \mathbf{R}_i(K)}} \left(\prod_{j=1}^{K-i} \alpha_{y_j} \right) \times \left(\prod_{j=1}^i \tilde{L}_{x_j}(h) \right) \right] \quad (1)$$

where $\mathbf{R}_i(K)$ is the set of i -element number sets in which the numbers are the nonrepeating selections from the set $\{1, \dots, K\}$. In (1), x_1 to x_i are inputs that have transit packets. y_1 to y_{K-i} are idle inputs. Hence, $\{y_1, \dots, y_{K-i}\} = \{1, \dots, K\} - \{x_1, \dots, x_i\}$.

2) *The Packet is a New Packet and $k \neq h$* : In this case, we can find the new packet at output h only if the packet has been deflected by one or more transit packets. If there are i transit packets with $p \leq i$ nonidentical preferred outputs which include output k but not h , the packet has a probability $1/(K-p)$ of being deflected to output h at random. The probability of finding a new packet with preferred output $k \neq h$ at output h , $\chi(k \neq h, h)$, is given by

$$\chi(k \neq h, h) = A(k) \sum_{i=1}^{K-1} \left[\sum_{\substack{\{x_1, \dots, x_i\} \\ \in \mathbf{R}_i(K)}} \left(\prod_{j=1}^{K-i} \alpha_{y_j} \right) \times \sum_{p=1}^i \frac{1}{K-p} \sum_{\substack{(z_1, \dots, z_i) \\ \in \mathbf{S}_i(K, p, h, k)}} \left(\prod_{j=1}^i L_{x_j}(z_j) \right) \right] \quad (2)$$

where $\mathbf{S}_i(K, p, h, k)$ is the set of number sequences of length i and each is built from p nonrepeating selections from the set $\{1, \dots, K\} - \{h\}$. Each number sequence in $\mathbf{S}_i(K, p, h, k)$ contains the number k but not h . $\mathbf{R}_i(K)$, x_j and y_j are defined as in (1). z_1 to z_i are the i transit packet preferred outputs which have only p nonidentical elements.

3) *The Packet is a Transit Packet and $k = h$* : $L_r(h)$ is the probability of a transit packet at input r with the preferred output h . The packet has a probability $1/(q+1)$ of reaching output h if there are $i \leq K-1$ other transit packets and $q \leq i$ of them request output h . For a transit packet coming from input r with preferred output h , the probability of finding it at output h , $\sigma_r(k = h, h)$, is given as

$$\sigma_r(k = h, h) = L_r(h) \sum_{i=0}^{K-1} \left[\sum_{\substack{\{x_1, \dots, x_i\} \\ \in \mathbf{W}_i(K, r)}} \left(\prod_{j=1}^{K-i-1} \alpha_{y_j} \right) \times \sum_{q=0}^i \frac{1}{q+1} \sum_{\substack{(z_1, \dots, z_i) \\ \in \mathbf{T}_i(K, q, h)}} \left(\prod_{j=1}^i L_{x_j}(z_j) \right) \right] \quad (3)$$

where $\mathbf{W}_i(K, r)$ is the set of i -element number sets in which the numbers are the nonrepeating selections from the set $\{1, \dots, K\} - \{r\}$. $\mathbf{T}_i(K, q, h)$ is the set of number sequences of length i in which the numbers are selected from $\{1, \dots, K\}$. Each number sequence in $\mathbf{T}_i(K, q, h)$ contains q number of

h . In (3), x_1 to x_i are the i inputs, apart from the r th input, that have transit packets. y_1 to y_{K-i-1} are idle inputs, and $\{y_1, \dots, y_{K-i-1}\}$ is equal to $\{1, \dots, K\} - \{x_1, \dots, x_i\} - \{r\}$. The z_1 to z_i are the preferred outputs of the transit packets from inputs of x_1 to x_i . There are q number of h in z_1 to z_i .

4) *The Packet is a Transit Packet and $k \neq h$* : The packet is deflected to output h only if it loses the output contention with other transit packets. The packet has probability $q/(q+1)$ of losing the contention if there are $i \leq K-1$ other transit packets and q of them require output k . Let the transit packets require p nonidentical preferred outputs that include output k but not h . We have $q \leq p \leq i$. The packet has a probability $1/(K-p)$ of being deflected to output h if there is no new packets, or there is a new packet with preferred output being one of the p outputs. The packet will have probability $1/(K-p-1)$ of being deflected to output h if there is a new packet with preferred output that is not equal to any of the p outputs and h . For a transit packet coming from input r with preferred output k , the probability of finding it at output h , $\sigma_r(k \neq h, h)$, is given as

