Buffering and Flow Control in Optical Switches for High Performance Computing

Xiaohui Ye, Roberto Proietti, Yawei Yin, S. J. B. Yoo, and Venkatesh Akella

***Abstract*—We investigate the advantages and disadvantages of different loopback buffer architectures for optical switches and compare their performance via simulation. The simulation results show that, without the use of virtual output queuing, the head-of-line blocking can be alleviated by wavelength parallelism when each separate queue in a loopback buffer has multiple transmitters. Furthermore, the proposed two-level flow control can eliminate packet drop at the switch, resolve rate mismatching due to output queuing at switch outputs, and ensure that congestion occurring at the hotspot port will not affect the performance of non-congested ports.**

***Index Terms*—Datacenter networks; Flow control; Hybrid optical switch; Loopback buffer.**

I. Introduction

The insatiable demand for higher bandwidth and lower power consumption in high performance computing applications has sparked an interest in optical switches. In this paper, high performance computing refers to both traditional scientific computing and emerging warehouse scale computing [1], so-called datacenters. Unlike traditional data communications, computing applications impose *four* new requirements on optical switches: first, the node-to-node latency is expected to be in the tens or hundreds of nanoseconds instead of the milliseconds or the hundreds of microseconds, as in Internet applications; second, packets cannot be dropped, i.e., a 0% packet loss should be ensured; third, very bursty traffic with a load reaching 90% at times must be handled; and, finally, scalability to thousands of nodes must be supported. Lack of random access memory, which can store packets for *arbitrary* amounts of time, in the optical domain, not only makes the implementation of a synchronous switch difficult, but also makes ensuring 0% packet loss particularly challenging. The (fixed) fiber delay lines, widely used in optical switches for data communications, can cause significant performance degradation in an asynchronous switch, since they cannot delay the packets for arbitrary amounts of time. Therefore, electronic buffers are essential in an optical switch in order to achieve low-latency switching. In this paper we discuss the design of a *hybrid* optical switch, which uses a limited amount of electronic memory together with the flow control scheme to eliminate packet loss and achieve low-latency and high-throughput switching.

In the proposed hybrid optical switch, the electronic buffer, which is used to handle contention resolution, is placed in the loopback path, referred to here as the loopback buffer. The compute nodes that are connected with the proposed switch also have the usual electronic buffers at their inputs and outputs. We investigate the following questions in this paper. What are the architectural alternatives for implementing the loopback buffer? What should be its size? What is the impact of the loopback buffer on the scalability of the hybrid optical switch? Given that optical data rates can be 10 Gbps or higher, how can electronic memory keep pace with them? Does the loopback buffer become the bottleneck in terms of the cost, power, and performance (especially latency) of the hybrid optical switch? In a network design supporting high performance computing applications, buffering cannot be studied without consideration of flow control. We can exploit the physical proximity of the compute nodes in a warehouse scale computer to develop link-level flow control schemes. Specifically, we propose a two-level flow control scheme, including the loopback buffer flow control and the optical channel adapter (OCA) flow control, and study the interplay between the two.

Our studies are based on a hybrid optical switch that utilizes a passive optical device, called the arrayed waveguide grating router (AWGR). The AWGR is an optical interconnect that can realize an all-to-all communication, in which each input can communicate with different outputs on different wavelengths simultaneously, and all inputs can reach the same output on different wavelengths. Although previous optical switch designs used the AWGR as a non-blocking switching fabric in order to guarantee optical transparency, the AWGR can naturally be used to implement output queuing because of its inherent wavelength parallelism. The architecture of this switch is described in [2].

The specific contributions of this paper are as follows:

1. We compare the continuum of loopback buffer strategies, from a fully distributed architecture called the distributed loopback buffer (DLB) to a fully shared configuration called the shared loopback buffer (SLB). The mixed loopback buffer (MLB) represents a tradeoff between the two extremes. We examine the tradeoffs in terms of the numbers of transmitters and receivers, the loopback memory bandwidth, the numbers of couplers and optical MUXes and DEMUXes, and the number of ports of the AWGR that are occupied by the buffering.

Manuscript received April 29, 2011; revised June 29, 2011; accepted June 30, 2011; published July 25, 2011 (Doc. ID 146735).

The authors are with the Department of Electrical and Computer Engineering, University of California, Davis, California 95616, USA (e-mail: xye@ucdavis.edu).

Digital Object Identifier 10.1364/JOCN.3.000A59

2. We propose a two-level flow control scheme to prevent buffer overflow, resolve the data rate mismatching caused by output queuing, and prevent the congestion occurring at certain ports from influencing other ports.
3. We evaluate the performance of the hybrid optical switch under both uniform random traffic and hotspot traffic via simulation.

The simulation shows that both the DLB and the MLB can provide lower latency than can the SLB proposed in [2], while the MLB occupies fewer AWGR ports and uses fewer tunable transmitters than does the DLB. The simulation results also confirm that the proposed two-level flow control not only prevents packet drop, but also ensures that the congestion occurring on the hotspot port does not impact the performance on non-congested ports.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III presents an overview of the architecture of the hybrid optical switch. Section IV discusses in detail the different loopback buffer architectures and associated flow control mechanisms. Section V evaluates the performance of the hybrid optical switch with the proposed two-level flow control. Section VI concludes this paper.

II. Related Work

Conventional datacenter networks are built in a hierarchical manner, with a large number of cheap, low-speed, small-radix switches at the bottom level to connect with the end nodes, and a few expensive, powerful, large-radix switches residing at the top level to aggregate and distribute the traffic [3]. Recently, network architects have adopted fat tree and Clos topologies to provide high aggregate bandwidth by constructing a switching network with small-radix switches [4,5]. Achieving low power consumption, low latency, and high throughput under high input loads is the key challenge with electrical switches. Farrington *et al.* [6] and Wang *et al.* [7] suggested placing MEMS-based optical circuit switches in parallel with electrical switches in the core network to carry slow changed inter-pod traffic [6] or latency insensitive traffic [7], thus reducing power consumption and cost. Nevertheless, those designs do not address the challenges of achieving low latency and high throughput under high input loads. Dragonfly [8] uses multiple small-radix routers to form large-radix switch groups, so that there is a connection between any two groups. Although Dragonfly can support high input load under uniform random traffic, it still saturates at moderate input loads when each node in one group sends traffic to a randomly selected node in another group.

