Multicasting in deflection-routed all-optical packet-switched networks

C. Y. Li, P. K. A. Wai,
Photonics Research Center and
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong
E-mails: {enli, enwai}@polyu.edu.hk

X. C. Yuan, and Victor O. K. Li
Department of Electrical and Electronic Engineering
The University of Hong Kong, Hong Kong
E-mails: {xcyuan, vli}@eee.hku.hk

***Abstract*—Two multicast protocols are proposed for deflection-routed all-optical packet-switched networks. One scheme sends a deflected multicast packet back to the root node while the other sends it back to the deflection point. Both schemes can be implemented using demonstrated optical signal processing technology. The performance of the two proposed multicast schemes are compared using Manhattan Street Networks. We found that the back-to-the-root-node scheme performed better than the back-to-the-deflection-node scheme. A hybrid approach can further improve the system performance.**

I. Introduction

Multicast is an important service that will be provided by most networks in the future. Traditional multicast routing protocols are based on store-and-forward routing strategy. These protocols assume that the nodes have the capabilities to buffer packets and have sufficient processing power to perform tasks such as multicast tree management and packet duplications at branch nodes. Both assumptions may not hold in all-optical networks. Fiber-based optical buffers have a fixed delay and the signals degrade with time inside the buffers. Besides, only simple logic gates have been realized all-optically at present. Complex optical logic circuits are not available yet [1]. Current proposals for multicasting services on optical networks such as the light tree approach [2] and the optical burst switching approach [3] assume that hybrid switching is used, *i.e.*, the packet headers are converted to the electrical domain for processing [3] while the packets remain in the optical domain.

Recently, we proposed a deflection routing address scheme for all-optical packet switched networks [4]. The proposed routing scheme requires only simple signal processing and can be implemented all-optically using the bitwise optical logical gates demonstrated in the literature. Deflection routing is used in order to eliminate or minimize the use of buffers. In deflection routing, packets that lose their contentions for their optimal output links are intentionally routed to the "wrong" output ports. The misrouted packets will eventually find their ways to their destination nodes with increased delay. In effect, deflection routing uses the network links as buffers for the packets instead of storing them at the nodes.

This research is 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 is provided by a grant from The Hong Kong Polytechnic University (Project Number A-PD69).

Incorporation of multicast service in the proposed deflection-routed all-optical packet-switched networks is not straightforward. For example, because of the lack of all-optical table look-up functions, information of the multicast trees cannot be stored locally at the nodes. Such information must be encoded in the address header of the packet similar to the self-routing address proposed in [5]. Moreover, a multicast packet can be forced off its path in deflection routing. The packet may skip some nodes and/or visit other nodes on the tree multiple times if proper measures are not taken. To avoid erroneous handling of a multicast packet, a node must be able to determine whether it has seen an arriving multicast packet before and whether the multicast packet has followed its path properly. This can be achieved if a node keeps a record of every packet's visit, or the node modifies the packet's address header. Both approaches are difficult to implement all-optically. The former approach requires all-optical memory retrieval while the latter requires all-optical modification of the address headers on-the-fly. To complicate matters further, the branch nodes of the tree have to determine whether there are sufficient output links before they duplicate the multicast packets. In deflection routing, one typically assumes that the number of input and output ports are equal. Therefore every arriving packet is guaranteed at least one choice of output link. The same assumption, however, does not hold in multicasting at the branch nodes of the tree.

In this paper, we propose two multicast protocols that are compatible with the deflection routing scheme proposed in [4]. Both protocols can be implemented using demonstrated optical technology. In Section II, we give a brief review of the basic unicast deflection routing scheme we proposed in [4]. In Section III, we describe how to encode the multicast tree in the address header. Sections IV discusses the two proposed multicast schemes. In Section V, we describe the all-optical implementation of the proposed multicast deflection routing schemes. Section VI compares the performance of the two schemes using the Manhattan Street Networks. Finally, we conclude in Section VII.