$$\sigma_r(k \neq h, h) = L_r(k) \sum_{i=1}^{K-1} \left[\sum_{\substack{\{x_1, \dots, x_i\} \\ \in \mathbf{W}_i(K, r)}} \left(\prod_{j=1}^{K-i-1} \alpha_{y_j} \right) \times \sum_{p=1}^i \sum_{q=1}^p \frac{q}{q+1} \sum_{\substack{(z_1, \dots, z_i) \\ \in \mathbf{U}_i(K, p, q, h, k)}} F(i, p) \times \left(\prod_{j=1}^i L_{x_j}(z_j) \right) \right] \quad (4)$$

and

$$F(i, p) = \begin{cases} \frac{\beta}{K-p} + \sum_{m \neq h} \frac{A(m)}{K-p-\varphi(m)} & i < K-1 \\ \frac{1}{K-p} & i = K-1 \end{cases}$$

where $\varphi(m) = 1$ if m is not equal to any of the z_j . Otherwise, $\varphi(m) = 0$. $\mathbf{U}_i(K, p, q, h, k)$ is the set of number sequences of length i in which the numbers are selected from $\{1, \dots, K\} - \{h\}$. Each number sequence contains q number of k and p nonidentical numbers but not h . In (4), $\mathbf{W}_i(K, r)$, x_j and y_j are defined as in (3). Similarly, z_1 to z_i are the preferred outputs of the transit packets from inputs x_1 to x_i . However, $\{z_1, \dots, z_i\}$ has only p nonidentical elements, and q out of the z_1 to z_i are h . $F(i, p)$ represents the probability of the packet being deflected to output h under different situations. If there are K transit packets, $i+1 = K$, the node cannot generate a new packet and $F(i, p)$ is equal to $1/(K-p)$. Otherwise, $F(i, p)$ depends on whether there is a new packet, and the preferred output of the newly generated packet.

After solving (1) to (4), we have $U_a(k, h) = \chi_a(k, h) + \sum_{i=1}^{K_a} \sigma_{a,i}(k, h)$. The expression for $u_{a,h}(v)$ is

$$u_{a,h}(v) = \frac{t_a(v)}{A_a(k)} \chi_a(k, h) + \sum_{i=1}^{K_a} \frac{\ell_{a,i}(v)}{L_{a,i}(k)} \sigma_{a,i}(k, h) \quad (5)$$

for a packet with destination v at node a , where the preferred output k is determined by the paths encoded in the packet

address header. Note that we have reinserted the subscript a to designate node a in the above quantities. Equation (5) is based on the observation that packets with the same preferred output will have the same deflection probability irrespective of their destinations. Hence, we can solve $u_{a,h}(v)$ with (1) to (5) by iterations.

C. Throughput and Delay

The number of packets with destination v that can be accepted by node $a \neq v$ in a slot time is given by

$$\text{th}_a(v) = \frac{t_a(v)}{A_a(k)} \sum_{h=1}^{K_a} \chi_a(k, h) \quad (6)$$

where K_a is the degree of node a , and k is the preferred output of packets with destination v at node a . The throughput of node v , i.e., the average number of packets that node v receives in a time slot is given as $\text{TH}(v) = \sum_{a \neq v} \text{th}_a(v)$, or

$$\text{TH}(v) = \sum_{i=1}^{K_v} \ell_{v,i}(v) \quad (7)$$

where K_v is the degree of node v . Recall that $\ell_{b,k}(v) = u_{a,h}(v)$ if the h th output of node a is connected to the k th input of node b . For a uniformly loaded N -node network, average throughput per node TH may be used for performance comparison, where TH is given by

$$\text{TH} = \frac{1}{N} \sum_{v=1}^N \text{TH}(v). \quad (8)$$

After we compute the throughput, we can solve the average delay for packets from other nodes to node v by Little's rule, $\mathcal{D} = \mathcal{N}/\lambda$ [31], where \mathcal{D} is the average network delay, \mathcal{N} is the average number of packets in the network, and λ is the average packet arrival rate to the network. The equivalent rate of packets with destination v that can enter the network is equal to $\text{TH}(v)$ for a steady system. The number of packets with destination v in the network depends on $u_{a,h}(v)$ and the length of the links between nodes. Hence, the average packet delay from other nodes to node v , $D(v)$, is given as

$$D(v) = \frac{1}{\text{TH}(v)} \sum_{a \neq v} \sum_{i=1}^{K_a} H(a, i) u_{a,h}(v) \quad (9)$$

where $H(a, i)$ is the length in time slots of the i th output link of node a . The average packet delay of a network is given by

$$D = \frac{1}{N \times \text{TH}} \sum_{a=1}^N \sum_{h=1}^{K_a} H(a, h) \sum_{k=1}^{K_a} U_a(k, h). \quad (10)$$

The performance model in Section VI-B does not assume any propagation delay between nodes. In steady state, the probability distributions of packets at both ends of a link should be equal regardless of the length of the link. Equations (9) and (10) compute the numbers of packet as well as the packet delays in the network at steady state. Numerical simulation results show that it takes longer for a system to reach steady state if the lengths of links between nodes are large.