Several efforts have been made to use optical technologies in the network design for computing applications. Louri *et al.* proposed SOCNs/SYMNET [9,10], a multi-level hierarchical architecture for a large-scale optical crossbar network and a tree-based address distribution sub-network. Although SOCNs/SYMNET can connect to a large number of nodes, parallel communication is not utilized and the system throughput is limited, since the optical token controlled address broadcasting scheme allows for transmitting messages only serially. Gemini [11] is an optical/electrical dual banyan network. The optical banyan network delivers long messages, while the electrical network transmits control signals and short messages. The drawbacks of Gemini are that the banyan network is a blocking network and that the optimal scheduling for a large-scale banyan network is complicated.

The OSMOSIS utilizes semiconductor optical amplifiers (SOAs) to realize a synchronous optical crossbar switching fabric with the use of a broadcast-and-select data path combined with both space- and wavelength-division multiplexing [12,13]. Strictly speaking, the OSMOSIS still uses the store-and-forward mechanism and adopts input and output queue structures, which are commonly used in electrical switches. Although the optical switching fabric allows the OSMOSIS to have two receivers at each output, thus sustaining high input load, the power requirements of the OSMOSIS can be very high because of its broadcast-and-select architecture—signals are delivered to every select unit, even though only one unit selects the signal. The Data Vortex is a distributed interconnection network architecture [14,15] based on deflection routing. To prevent packet drop when contention occurs, the packet is deflected to another output and an access control is adopted to ensure that the network will not carry traffic beyond its capability. In other words, Data Vortex treats the deflection route as temporary network storage. Due to the deflection routing and access control, Data Vortex saturates before the offered load exceeds 50% [16]. In addition, as the number of nodes increases, the end-to-end latency becomes large and non-deterministic.

AWGR-based optical switches and optical routers with packet switching capability have been investigated for a number of years. Previous work [17–23] mainly focused on the application of the AWGR in access networks and in telecommunication/IP networks. An AWGR serves as a non-blocking switching fabric in many switch architecture designs. However, the wavelength parallelism on AWGR outputs is not explored in those designs. Because each AWGR output is connected with a fixed wavelength converter to convert the signal to a particular wavelength in order to ensure wavelength consistency on the input and output fibers, the occurrence of multiple packets is not allowed. Since no practical optical buffer is yet available, the store-and-forwarding scheme, which is commonly used in the electrical switch, cannot be duplicated in the optical domain. A fiber delay line (FDL) is commonly used to resolve the contention, provide temporary storage, and allow packets that cannot gain the resource to compete for the resource at a later time. Use of the FDL in resolving the contention helps to significantly reduce the dropping probability, but packet loss is still possible and cannot be eliminated. In addition, the FDL cannot provide arbitrary delays, which is more critical in asynchronous switching. The resource may be available, but the delayed packet cannot access it, since the packet is still traveling through the FDL.

The datacenter optical switch (DOS) [2] is an optical hybrid switch that adopts the AWGR as the switching fabric and utilizes wavelength parallelism to achieve output queuing. The SLB with N parallel transmitters and N parallel receivers can simultaneously receive contended packets from multiple input ports and transmit packets to multiple output ports when resources are available. However, the SLB limits the scalability of the DOS due to the bandwidth requirement of the shared buffer. The DOS also assumes that end nodes have asymmetric transmitting and receiving rates, which may be true only for some systems.

A theoretical model for a synchronous optical switch with electronic buffers is presented and the packet scheduling is discussed in [24–26]. The analysis shows the effectiveness of adding electronic buffers in the optical switch. In practice, without input queuing, it is difficult to realize synchronous switching, except in an on-chip network, because it is difficult to ensure that packets sent from different end nodes arrive at the switch inputs in the same clock cycle. Each cycle in a synchronous switch must be long enough to accommodate path differences among end nodes, which in turn degrades the system performance when the data rate is high or the packet size is small. Therefore, we adopt asynchronous switching for the proposed switch. In addition, we focus on the buffer and flow control design to eliminate packet drop, whereas packet drop is still allowed in [24–26].

III. Hybrid Optical Switch Overview

Figure 1 shows an optical hybrid switch, and, for comparison, a generic electrical switch is shown in Fig. 2. It is well known that output queuing can achieve lower switching latency and higher throughput than can input queuing. However, when the line rate and the switch radix increase, it is difficult to realize output queuing because it is difficult to realize a switching fabric that can accommodate such a high aggregated bandwidth. Therefore, electrical switch designs focus on complicated input queuing structures, such as virtual output queue (VOQ), and associated multi-stage arbitration schemes. On the other hand, the AWGR-based switching fabric has the unique strength of wavelength parallelism, allowing optics to cross over and propagate in parallel in different colors. The AWGR-based switching fabric can easily realize the output queue, provided that a $1:N$ optical DEMUX with N receivers is available at each AWGR output. However, requiring N receivers at each output may not be practical or scalable, since a total of N^2 receivers for the whole switch is required. We assume that each output is equipped with a $1:k$ optical DEMUX and k receivers with $k<N$, thus realizing an output queuing with a speedup of k. We define a *wavegroup* as a set of wavelengths that will emerge from the same output port of the $1:k$ optical DEMUX.

In electrical switches, a packet is first stored at the input queue, and the input queue sends the request to the control plane for arbitration. If the input queue cannot gain the resource, it can hold the packet for an arbitrary amount of time until the packet can be switched. After it gains the resource, the packet is delivered to the desired output port through the switching fabric. In an optical switch, the store-and-forward mechanism cannot be applied due to the lack of optical buffers. The packet travels toward the switching fabric without being "stopped." The arbitration is conducted after the label is extracted, and the decision must be made before the packet reaches an input of the fabric. If the packet cannot gain the resource, it will be directed to the loopback buffer. Because the loopback buffer uses electronic buffers, the packet can be stored for an arbitrary time before it goes to the desired output. Note that an optical switch uses a *forward-store* strategy, as opposed to the store-and-forward strategy employed in an electrical switch, so that *only* packets that are denied access due to contention are stored, as opposed to *all* packets in an electronic switch. This strategy results in significant benefits both in terms of latency and power consumption.