II. The basic unicast deflection routing scheme

A. Self-routing address

In the proposed self-routing address, 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

0-7803-7632-3/02/$17.00 ©2002 IEEE

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 links and nodes used by the two paths must be the same.

For a network of N nodes and L output links, we assume that all links are bi-directional, or each node has the same number of input/output links. Each node is labeled from 1 to N. The output links connecting to each node are labeled from 1 to $n(i)$ where $n(i)$ is the number of output links connected to the i-th node. We have $\sum_{i=1}^{N} n(i) = 2L$. The self-routing 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 $n(i)$ 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 $n(j)$ bits, the x-th bit say, is set to 1. The other bits at the j-th address field are set to zeroes. A non-zero 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 link [4], [5].

B. Deflection routing

To implement deflection routing, alternate link choices called deflection preference fields are encoded in the address header after the primary link choice in the same fashion as described in II-A [4]. The deflection preference fields have the same number of bits as the primary address field. Each bit position in the deflection preference fields is associated with an output port of the node. Each deflection preference field identifies an output that is different from all the previous choices of outputs by the packet. Figure 1 shows an example of the address field corresponding to a 3×3 node j in a network with full deflection preference.

C. Routing operation

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 zeroes. Otherwise it forwards the packet to the local output link as specified. A node assigns its output ports to the arriving packets in successive rounds in accordance to the order of the output port choices indicated in the packet's deflection preference fields. In the first round of output port assignment, the node considers the primary address fields of the arriving packets. The node assigns the uncontended output ports to the packets requesting them. The node assigns each of the contended output ports to one of the contending packets at random. In the second round of output ports assignment, the node processes the first deflection preference fields of all the packets that have not been assigned an output port in the first round. These are the packets that have lost in their respective contentions. If the output port indicated at the first deflection

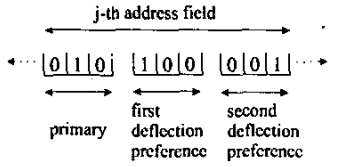

Fig. 1. An example of the address field of a 3×3 node with full deflection preference fields added.

preference field of a packet has already been assigned in the previous round, the packet will not be assigned an output port in this round. If more than one of the remaining packets request the same output port in their first deflection preference field, the contended output ports are assigned to one of the contending packets at random. The uncontended output ports are assigned to the packets requesting them. The procedure will be repeated until all the packets are assigned an output port or the deflection preference fields are exhausted. In the latter case, the packets that have not been assigned an output port will be assigned to the remaining available output ports randomly.

III. Encoding the Multicast Tree

We assume that the multicast trees are pre-computed. We encode the multicast tree in the packet header so that intermediate nodes can route and duplicate the multicast packets with minimum processing. Similar to the self-routing address described in Section II, a multicast address contains N fields and each address field corresponds to a node in the network. But the i-th field now contains $n(i) + 1$ bits, where $i = 1, \ldots, N$. The address fields corresponding to nodes not belonging to the multicast tree are set to **0**. For nodes belonging to the tree, the first bit in the corresponding address field is used as a copy bit which instructs the node to make a local copy of the packet if it is set to **1**. The copy bit is set to **0** for nodes that only forward and/or duplicate the packet. For each node on the tree except the leave nodes, the $(j + 1)$-th bit of the i-th field in the multicast address is set to **1** if the j-th output link of node i is an edge of the multicast tree. Note that the address fields corresponding to the branch nodes of the multicast tree contain multiple **1** bits apart from the copy bit. The leave nodes have their copy bit set to **1** and the rest of the address field set to **0**. The number of **1** bits in the multicast address is equal to the number of edges plus the number of receiving nodes in the multicast tree. Deflection preference fields can be added after the primary address fields in the manner described in Section II-B.