D. Computational Requirement

We solve $u_{a,h}(v)$'s using (1)–(5) by iterations. In an iteration, we apply (1)–(4) on each output of each node to solve for $U_a(k, h)$'s from $\sigma_{a,i}(k, h)$'s and $\chi_a(k, h)$'s. We need to solve $\mathcal{O}(N\mathcal{K}^2)$ equations and each computational requirement is proportional to $\mathcal{O}(\mathcal{K}^{\mathcal{K}})$, where N and \mathcal{K} are the number of nodes and average node degree, respectively. After solving the $U_a(k, h)$'s, we apply (5) to each output of each node to compute $u_{a,h}(v)$'s. We need to solve another $\mathcal{O}(N^2\mathcal{K})$ equations each with computational requirement of $\mathcal{O}(\mathcal{K})$. This process repeats until all $u_{a,h}(v)$'s converge to their final values. The required computations grow exponentially with the average node degree \mathcal{K} but not the number of node N . The computational savings will be significant in large networks with arbitrary topology if compared to existing methods [27]–[29].

VII. NETWORK SIMULATIONS

In this section, we study the performance of the proposed deflection routing algorithm. We consider a network based on the topology of the AT&T North America OC-48 fiber network shown in Fig. 3, [7]. Three different implementations of the deflection routing algorithm in which zero, one, and the maximum number of deflection preference fields are used. Analytic results for the no deflection preference case are obtained using (1)–(10). Evaluation of the (1)–(5) starts from an empty network. Convergence to the final results in general occurs after 10 to 80 iterations and takes several minutes on a 600-MHz Alpha machine. We have verified the results using the method presented in [28]. It provides the same results, but the computational time required is in the order of hours. The network performance for the cases of one deflection preference and full deflection preferences are studied by computer simulations. All simulation results are based on the observations of the arrivals of not less than 2×10^6 new packets at each node. The 95% confidence interval is about 1% of the values shown.

We assume that a node sends the packets it generates uniformly to every node in the network except itself. All links in the networks are bidirectional. In Fig. 3, there are 27 nodes and 37 links. The number of nodes with 1, 2, 3, 4, and 5 I/O ports are 1, 15, 3, 5, and 3, respectively. The average number of outputs per node is 2.8. The basic uncompressed self-routing address of a packet header is 74 bits long [5]. The shortest paths between the nodes are used in the construction of the self-routing addresses. The shortest propagation delay between two nodes is estimated to be 0.75 ms. All other delays are rounded up to integral multiples of 0.75 ms. The maximum propagation delay is 16.5 ms. The delay between the nodes normalized by the minimum propagation delay are given in parentheses in Fig. 3. The minimum propagation delay between nodes is chosen as the slot size, e.g., a slot is 234 kB long at OC-48 (2.5 Gb/s). Hence, the result from (10) is further multiplied with 0.75 ms to obtain the packet propagation delay between nodes.

Fig. 4 shows the analytic results as well as simulation results for the delay throughput curves. The delay and the throughput values shown are averaged over all nodes. The offered loads are