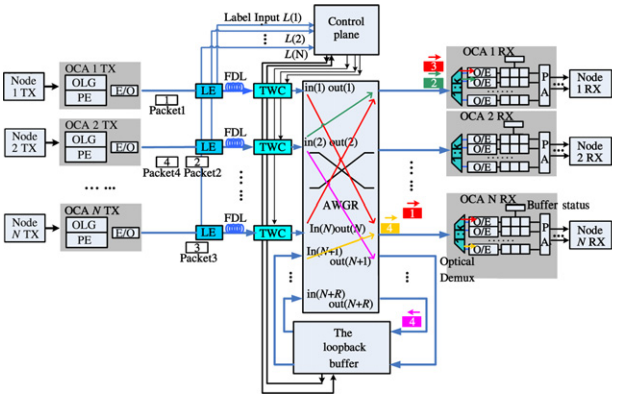

Fig. 1. (Color online) Overview of the proposed optical hybrid switch architecture. OLG: optical label generator; PE: packet encapsulation; LE: label extractor; FDL: fiber delay line; PA: packet aggregation; O/E: optical-to-electrical converter; E/O: electrical-to-optical converter; TX: transmitter; RX: receiver; $L(i)$: label from node i.

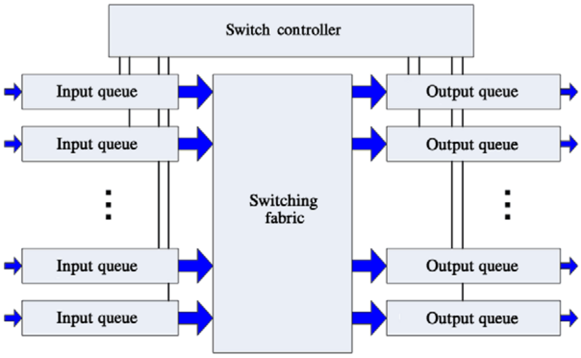

Fig. 2. (Color online) A block diagram of a traditional electrical switch.

The proposed optical hybrid switch uses an AWGR as the core of the switching fabric. The core of the switch contains, in addition to the AWGR, tunable wavelength converters (TWCs), an electrical control plane, an electrical distributed loopback buffer, label extractors, and FDLs. Between the switch and each end node, there is an optical channel adapter (OCA) that serves as the media interface.

At the OCA transmitter (OCA TX), each packet sent from the end node is modulated on a wavelength as the optical payload. The OCA TX also generates an optical label based on the content from the packet header and modulates this label on a different wavelength. By using wavelength division multiplexing (WDM), the optical payload and optical label can be transmitted, with different data rates, in parallel on different wavelengths. Because the data rate used to transmit optical labels can be much lower than the data rate for the optical payload, the electrical control plane can operate at a much lower frequency, thus reducing the complexity of the control plane circuits.

The label extractor (LE) separates the optical labels and the optical payloads. While the optical payload is traveling through the fixed FDL toward the TWC, the optical label is sent to the control plane for processing and arbitration. After arbitration, the control plane generates control signals for the TWC, so that the TWC can be set to the proper state before the payload arrives. The length of the fixed FDL is set to compensate for the latency caused by label processing, arbitration, and TWC tuning. The TWC converts the optical payload to the proper wavelength so that the optical payload can reach the output determined by the arbitration. The TWC can be a simple combination of an optical wavelength converter [27] and a tunable laser. If the packet cannot gain the resource on the desired output, the control plane directs it to the loopback buffer.

The loopback buffer must be able to *receive* packets from any input port and *send* packets to any output port. In Section IV, we propose two loopback buffers, the DLB and the MLB, and compare them with the SLB used in the DOS [2]. Since we want to ensure that no packet drop occurs, we also discuss the flow control schemes for different loopback buffers.

At the OCA receiver (OCA RX), the optical DEMUX separates multiple wavelengths. Each optical DEMUX output connects to a separate O/E converter and a dedicated buffer, so that packets arriving on different wavelengths can be received simultaneously. The received packets will then be sent to the end node. Packet aggregation may be required, depending on the receiving capability of the end node.

The DOS design assumes that the output rate for the OCA RX is the same as its aggregated input rate, meaning that the capability of the end node in transmitting and receiving data is asymmetric. This asymmetry is the case for some real systems, e.g., ADSL and cellular service. But to make the system compatible with different end hosts, we must eliminate this assumption of rate asymmetry. Therefore, we may have rate mismatching between the OCA RX input (the same as the AWGR output) and the OCA RX output. To resolve this rate mismatching and prevent buffer overflow at the OCA RX, it is essential to add another flow control, called the OCA flow control, between the OCA RX and the control plane. Section V discusses the OCA flow control and evaluates the system performance with the two-level flow control under different traffic models.

The arbitration in the optical hybrid switch is much simpler and more scalable than is the arbitration for the same size electrical switch in the following aspects. First, because no packet is buffered at the input and all labels are processed just in time, no input will generate repeated requests, except for those requests from the loopback buffer controller. Second, since the VOQ is not used, every input connected with an end node makes only one request, and the input accepts the grant when notified. Therefore, a simple 2-phase arbiter is sufficient in the optical hybrid switch, instead of the 3-phase arbiter, as in a conventional VOQ-based switch. Moreover, the multiple iterations (on the average $O(\log_2 N)$) that are required in a traditional electronic switch adopting VOQ are not necessary. A single iteration is sufficient. Third, and most important, due to the wavelength parallelism offered by the AWGR fabric and the cyclic nature of the AWGR operation, the number of inputs contending for a given output in the worst case can be reduced by a factor of k, where k is the number of concurrent wavelengths allowed for each AWGR output. As the contention occurs only inside a small contention group [2], not between different contention groups, contentions occurring at different contention groups can be processed in parallel with the use of $N * k$ parallel arbiters. Optical parallelism in the optical hybrid switch allows for breaking the global arbitration and contention resolution into small processes in parallel.

IV. Loopback Buffers for Optical Hybrid Switches

The loopback buffer in the optical hybrid switch plays an important role in contention resolution. If the loopback buffer does not have enough resources to hold contended packets, the packets that fail to reach the desired output will be dropped for retransmission, which will incur degraded computing and datacenter performance. In this section, we will first review the SLB proposed in [2] and then discuss the proposed DLB and MLB. We then propose flow control schemes to prevent buffer overflow for different loopback buffers. We also evaluate and compare the performance of different loopback buffers via simulation.