When a node on the multicast tree receive a packet, it will make a local copy of the packet if the copy bit is set to **1**. The node makes $x - 1$ copies of the packet where x is the number of **1**'s in the address field corresponding to the node. The node then sends the x packets (the $x - 1$ copies and the original packet) to the x different output links identified by the **1** bits. The multicast address described above would function properly

if there were only one packet in the network, *i.e.*, there is no output link contention.

IV. Multicast in Deflection Routing

In deflection routing, a multicast packet may be deflected off the multicast tree due to output link contentions. The packet therefore has to be able to find its way back to the deflection point after each deflection and resumes its journey. The nodes on the tree that the deflected packet encounters, if any, during its trip back to the deflection point should ignore the multicast instruction encoded in the packet. There are different ways to carry this out. In the following, we proposed two schemes that can be implemented using available optical technology. In the first scheme, a deflected multicast packet is directed back to the root node while in the second scheme the deflected multicast packet is directed back to the deflection node instead. This is achieved by including in the address header the address of the root node or the deflection node in the first and second multicast scheme, respectively, besides the multicast address as described in Section III.

A. The back-to-the-root-node (BRN) scheme

The packet address contains three parts. Part I contains two bits, part II contains the unicast address of the root node, and Part III contains the multicast address we described in III. The first bit in Part I identifies the type of the packet; unicast (**0**) or multicast (**1**). The second bit in Part I of the address identifies whether the packet is a normal multicast packet (**1**) or a deflected multicast packet (**0**). When a node i receives a packet, there are three possibilities.

1) Part I of the address = **11**: The packet is a normal multicast packet. The node checks the address field corresponding to the node in the multicast address (part III of the address). If the first bit in the corresponding field, *i.e.*, the copy bit, is **1**, the node makes a local copy of the packet. The node then erases the copy bit. Let the number of **1** bits in the rest of the corresponding field be x. If $x = 0$, the node is a leaf node of the multicast tree. The node does not forward the packet. If $x = 1$, the node forwards the packet to the output link identified by the **1** bit. If $x > 1$, the node then checks whether all the output links identified by the **1** bits are available. If so, the node makes $x - 1$ copies of the packet. In each of the x copies of the packet, the node erases all but one of the **1** bits in its address field such that the remaining **1** bits are all in different bit positions. The node then sends the x packets to the x different output links identified by the **1** bits. If one or more of the output links identified by the multicast packet is in contention with other packets, the node will not duplicate any packet and the packet will be deflected. The node first erases the second bit in part I of the address and the packet will be routed according to the deflection preference fields in the multicast address.

The above procedure ensures that there are sufficient output links before a node makes duplicates of a multicast packet in a branch node. It also prevents the duplicated packets from erroneous copying and duplication if they are deflected at a downstream part of the tree and re-visit the node.

2) Part I of address = **10**: The packet is a deflected multicast packet. The node processes the unicast address (part II of the address field) in the header and routes the packet accordingly. If the corresponding address field is all zero, the packet has reached the root node. The node then changes the second bit of part I of the address to **1** and proceeds to process the multicast part of the address.

3) Part I of address = **00** *or* **01**: The packet is a unicast packet. The node processes part II of the address field and routes the packet accordingly. If the corresponding address field is all zeros, the packet has reached the destination node.

In this scheme, a deflected multicast packet is temporarily converted to a unicast packet destined for the root of the multicast tree. Once it reaches the root node, it is converted back to a multicast packet. The scheme prevents any erroneous copying or duplication of a multicast packet after it is deflected.

B. The back-to-the-deflection-node (BDN) scheme

The address consists of two parts. Unicast part which is initially set to all **0**'s contains the unicast address of the deflection node. Multicast part contains the multicast address as described in Section III

When a node receives a packet, it first checks the corresponding field in unicast part of the address. If there is a non-zero element, the node routes the packet to the link identified by the **1** bit. If the corresponding field in unicast part of the address is all zeros, the node processes its address field in multicast part of the address similar to that described in Section IV-A.1 with the following modifications. If the multicast packet is routed properly, the node erases all the bits in unicast part of the address before routing the packet or packets. If the packet is deflected, the node put its address in unicast part of the address.