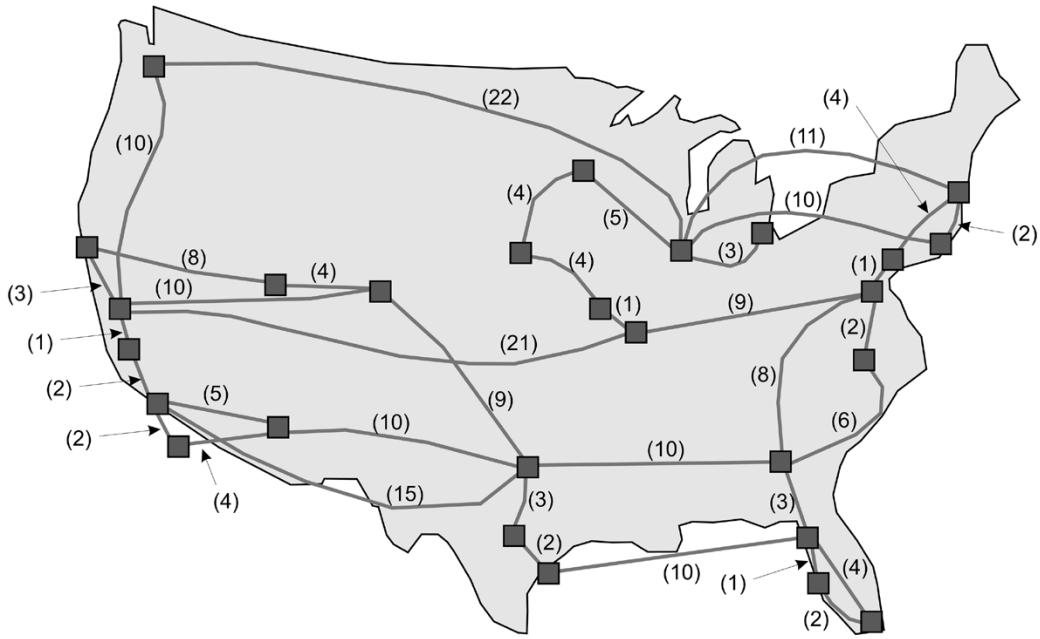


Fig. 3. AT&T North America OC-48 optical fiber network obtained from [7].

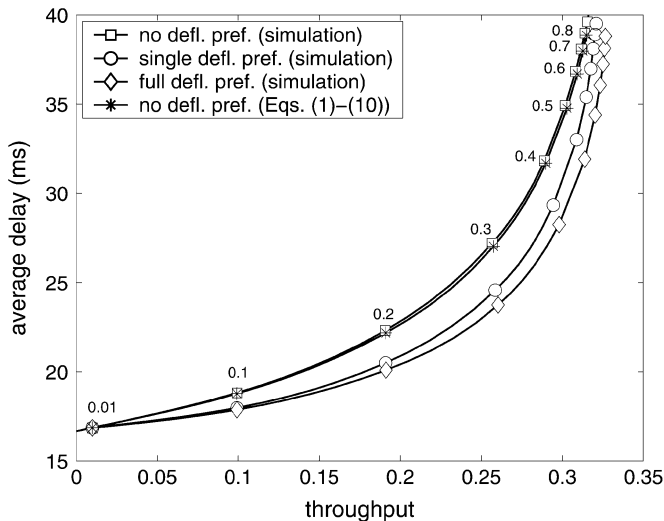


Fig. 4. Delay throughput curves. The offered loads are marked beside the curves.

marked beside the curves. Squares, circles, and diamonds represent simulation results for the cases of no deflection preference, a single deflection preference, and full deflection preferences, respectively. Asterisks represent the analytic results for the no deflection preference case. From Fig. 4, the analytic results agree well with the numerical simulation results indicating that the assumption of independence between transit packets arrivals and new packet generation is a good approximation. In Fig. 4, the system performance of the original address scheme, i.e., without deflection preference, improves if a single deflection preference field is added. There is much less performance improvement, however, when the number of deflection preference fields is increased beyond one, indicating that most of the deflected packets are routed to the outputs of their first deflection preference. Therefore, the addition of a single deflection preference field should be sufficient in most cases.

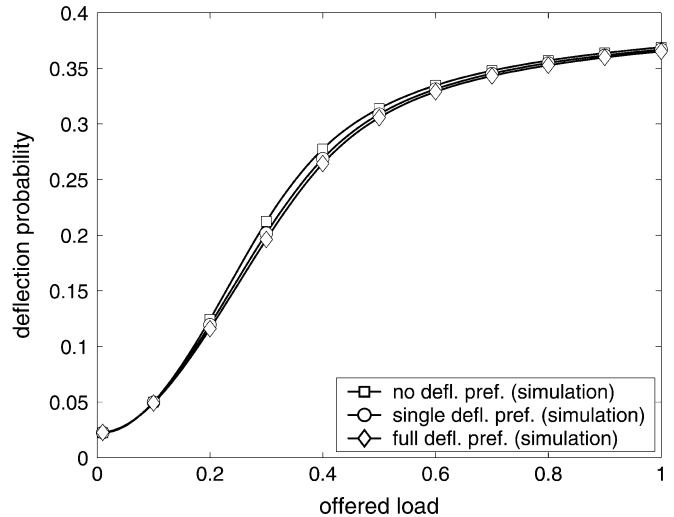


Fig. 5. Average packet deflection probability at a node versus offered load.