A. The Shared Loopback Buffer

The SLB has the structure shown in Fig. 3. A $1:N$ optical DEMUX and N receivers are necessary, because the SLB may receive delayed packets from different inputs concurrently on different wavelengths. An $N:1$ optical MUX and N transmitters are also required to allow the SLB to send delayed packets to different outputs on different wavelengths concurrently. In the SLB, the optical DEMUX and MUX can both be realized by a $1:N$ arrayed waveguide grating (AWG). The packets received on different wavelengths will be copied to the shared memory in parallel. All delayed packets will be stored in the shared memory before transmission. The queuing structure in the shared memory is organized based on outputs, since the packets destined for the same output should be sent out serially. As, on average, we need to store only a few packets for each input, even under hotspot traffic, with appropriate flow control, the size of the shared memory can be small. The transmission for a delayed packet can often start before the entire packet arrives at the SLB. When a grant is given to the SLB for a particular output, the SLB sends out all delayed packets going to that output serially.

The benefits of the SLB are the following. First, the total size of the buffer can be small, as one input port can cause contention at only one output, but not at multiple outputs, at any time. Second, the SLB can have a simple buffer controller; since packets are stored based on outputs, no further scheduling among delayed packets is required. Nevertheless, the main drawback of the SLB is that the shared memory limits the scalability of the optical hybrid switch, since the required memory I/O bandwidth is proportional to both the switch radix and the data rate on each wavelength.

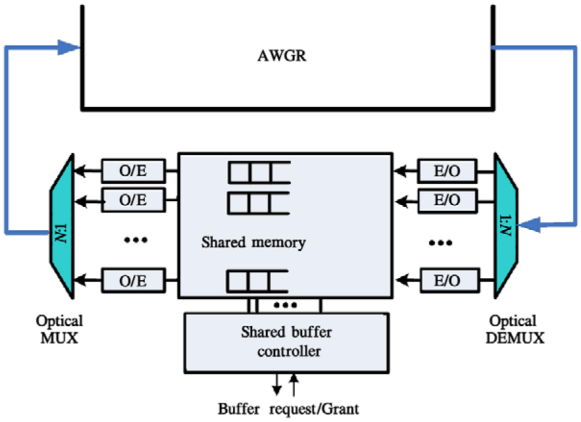

Fig. 3. (Color online) The shared loopback buffer.

Although the use of an electronic buffer can prevent packet drop due to contention, flow control is still necessary to prevent buffer overflow. On–off flow control is adopted for all flow control schemes discussed in this paper, because it is simple and effective. Usually congestion occurs on a few ports, not on all ports. One input can trigger contention on only one output at a time, not on many outputs concurrently. It is desirable to have a flow control scheme that can resolve congestion quickly on congested output ports and prevent that congestion from affecting non-congested output ports. To achieve this goal, flow control messages sent to end nodes must identify which output is congested. Then the end node can hold packets destined for those congested outputs but continue transmitting packets to other outputs. To achieve this differentiation, we must measure the queue depth for each output-based queue and send out the flow control message based on the status of each individual queue, instead of measuring only the total buffer size [2]. The new flow control will have two sets of thresholds, one for each individual queue and the other for the entire shared memory. For a particular queue, the flow control will be triggered when either of the high thresholds is reached; the flow control will be turned off when both the individual queue size and the entire buffer size are lower than their low thresholds.

As packets stored in one queue may come from any input, we cannot estimate which node will send packets to this output in the future. When the queue size for one output reaches the high threshold, we must send flow control messages to all end nodes, so that no end node will send packets to that particular switch output upon receiving the message. The flow control message contains an N-bit payload plus a 2-byte header, including the preamble and the destination address. Each bit in the payload corresponds to the status of one queue: "1" means the flow control is ON for that queue, and "0" means the flow control is OFF. When an end node receives the flow control message, it will suspend or resume the transmission for one or multiple destinations according to the flow control message.

The flow control message can be inserted into the head of the queue after generation, so that it can be sent to the end node as soon as possible. If a new flow control message is generated when the old flow control message is still waiting for transmission at the head of the queue, we can replace the old flow control message with the new one, so that the latest queue status can be distributed and only one flow control message will wait at the head of the queue at any time.

In the worst case, the flow control packet is inserted just as the corresponding queue starts the transmission for a delayed packet, and the same situation applies to all queues. Then all flow control packets will undergo a queuing latency, which equals the packet transmission latency, before being sent out. Considering that the length of the flow control message is shorter than are normal packets, the total time for a flow control message being sent out, received, and processed should be less than the time taken to transmit a normal packet. Therefore, in the worst case, when all nodes send packets to one destination continuously (at most k packets can go directly to the desired output and the remaining $N-k$ packets will be delayed), we can prevent packet drop if the shared buffer can hold $2*N+High\_Threshold_{SLB}$ packets. In simulations, the high threshold for the entire buffer is set to $16*N$ packets and the low threshold for the entire buffer is set to $8*N$ packets, respectively, while the high threshold for each individual queue is set to $6*N$ packets and the low threshold for each individual queue is set to $3*N$ packets, respectively.

B. The Distributed Loopback Buffer

The loopback buffer will become more scalable if the queues can be organized based on the input ports and not on the output ports. The drawbacks of the input-based buffer are the following. First, the buffer controller design will become more complicated, as contention may occur among queues for different inputs. Second, head-of-line (HOL) blocking may occur, and end-to-end latency may increase. However, the input-based buffer can be realized in a distributed manner with multiple separate buffers. For each buffer, the required memory I/O bandwidth can be reduced by a factor that is proportional to the number of separate buffers. Therefore, the loopback buffer can support a switch with a higher port count and a higher data rate.

The proposed DLB has N separate memory units to realize N separate queues, with each unit serving delayed packets from one particular switch input. In the simplest case, each queue has one transmitter and does not adopt VOQ, as shown in Fig. 4. The head packets at different queues compete for the resource if their desired switch outputs are the same. In addition to the request and grant exchange between the buffer controller and arbiters in the control plane, the buffer controller needs to collect requests from each queue and allocate grants that are received from the control plane among different queues. The buffer controller can still use two-stage simple arbitration (only the request stage and the grant stage without the grant acknowledgment stage), as VOQ is not adopted. On the other hand, HOL blocking is inevitable.