In this scheme, a deflected multicast packet is temporarily converted to a unicast packet destined for the deflection node. The packet is converted back to a normal multicast packet when it reaches the deflection node and is routed without deflection.

V. All Optical Implementation

We have proposed an all-optical implementation of a 2×2 deflection routing node [4] based on the all-optical crossbar switches demonstrated by Glesk *et al.* [6]. A $k \times k$ deflection routing node can be constructed in a similar fashion although the complexity of the node architecture will increase rapidly with k.

The BRN scheme requires a node to write or delete a bit all-optically. Note that no optical table look-up is required and the scheme minimizes the number of write and delete operations. The BDN scheme requires a node to add or remove part of the address that contains multiple bits. The scheme also requires

the node to store and retrieve its address all-optically. The required operations can be carried out using the all-optical crossbar switches and the bit controlled wavelength converters [7].

The addition and removal of a bit in principle can be carried out using Nonlinear Optical Loop Mirrors, and such operations have been demonstrated at over 100 Gbits/s [8]. Recently, all-optical header swapping has been demonstrated [9]. The method can be adapted to swapping only part of an address as required in the BDN scheme. We can store the address of the node in an optical fiber storage ring.

VI. Network Simulations

We study and compare the performance of the two proposed multicast deflection routing schemes using discrete event simulations on the slotted Manhattan Street Networks (MSNs). We use the batched mean method to compute the results. The batch size is 10^4 time slots. The first batch result is discarded. All simulations are run sufficiently long such that the 95% confidence intervals are less than 1% of the results. The time slots required to achieve the required confidence level are in the range of 2×10^6 to 10^7.

MSNs are two connected networks in a two-dimensional mesh with toroidal boundaries. Node transmission directions alternate with rows and with columns. Only one packet can be transmitted on a link per time slot. In general, we assume that a node has at most one new arrival packet per time slot but can receive two packets simultaneously. Apart from the multicast traffic, there is background unicast traffic. The probability that a new unicast packet arrives at a node is the unicast offered load. The unicast load offered to each node is assumed to be the same. We also assume that a node sends unicast packets uniformly to each node in the network except itself.

For an $r \times c$ MSN, there are r rows and c columns of node. We label them from left to right, and top to bottom. The node in the upper left corner is labeled 1, and that in lower right corner is labeled N, *i.e.*, $N = r \times c$. Figure 2 shows part of a 10×10 MSN and a multicast tree: $\{(1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 17 \rightarrow 27), (1 \rightarrow 11), (3 \rightarrow 13), (5 \rightarrow 15)\}$. The arrows in Fig. 2 represent the transmission directions of the links. The multicast tree is shown by the shaded nodes and thick arrows. Node 1 is the root node of the tree and is denoted by a double circle. Nodes 1, 3, and 5 are the branch nodes. Node 1 generates a new packet according to the total offered load. The packet is then classified at random according to the assigned ratio of multicast to unicast offered loads. If the packet is a multicast packet, it will be forwarded to the nodes on the tree.

In MSNs, a node always routes the packets to their destinations using the shortest paths if possible. If two packets of the same type contend for the same output link, one of them is deflected. A packet that has the same path length to the destination node from either of the output ports is always deflected if the other packet does have a preference. These packets are said to have a *don't care* routing preference.

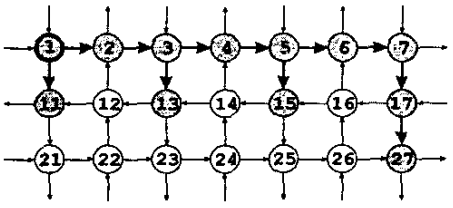

Fig. 2. A 10×10 MSN multicast tree: Node 1 is the root of the tree. Nodes 1, 3, and 5 are the branch nodes.