Figs. 5–7 illustrate that deflection preferences improve system performance by reducing the average additional delay due to a deflection while the reduction of packet deflections is not significant. Fig. 5 plots the average packet deflection probability that a packet encountered on each node with different numbers of deflection preferences. Although the number of deflections a packet has encountered in the journey to its destination is proportional to the average packet deflection probability, it also depends on the path that will become random once a deflection occurs. Fig. 6 plots the average number of deflections that a packet encountered between source and destination. We observed that neither the deflection probability nor the average number of deflections depends strongly on the number of deflection preferences. In networks with regular topology, each deflection causes a fix delay. For example, each deflection results in a delay of four hops in an MSN. In a

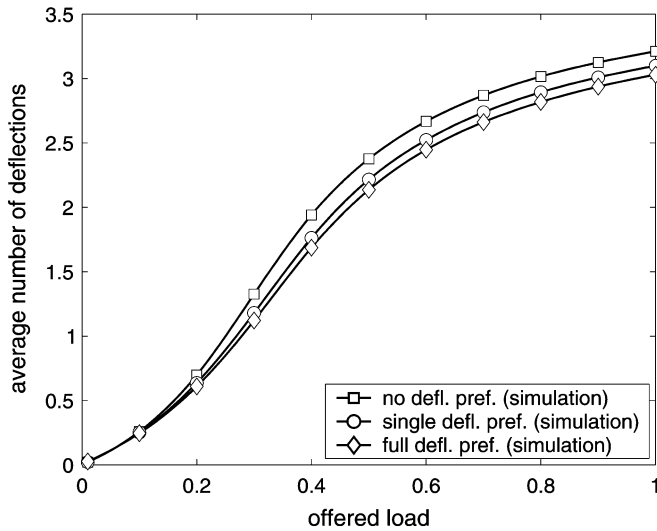


Fig. 6. Average number of deflections experienced by a packet versus offered load.

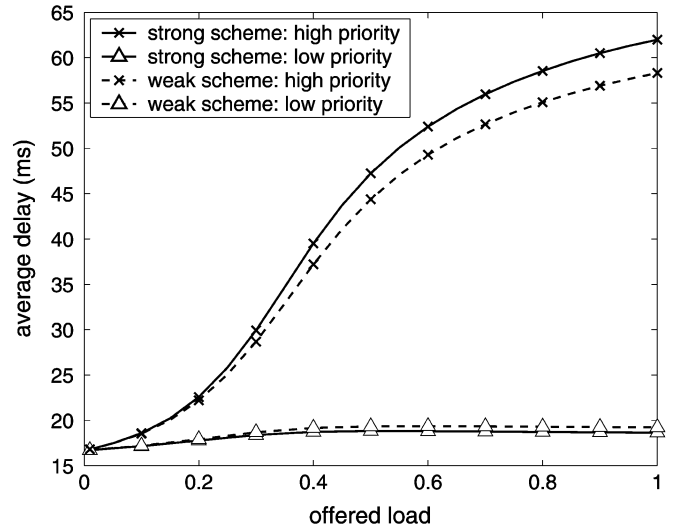


Fig. 8. Average delay versus the total offered load of the strong and the weak priority schemes. The ratio of high/low priority traffic offered loads is 1 : 1.

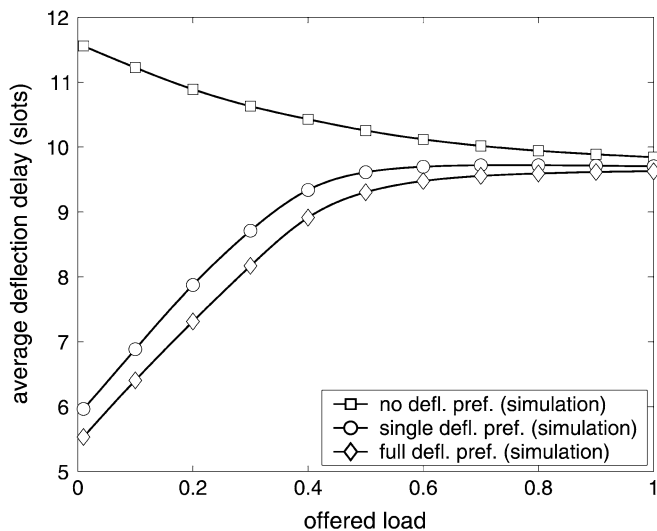


Fig. 7. Average additional delay due to a deflection versus offered load.

network with arbitrary topology, the additional delay depends on the location of the deflection and the system loading at the moment. We plot the average additional delay due to a deflection in Fig. 7. At small offered load, the additional delay due to a deflection in the no deflection preference case nearly doubles that with deflection preferences. When the offered load increases, the additional deflection delay for the no deflection preference case decreases while that for the single and full deflection preference cases increase. When the offered load reached unity, all three cases gave about the same deflection delay. The apparent difference in the behaviors of the average additional deflection delay curves is due to local deflection hot spots which depend on the topology of the network, the number of deflection preference fields defined, and the offered load. Note that both the number of outputs and the delay caused by a deflection vary from node to node. The locations of deflection hot spots also change when the offered load increases.