To eliminate the HOL blocking, we can adopt VOQs at each queue. But arbitration will become much more complicated, since some grants may not be accepted, making three-stage arbitration necessary. Instead of using VOQs with complicated arbitration, we can also exploit the intrinsic wavelength parallelism to alleviate the effect of the HOL blocking and keep the arbitration relatively simple. We can deploy multiple

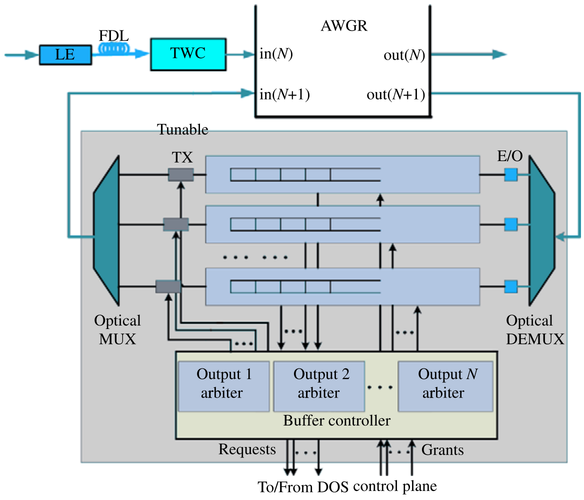

Fig. 4. (Color online) The distributed loopback buffer occupies one AWGR input, with each queue having one transmitter.

transmitters for each queue, thus making it capable of sending multiple packets to different switch outputs on different wavelengths concurrently. With respect to scalability, it is not desirable to have N transmitters for each queue, although this helps to eliminate the HOL blocking and achieve the minimum latency. Here we consider the deployment of only a fixed number of transmitters for each queue. To alleviate the HOL blocking, we need to ensure that packets transmitted by different transmitters of a queue, at any time, all go to different outputs.

The complexity of arbitration for the VOQ-based input queue comes primarily from the fact that each input queue may make more requests than it can actually accept. If we impose the restriction that only the packets that gain the transmitters can make the request, the queue can then accept all grants assigned to the buffer controller. With multiple transmitters, each queue can make multiple requests, and the chance that all requests will be denied will be significantly lower than the chance of one request being rejected.

The DLB uses tunable transmitters, because packets stored at one queue may go to different AWGR outputs. If the loopback buffer occupies only one AWGR input, we cannot use an $N:1$ AWG as the optical MUX, because the AWG requires a certain wavelength at a certain input. The use of couplers to combine optical signals will cause large power losses. One solution is that each queue in the DLB be connected to a separate AWGR input. Then a $2*N \times 2*N$ AWGR is needed to connect to N end nodes and the DLB, as shown in Fig. 5. The delayed packets from one end node (coming from one AWGR input) are all directed to a certain AWGR output connecting to the queue associated with the input. Couplers are used to realize the optical MUX if each queue has multiple transmitters.

We can interleave the inputs connecting with the end nodes and the inputs connecting with the loopback queues, so that, in each contention group, half of the inputs connect with end nodes and half of the inputs connect with loopback queues. With this connection, k wavegroups for one output will be used equally. Multiple delayed packets going to the same output can be sent out concurrently from different queues on different wavelengths, thus reducing the end-to-end latency. Moreover, through the careful association of the inputs connected to end nodes with the inputs connected to queues, the waiting time for a packet at the DLB can be further reduced. For example, we can connect an end node with the input i and the associated queue with the input j $(0 \le i,j \le 2*N)$; if i and j satisfy the relation $j = \mathrm{mod}(i+N+1, 2*N)$, the inputs i and j will belong to different contention groups. Therefore, if a packet first reaches the input i but the request is rejected, it will make a request to a different contention group when it arrives at the DLB, because it will use the input j.

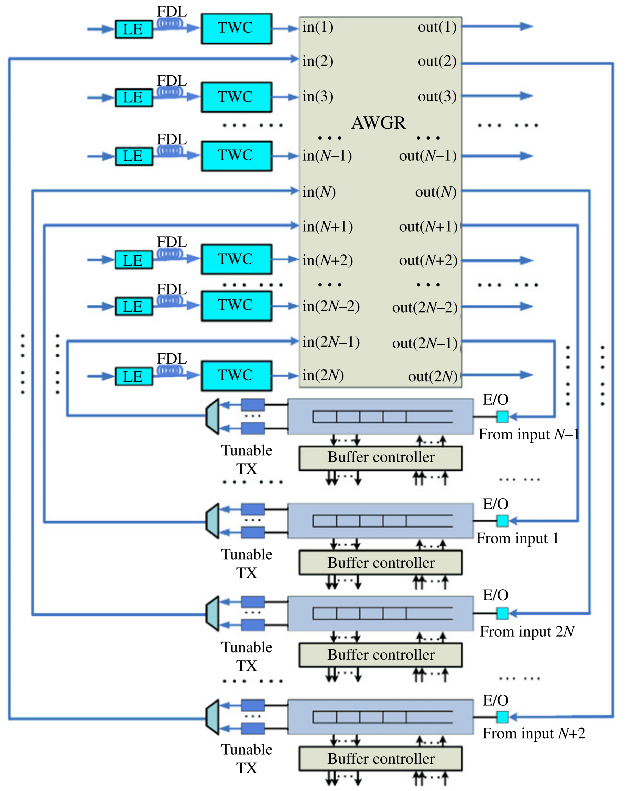

Fig. 5. (Color online) The distributed loopback buffer occupies N AWGR ports, with each queue having multiple transmitters.

In contrast to the SLB, in which the packets stored in one queue (for one particular output) may come from any input, the packets stored in one queue of the DLB all come from the same end node. Although the draining rate for the queue depends on the activities of the other queues, the input rate for the queue depends solely on the sending rate of the associated end node and the contention probability. Suspension of the transmission at the corresponding end node can prevent an increase in the queue depth. Therefore, for one specific queue, we need to deliver the flow control message only to the corresponding end node, not necessarily to other end nodes. In other words, if the queue serves delayed packets from the input i, the flow control message for the queue will be sent only to the output i. To provide differentiated flow control, each queue must record the number of packets destined for different outputs. Accordingly, the DLB has two sets of thresholds, one for the queue and the other for the individual output. Similarly to the SLB, the flow control message contains an N-bit payload and a 2-byte header; it will be inserted in the head of the queue after generation; the new flow control message will replace the old one if the old message has not been transmitted when the new one is generated. Applying an analysis similar to the one used for the SLB and assuming that flow control packets always have the highest priority, in the worst case, we can prevent packet drop if one queue can hold $2+High\_Threshold_{DLB}$ packets. The DLB must be able to hold at least $N*(2+High\_Threshold_{DLB})$ packets in total. In simulations, we set $High\_Threshold_{DLB}$ to 16 and $Low\_Threshold_{DLB}$ to 8, whereas the high threshold for an output is set to 6 and the low threshold for an output is set to 3.