Traditionally, the nodes in MSNs are assumed to have the capability to classify the packet types from the packet destinations. For all-optical MSNs, we use the priority schemes proposed in [4] to provide the same feature. We assume that a priority bit is associated with each address field. The priority bit is set to 1 if the packet's desired output link is the only shortest path. Otherwise, it is set to 0. This modification slightly increases the header length but simplifies the node design. In the simulations, unicast packets are assumed to have the *don't care* routing feature using the priority approach. Multicast packets do not have this feature because they should be routed according to the multicast tree. We treat undeflected multicast packets with all priority bits set to ones.

A. Simulation Results

Intuitively the BDN scheme is expected to have better performance because the resource consumption per deflection of multicast packet is smaller. Simulation results, however, show that the opposite is true. Figure 3 plots the throughput–delay curves of the two schemes using the multicast tree shown in Fig. 2. We set the offered load of the background unicast traffic to 0.1 and measure the throughput and delay under different multicast offered loads. The asterisks and circles represent the results of the BRN and BDN schemes, respectively. From Fig. 3, the delay of the multicast packet in the BDN scheme is lower than that of the BRN scheme when the throughput is below 0.16. When the throughput is larger than 0.16, the delay for BDN increases rapidly. The maximum achievable throughput of the BDN scheme is found to be about 0.21 while that of the BRN scheme is 0.28. Although BRN has higher multicast throughput, its throughput of background unicast traffic is also higher as shown in Fig. 4. The BRN scheme consumes less system resources. We have similar observations on simulations of different network sizes and different multicast tree topologies. Due to space limitation, we have not showed these results.

We observe that the deflection probability of the multicast packets at the branch nodes of the multicast tree is much higher in the BDN scheme than in the BRN scheme. It is because a deflected multicast packet in the BRN scheme is sent back to the root node. The transit packet intensity at the root node therefore increases with the deflection probability of the multicast packets. Since transit packets have higher priority than new arrival packets, the deflected multicast packets limit new packet

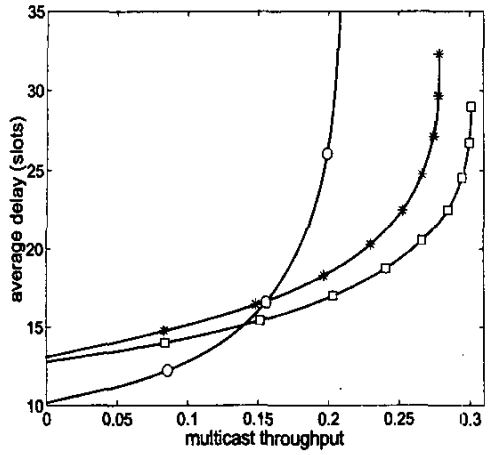

Fig. 3. Throughput–delay curves of the multicast traffic using the multicast tree shown in Fig. 2. The asterisks and circles represent the results of the back-to-the-root-node (BRN), and the back-to-the-deflection-node (BDN), respectively. The squares represents the hybrid scheme. Background unicast traffic offered load is 0.1. The 95% confident intervals are smaller than 1% of the values.

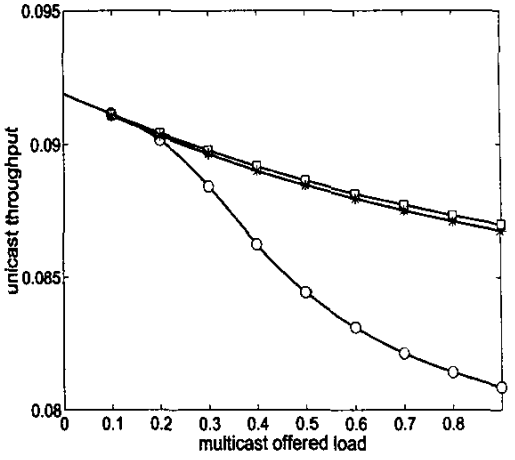

Fig. 4. Throughput curves of the background unicast traffic using the multicast tree shown in Fig. 2. The symbols and conditions are the same as those used in Fig. 3