Next, we compare the performance of the strong and weak priority schemes discussed in Section IV-A. Fig. 8 shows the average delay versus the total offered load for both the strong priority scheme and the weak priority scheme. Full deflection preferences are used. There are only two priority classes and the ratio of offered load for the packets in the two priority classes is one to one. Solid lines represent the average delay of the strong priority scheme while the dashed lines represent the average delay of the weak priority scheme. Crosses correspond to the packets of the higher priority class while triangles correspond to the packets of the lower priority class. From Fig. 8, the higher priority packets of the strong priority scheme have a shorter average delay than that of the weak priority scheme, but the lower priority packets of the strong priority scheme have a longer average delay than that of the weak priority scheme. The difference, however, is not very significant. The priority scheme to be used in practice will therefore largely depend on the relative ease in implementation.

VIII. CONCLUSION

We have proposed a deflection routing algorithm that can be implemented with the self-routing address scheme proposed in [5] for small to medium size networks. Alternate choices of outputs are encoded in the address header as additional deflection preference fields. If more than one packet contend for the same output, one of the contending packets is selected to use its preferred outputs, and other packets will attempt to use the outputs specified in their deflection preference fields. We have discussed all-optical implementations of the proposed deflection routing scheme. We show the design of a 2×2 all-optical deflection routing node which is based on only single-bit logic gates and does not use any buffers. For the no deflection preference case, we derive the equations that determine the deflection probabilities based on the preferred output at the node instead of the destinations. The alternate approach provides significant computational savings if compared to the conventional destination-based

methods. Finally, we study the performance of the proposed deflection routing algorithm using a network topology based on the AT&T North America OC-48 optical fiber network. We find that the use of a single deflection preference field shows significant improvement in the network performance, as compared with no deflection preference. The use of a single deflection preference is sufficient in the example studied.

REFERENCES

[1] K. L. Hall and K. A. Rauschenbach, "100-Gbit/s bitwise logic," *Opt. Lett.*, vol. 23, pp. 1271–1273, 1998.

[2] K. E. Stubkjaer, "Semiconductor optical amplifier-based all-optical gates for high-speed optical processing," *IEEE J. Select. Topics Quantum Electron.*, vol. 6, pp. 1428–1435, Nov.–Dec. 2000.

[3] C. Bintjas, M. Kalyvas, G. Theophilopoulos, T. Stathopoulos, H. Avramopoulos, L. Occhi, L. Schares, G. Guekos, S. Hansmann, and R. Dall'Ara, "20 Gb/s all-optical XOR with UNI gate," *IEEE Photon. Technol. Lett.*, vol. 12, pp. 834–836, July 2000.

[4] V. W. S. Chan, K. L. Hall, E. Modiano, and K. A. Rauschenbach, "Architectures and technologies for high-speed optical data networks," *J. Lightwave Technol.*, vol. 16, pp. 2146–2168, Dec. 1998.

[5] X. C. Yuan, V. O. K. Li, C. Y. Li, and P. K. A. Wai, "A novel self-routing scheme for all-optical packet switched networks with arbitrary topology," *IEEE J. Lightwave Technol.*, vol. 21, pp. 329–339, Feb. 2003.

[6] P. K. A. Wai, L. Y. Chan, L. F. K. Lui, H.-Y. Tam, and M. S. Demokan, "1 × N all-optical packet switch at 10 Gb/s," presented at the Conf. Lasers and Electro-Optics CLEO'2004, San Francisco, CA, May 16–21, 2004, Paper CTuFF2.

[7] (2001, Mar.). [Online]. Available: <http://www.ipservices.att.com/backbone/bbone-map.cfm>

[8] (2002, Apr.). [Online]. Available: <http://www.dante.net/geant/Schematic.jpg>

[9] A. S. Acampora and S. I. A. Shah, "Multihop lightwave networks: A comparison of store-and-forward and hot-potato routing," *IEEE Trans. Commun.*, vol. 40, pp. 1082–1090, June 1992.

[10] L. Y. Chan, K. K. Qureshi, P. K. A. Wai, B. Moses, L. F. K. Lui, H. Y. Tam, and M. S. Demokan, "All-optical bit-error monitoring system using cascaded inverted wavelength converter and optical NOR gate," *IEEE Photon. Technol. Lett.*, vol. 15, pp. 593–595, Apr. 2003.

[11] I. Glesk, K. I. Kang, and P. R. Prucnal, "Demonstration of ultrafast all-optical packet routing," *Electron. Lett.*, vol. 33, pp. 794–795, Apr. 1997.

[12] A. K. Choudhury and V. O. K. Li, "An approximate analysis of the performance of deflection routing in regular networks," *IEEE J. Select. Areas Commun.*, vol. 11, pp. 1302–1316, Oct. 1993.