C. The Mixed Loopback Buffer

The DLB can achieve lower end-to-end latency than can the SLB. The I/O bandwidth requirement for each memory unit increases only when the data rate increases and not as the port count increases. However, the DLB occupies N AWGR ports to support the queues for N end nodes, while the SLB occupies only one AWGR port. The DLB also requires more transmitters than does the SLB in order to alleviate the HOL blocking. To achieve the benefits from both the SLB and the DLB while mitigating their disadvantages, we propose the MLB, as shown in Fig. 6. The MLB still occupies multiple AWGR inputs to support multiple separate queues, so that each queue occupies one AWGR input. Unlike the DLB, in which each queue serves only one end node, each queue in the MLB serves r end nodes and connects to r outputs of the $1:N$ optical DEMUX. Therefore, the MLB occupies N/r AWGR inputs if the switch connects to N end nodes. Again, each queue in the MLB can have multiple transmitters to alleviate the HOL blocking. Since the MLB has only N/r queues, even if each queue has r tunable transmitters, the MLB has N tunable transmitters in total, which is much fewer than the number of tunable transmitters required by the DLB. In the MLB, we can adopt scheduling and arbitration similar to those used in the DLB to schedule the transmission for delayed packets. At each queue, a delayed packet must first gain a transmitter and then make a request to the control plane. Since the MLB occupies multiple AWGR inputs, as in the DLB, the MLB can send multiple delayed packets to the same AWGR output if we interleave the ports connecting with the end nodes and those connecting with the MLB. Furthermore, through the careful assignment of the inputs to the MLB and association of the inputs connected to end nodes with the inputs connected to the queues, a delayed packet can make a request to a different contention group upon arriving at the MLB. For example, a set of AWGR inputs $\{j|\mathrm{mod}(j,r+1)=0, 0\leq j<N+N/r\}$ can be used to connect with N/r queues of the MLB; the queue connected to a particular input j can be used to serve delayed packets from a set of AWGR inputs $\{i|i=\mathrm{mod}(j+N/2+N/(2*r)+h,N+N/r), 1\leq h\leq r\}$ connected to the end nodes, so that the set of inputs and the input connected to the corresponding queue always belong to different contention groups. Although the memory used in the MLB requires more I/O bandwidth than does the memory used in the DLB, the memory used in the MLB requires much less I/O bandwidth than does the memory used in the SLB, if r is a fixed small number. In simulations, we set r to 4 and 16.

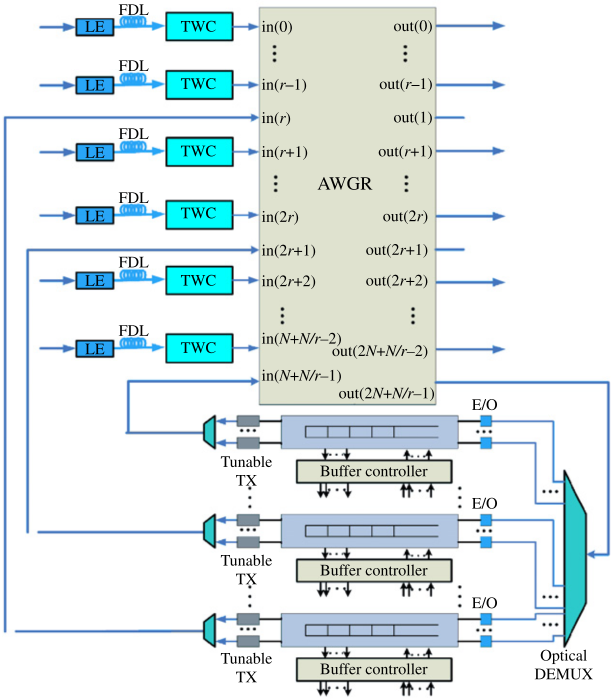

Fig. 6. (Color online) The mixed loopback buffer occupies N/r AWGR inputs, with each queue having multiple transmitters.

The flow control for the MLB is similar to that for the DLB. The only difference is that, in the DLB, the flow control messages for a queue are sent to one end node, while, in the MLB, the flow control messages for a queue must be sent to r end nodes. From a similar analysis, we find that, in the worst case, we can prevent packet drop if one queue can hold at least $2*r+High\_Threshold_{MLB}$ packets. The MLB must be able to hold at least $N*(2+High\_Threshold_{MLB}/r)$ packets. In simulations, $High\_Threshold_{MLB}$ can be set to $16*r$ and $Low\_Threshold_{MLB}$ can be set to $8*r$, whereas the high threshold for an output is set to $6*r$ and the low threshold for an output is set to $3*r$.

D. Comparison of Different Loopback Buffers

Table I presents a detailed comparison of the three different loopback buffers. The SLB requires the most memory I/O bandwidth, while it occupies only one additional AWGR input. On the other hand, the DLB requires the least memory I/O bandwidth, but it requires more tunable transmitters, and the size of the AWGR must be doubled to support the DLB. While the SLB and the DLB represent the two extremes, the MLB provides a tradeoff between the SLB and the DLB, which is more attractive when we consider the complexity of both electronic devices and optical devices.

E. Performance Evaluation and Comparison for Different Loopback Buffers

All simulations in this section conform to the following configurations. The end node sends packets at a data rate of

TABLE I
COMPARISON OF DIFFERENT LOOPBACK BUFFERS

	SLB	MLB	DLB
Occupied AWGR ports	1	N/r	N
Number of receivers	N	N (r for each queue)	N (1 for each queue)
Number of transmitters	N (fixed)	N (tunable, r for each queue)	$N*m$ (tunable, m for each queue)
Number of optical DEMUXes	One $1:N$ optical DEMUX	One $1:N$ optical DEMUX	None
Number of optical MUXes	One $N:1$ optical MUX	None	None
Number of couplers	None	$(r-1)*(N/r)$	$N*(m-1)$
Memory write/read bandwidth	$N*B$ (read) $N*B$ (write)	$r*B$ (read) $r*B$ (write)	$m*B$ (read) B (write)
Number of memories	1	N/r	N

Notes.
N: the number of end nodes; k: the number of receivers at each OCA RX; m: the number of transmitters at each queue in DLB; r: the number of inputs each queue serves in MLB; B: the data rate at each wavelength.