admission at the root node. Such negative feedback does not exist in the BDN scheme unless the branch nodes are close to the root node. Our simulations show that the performance of the BDN scheme deteriorates if the branch nodes are moved further downstream from the root node. Another reason for the better performance of the BRN scheme is that the deflected multicast packets re-enter the tree at the root node only. This prevents the deflected multicast packets from contending with transit packets of the same type. In BDN, however, the deflected packets contend with normal multicast packets on the tree and thus the deflection probability of the multicast packets is increased.

Although the performance of the BDN scheme is in general inferior, it consumes less resources at the non-branch nodes. A hybrid of the BRN and BDN schemes is expected to have better performance than either schemes alone. In Figs. 3 and 4, the squares denote the results of a hybrid multicast scheme in which we use BRN scheme at the branch nodes and the BDN scheme at other nodes. Figure 3 shows that the hybrid scheme has a better throughput and delay than the two original schemes. The resource utilization of the hybrid scheme is also improved as shown in Fig. 4. All-optical implementation of the hybrid scheme is of course a bit more difficult than in the two proposed schemes.

VII. Conclusion

In this paper, we investigate multicast services on deflection-routed all-optical packet-switched networks. We proposed two multicast schemes that are compatible with deflection routing and can be implemented by demonstrated optical signal processing technology. The first scheme sends a deflected multicast packet back to the root node of the multicast tree while the second one sends the packet back to the deflection point. We compare the performance of the two multicast schemes by discrete event simulations using Manhattan Street Networks. We find that the back-to-the-root-node scheme in general performs better than the back-to-the-deflection-node scheme because the former has a smaller deflection probability at the branch nodes of the tree. As the back-to-the-deflection-node scheme consumes less system resource per deflection, we find that a hybrid of the two schemes has the best system performance at the expense of increased implementation complexity.

References

[1] 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, No. 12, pp. 2146-2168, 1998.

[2] L. H. Sahasrabuddhe, and B. Mukherjee, " Light-trees: Optical multicasting for improved performance in wavelength-routed networks," *IEEE Comm. Mag.*, Vol. 37, No. 2, pp. 67-73, 1999.

[3] M. Jeong, C. Qiao, and Y. Xiong, "A reliable WDM multicast protocol for optical burst-switched networks," *Optical Networks Magazine*, Vol.2, No. 2, pp. 29-40, 2001.

[4] C.Y. Li, P.K.A. Wai, X.C. Yuan, and V.O.K. Li, "Deflection routing in slotted self-routing networks with arbitrary topology," *Proceedings of ICC 2002*, pp. 2781-2785, New York, April 28 - May 2, 2002.

[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," *Proceedings of ICC 2001*, Vol. 7, pp. 2155-2159, Helsinki, Finland, 11-15 June 2001.

[6] I. Glesk, K. I. Kang, and P. R. Prucnal, "Demonstration of ultrafast all-optical packet routing," *Electron. Lett.*, , Vol. 33, No. 9, pp. 794-795, 1997.

[7] P.K.A. Wai, L.Y. Chan, W.H. Chung, and H.Y. Tam, "Bit-controlled wavelength conversion using injection locking," in preparation.

[8] M. Eiselt, W. Pieper, and H.G. Weber, "SLALOM: Semiconductor laser amplifier in a loop mirror," *J. Lightwave Technol.*, Vol. 13, No. 10, pp. 2099-2112, 1995.

[9] D.J. Blumenthal, et at., "All-Optical Label Swapping Networks and Technologies," *J. of Lightwave technol.*, Vol. 18, No. 12, pp. 2058-2075, 2000.