[13] C. Y. Li, P. K. A. Wai, X. C. Yuan, and V. O. K. Li, "Multicasting in deflection-routed all-optical packet-switched networks," in *Proc. IEEE Globecom*, Taipei, Taiwan, R.O.C., Nov. 2002.

[14] J. P. Sokoloff, P. R. Prucnal, I. Glesk, and M. Kane, "A Terahertz optical asymmetric demultiplexer," *IEEE Photon. Technol. Lett.*, vol. 5, pp. 787–790, July 1993.

[15] M. Eiselt, W. Pieper, and H. G. Weber, "SLALOM: Semiconductor laser amplifier-based processing nodes," *J. Lightwave Technol.*, vol. 13, pp. 2099–2112, Oct. 1995.

[16] I. Glesk, J. P. Sokoloff, and P. R. Prucnal, "All-optical address recognition and self-routing in a 250 Gbit/s packet switched network," *Electron. Lett.*, vol. 30, pp. 1322–1323, Aug. 1994.

[17] C. L. Lu, T. K. Fong, R. T. Hofmeister, P. Poggiolini, and L. G. Kazovsky, "CORD-A WDM optical network: Design and experiment of fast data synchronization by pilot-tone transport," *IEEE Photon. Technol. Lett.*, vol. 8, pp. 1070–1072, Aug. 1996.

[18] R. T. Hofmeister, C.-L. Lu, M.-C. Ho, P. Poggiolini, and L. G. Kazovsky, "Distributed slot synchronization (DSS): A network-wide slot synchronization technique for packet-switched optical networks," *J. Lightwave Technol.*, vol. 16, pp. 2109–2116, Dec. 1998.

[19] M. C. Cardakli and A. E. Willner, "Synchronization of a network element for optical packet switching using optical correlators and wavelength shifting," *IEEE Photon. Technol. Lett.*, vol. 14, pp. 1375–1377, Sept. 2002.

[20] I. Chlamtac, A. Fumagalli, L. G. Kazovsky, P. Melman, W. H. Nelson, P. Poggiolini, M. Cerisola, A. N. M. M. Choudhury, T. K. Fong, R. T. Hofmeister, C. L. Lu, A. Mekikittikul, D. J. M. Sabido IX, C. J. Suh, and E. W. M. Wong, "CORD: Contention resolution by delay lines," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 1014–1029, June 1996.

[21] N. F. Maxemchuk, "Routing in Manhattan street network," *IEEE Trans. Commun.*, vol. COM-35, no. 5, pp. 503–512, 1987.

[22] A. S. Acampora, M. J. Karol, and M. G. Hluchyj, "Terabit lightwave networks: The multihop approach," *AT&T Tech. J.*, vol. 6, pp. 21–34, 1987.

[23] A. G. Greenberg and J. Goodman, "Sharp approximate models of deflection routing in mesh networks," *IEEE Trans. Commun.*, vol. 41, pp. 210–223, Jan. 1993.

[24] A. Krishna and B. Hajek, "Performance of shuffle-like switching networks with deflection," in *Proc. INFOCOM'90*, vol. 2, San Francisco, CA, June 1990, pp. 473–480.

[25] A. G. Greenberg and B. Hajek, "Deflection routing in hypercube networks," *IEEE Trans. Commun.*, vol. 40, pp. 1070–1081, June 1992.

[26] Z. Zhang and A. Acampora, "Performance analysis of multihop lightwave networks with hot potato routing and distance-age priorities," *IEEE Trans. Commun.*, vol. 42, pp. 2571–2581, Aug. 1994.

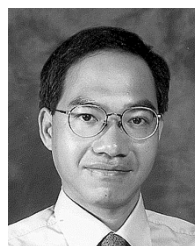
[27] J. Brassil and R. Cruz, "Non-uniform traffic in Manhattan street network," *Perform. Eval.*, vol. 25, pp. 233–242, 1996.

[28] E. Noel and K. W. Tang, "Performance modeling of multihop network subject to uniform and nonuniform geometric traffic," in *IEEE/ACM Trans. Networking*, vol. 8, Dec. 2000, pp. 763–774.

[29] J. Bannister, F. Borgonovo, L. Fratta, and M. Gerla, "A versatile model for predicting the performance of deflection-routing networks," *Perform. Eval.*, vol. 16, no. 1–3, pp. 201–222, 1992.

[30] —, "A performance model of deflection routing in multibuffer networks with nonuniform traffic," in *IEEE/ACM Trans. Networking*, vol. 3, Oct. 1995, pp. 509–520.

[31] L. Kleinrock, *Queueing System, Vol. 1: Theory*. New York: Wiley, 1976.