10 Gbps. The OCA TX transmits the optical label at 2 Gbps. The optical label is 5 bytes in length. The OCA RX output rate is modeled as $(10*k)$ Gbps. The control plane clock speed is 2 GHz, and it requires 40 cycles to receive the entire 5-byte label. We assume that the end node uses InfiniBand [5] to connect with the optical hybrid switch and that the InfiniBand header is 66 bytes, excluding the start of frame (SOF) and the end of frame (EOF). The simulation results in [2] show that the contention probability is insensitive to the changes in the switch size and the packet length. The latency varies slightly when the size of the switch varies. Similar trends can be seen when the packet length varies. Simulation results in [2] also show that the system performance can be greatly improved when k increases from 2 to 4. Although we can further improve the performance by increasing k, the improvement will not be that significant. Since we have limited space, we fix the message size to 256 bytes, the value of k to 4, and the number of end nodes connected to 128 for all simulations in this paper.

The control plane latency, which is the sum of the latencies for label processing, arbitration, and TWC tuning, is compensated by the FDL between the LE and the TWC. Therefore, on the data plane, the switching latency includes only the transit latencies of the FDL, the TWC, and the AWGR for non-contended packets. For contended packets, in addition to the latencies mentioned above, the switching latency includes the time that the packet dwells at the loopback buffer and a second AWGR transit latency. In asynchronous switching, the packet transmission takes multiple cycles; thus, the transmission for the current packet can be pipelined with the arbitration for the next packet. Although the arbitration for contended packets in the DLB and the MLB takes more cycles than does the arbitration in the SLB, this increment will not affect the performance on the data plane due to the use of the pipelining technique. For the SLB, packets sent from one transmitter all go to the same switch output; thus, a fixed wavelength laser is used, and a guard time between two packets is not necessary. However, for the DLB and the MLB, because the packets sent from one transmitter may go to different switch outputs, the guard time is necessary in order to tune the tunable laser. The label processing, arbitration, TWC tuning, TWC transit, and AWGR transit latencies are set to 40, 3, 16, 1, and 10 cycles, respectively, for all simulations. Figures 7 to 11 show comparisons of the performances of the optical hybrid switches adopting the different loopback buffers discussed in this section. We conduct simulations under both uniform random traffic and hotspot traffic. For hotspot traffic, the hotspot parameter α is set to 4%, meaning that 4% of the entire traffic tries to reach the particular hotspot destination, while the remaining traffic goes to other destinations under uniform random distribution. The reason for setting the hotspot parameter to 4% is that the switch performance both before saturation and after saturation can be clearly observed as the load increases. If the hotspot parameter is small, the hotspot port may not saturate even at a high input load; if the hotspot parameter is large, the hotspot port may start saturating at a very low input load.

Figure 7 compares the average end-to-end latencies. We exclude the propagation delay that is determined solely by the distance between the switch and end nodes. Under uniform random traffic (Fig. 7(a)), the average end-to-end latency of the switch using the SLB is larger than that of the switch using the DLB or the MLB, because the SLB can use only one wavelength to send delayed packets to an output. The performance of the switch using the DLB with one transmitter per queue is slightly worse than those of the switches using the DLB with multiple transmitters per queue. The latencies are almost the same for the switches using the DLB with 2 or 4 transmitters per queue and for the switches using the MLB with different r values. We show the latency of the ideal output queue as a reference. The ideal output queue can be realized when each OCA RX has N receivers and a $1:N$ optical DEMUX.

Figures 7(b) and 7(c) compare the average end-to-end latencies under hotspot traffic. Figure 7(b) shows the average latencies for the hotspot destination; Fig. 7(c) shows the average latencies for non-hotspot destinations. The average end-to-end latencies for packets going to the hotspot destination increase significantly as the load increases. Since the DLB and the MLB can send out multiple delayed packets to the same output on different wavelengths, the switches using the DLB or the MLB saturate at a higher input load than does the switch adopting the SLB. On the other hand, with the proposed flow control, the average end-to-end latencies for packets going to non-hotspot destinations are kept low, as are the latencies under uniform random traffic, except for the switch using the DLB with one transmitter per queue. The reason is that the packets going to the hotspot port occupy only one transmitter at a queue, and other transmitters are still available for sending the delayed packets to non-hotspot destinations if one queue has multiple transmitters. For the switch using the DLB with one transmitter per queue, due to the HOL blocking, the average end-to-end latency increases

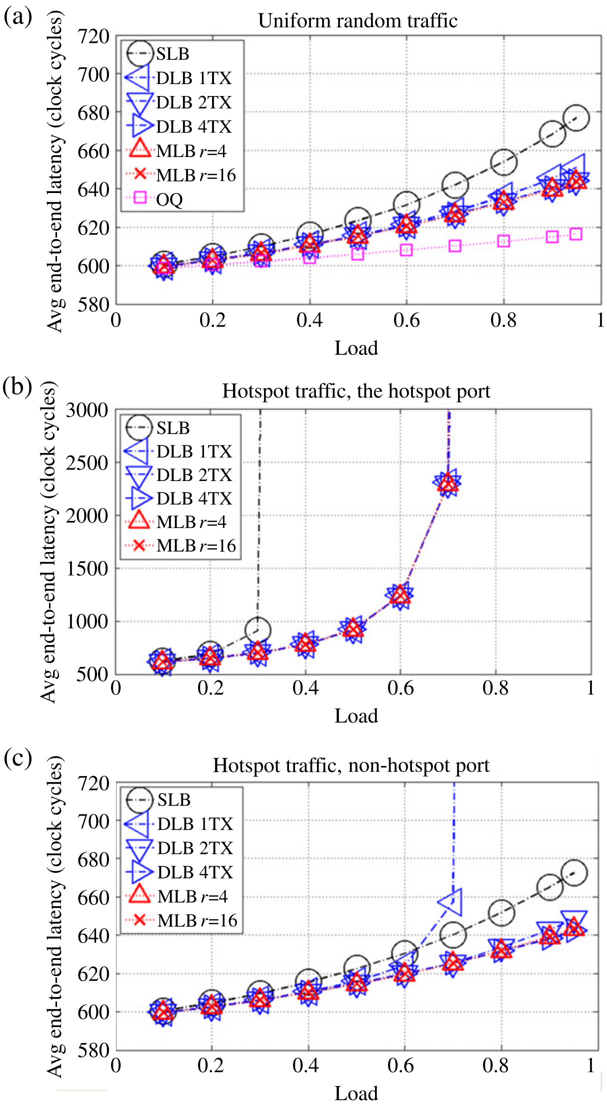

Fig. 7. (Color online) The end-to-end latency comparison of different loopback buffers: (a) under uniform random traffic, OQ is the ideal output queue, $k = N$; (b) the hotpot port under hotspot traffic; and (c) non-hotspot ports under hotspot traffic.

dramatically for non-hotspot destinations after the hotspot port saturates.