Chun-Yin Li (S'91–M'93–S'93–M'97) received the B.S. degree from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1986 and the Ph.D. degree from The Hong Kong Polytechnic University, Hong Kong, in 2000.

In 1986, he joined Taicom Ltd., Taiwan, where he worked as a Transmission Engineer on the M90, M135, and M405 Optical Fiber Telecommunication System projects. In 1988, he joined ROCTEC Ltd., Hong Kong, where he worked as a Design Engineer on the computer products development. In 1993, he joined the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University as a Research Assistant, and then as a Research Fellow. In 2000, he worked as a Research Associate in the same department. His research interests include network performance evaluation, all-optical network routing, and network theory. He published 17 papers in these areas.



P. K. A. Wai (SM'96) received the B.S. degree with first class honors from the University of Hong Kong, Hong Kong, in 1981, and the M.S. and Ph.D. degrees from the University of Maryland, College Park, in 1985 and 1988, respectively.

In 1988, he joined Science Applications International Corporation in McLean, VA, where he worked as a Research Scientist on the Tethered Satellite System project. In 1990, he became a Research Associate in the Department of Electrical Engineering, University of Maryland. In 1996 he joined the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, as an Assistant Professor. He became Associate Professor in 1997, and Professor and Head of the Department in 2002. His research interests include theory of solitons, modeling of fiber lasers, simulations of integrated optical devices, long haul optical fiber communications, all-optical packet switching, and network theories. He is an active contributor to the technical field, having over 150 international publications.

Prof. Wai is a member of the Optical Society of America.

Xiao Chun Yuan was born in China in 1974. He received the B.S. degree in computer science from Wuhan University, Wuhan, China, in 1993, the M.S. degree from the Institute of Computing of the Chinese Academy of Science, Beijing, China, in 1996, and the Ph.D. degree in electrical and electronic engineering from the University of Hong Kong, Hong Kong, in 2002.

He worked at the Chinese Academy of Science from 1996 to 1998, and joined the University of Hong Kong as a Research Assistant in 1998. His research area is in optical network routing, and he has published five papers in this area. He has also published three books on the Internet, JAVA programming, and website design.



Victor O. K. Li (S'80–M'81–SM'86–F'92) was born in Hong Kong in 1954. He received the S.B., S.M., E.E., and Sc.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 1977, 1979, 1980, and 1981, respectively.

He joined the University of Southern California (USC), Los Angeles, in February 1981, and became Professor of Electrical Engineering and Director of the USC Communication Sciences Institute. Since September 1997, he has been with the University of Hong Kong, Hong Kong, where he is Chair Professor of Information Engineering at the Department of Electrical and Electronic Engineering, and Managing Director of Versitech Ltd., the technology transfer and commercial arm of the University. He also serves on various corporate boards. His research is in information technology, including high-speed communication networks, wireless networks, and Internet technologies and applications. He is a Principal Investigator of the Area of Excellence in Information Technology funded by the Hong Kong government. Sought by government, industry, and academic organizations, he has lectured and consulted extensively around the world. He chaired the Computer Communications Technical Committee of the IEEE Communications Society (1987–1989), and the Los Angeles Chapter of the IEEE Information Theory Group (1983–1985). He co-founded the International Conference on Computer Communications and Networks (IC3N), and chaired its Steering Committee 1992–1997. He also chaired various international workshops and conferences, including, most recently, IEEE INFOCOM 2004. He was appointed to the Hong Kong Information Infrastructure Advisory Committee by the Chief Executive of the Hong Kong Special Administrative Region. He also serves on the Innovation and Technology Fund (Electronics) Vetting Committee, the Small Entrepreneur Research Assistance Programme Committee, the Engineering Panel of the Research Grants Council, and the Task Force for the Hong Kong Academic and Research Network (HARNET) Development Fund of the University Grants Committee. He was a Distinguished Lecturer at the University of California at San Diego, at the National Science Council of Taiwan, and at the California Polytechnic Institute. He has also delivered keynote speeches at many international conferences.

Prof. Li has received numerous awards, including, most recently, the Outstanding Researcher Award of the University of Hong Kong, the KC Wong Education Foundation Lectureship, the Croucher Foundation Senior Research Fellowship, and the Bronze Bauhinia Star, Government of the Hong Kong Special Administrative Region, China. He has served as an Editor of *IEEE Network*, *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC)* Wireless Communications Series, and Telecommunication Systems. He also guest edited special issues of *IEEE JSAC*, *Computer Networks and ISDN Systems*, and *KICS/IEEE Journal of Communications and Networking*. He is now serving as an editor of *ACM/Kluwer Wireless Networks* and *IEEE Communications Surveys and Tutorials*.