Figure 8 shows the injected traffic and the throughput comparison for different loopback buffers under uniform random traffic and hotspot traffic. The injected traffic reflects the actual load injected into the network when the flow control takes effect. The throughput is the same as the injected traffic for all cases, confirming that no packet is dropped in the switch. Under uniform random traffic, the throughput for all switches adopting different loopback buffers increases as the load increases, since no port saturates. Under hotspot traffic, the throughput for non-congested ports still increases linearly as the load increases for switches using different loopback buffers, except for the DLB with one transmitter per queue. Both the DLB and the MLB provide higher throughput for the hotspot destination than does the SLB, as more wavelengths can be used to deliver the delayed packets for the hotspot port.

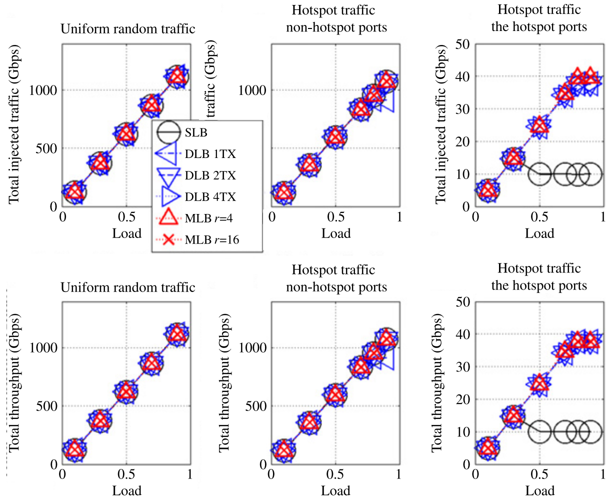

Fig. 8. (Color online) The injected traffic and the throughput comparisons for different loopback buffers under uniform random traffic and hotspot traffic.

The results confirm that the performance of non-congested ports is not influenced by the congestion on the hotspot port under hotspot traffic, except for the switch using the DLB with one transmitter per queue.

In addition to contention probability, we are interested in determining the time that delayed packets stay in the loopback buffer. We can break down the total contention percentage into three parts: the percentage of packets for which the transmission can start immediately upon the head of the packet arriving at the loopback buffer, the percentage of packets that cannot be immediately transmitted but the transmission of which can start before the entire packet is received, and the percentage of packets for which the transmission cannot start until the entire packet is received. We assume that D_0 is the latency experienced by delayed packets for which the transmission can start immediately upon the head of the packet arriving at the loopback buffer. Figure 9 shows the percentages of the three parts for different loopback buffers under uniform random traffic. Because of wavelength parallelism, the transmission for most delayed packets can start before their last bits arrive at the loopback buffer. The results also show that, compared with the SLB, in the DLB and the MLB there are more delayed packets for which the transmission can start immediately upon the head of the packet arriving at the loopback buffer. Furthermore, compared with the SLB, there are fewer delayed packets in the DLB and the MLB for which the transmission cannot start before the entire packet is received. The improvement is primarily because the DLB and the MLB can use multiple wavelengths to drain out delayed packets concurrently for the same output.

Figure 10 shows the contention probability breakdown for different loopback buffers under hotspot traffic. Figure 10(a) shows the breakdown for the delayed packets going to the hotspot port, and Fig. 10(b) shows the breakdown for the delayed packets going to non-hotspot ports. For the hotspot port, the percentage of delayed packets for which the transmission can start before their last bit arrives drops dramatically as the load increases. In addition, most delayed

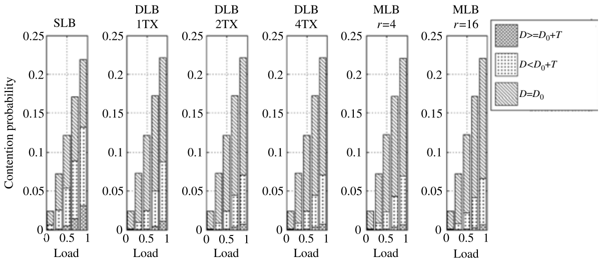

Fig. 9. The buffer latency distribution comparison for different loopback buffers under uniform random traffic.

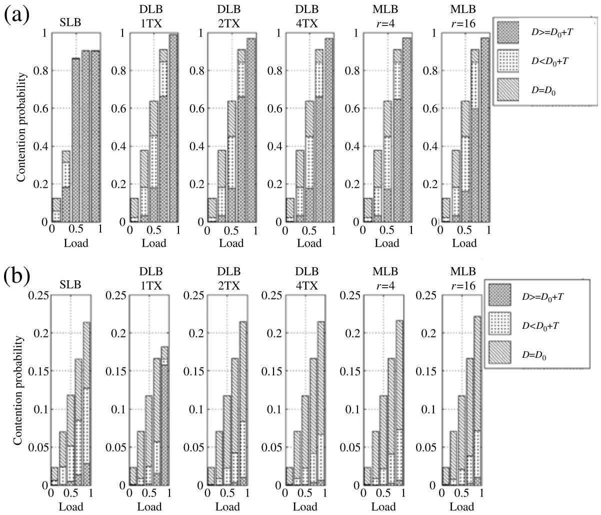

Fig. 10. The contention probability breakdown comparison for different loopback buffers under hotspot traffic: (a) the hotspot port; (b) non-hotspot ports.

packets going to the hotspot destination undergo a large delay after the hotspot port saturates. The switch using the SLB saturates at a lower input load than does the switch using the DLB or the MLB, because the SLB can use only one wavelength to send out delayed packets for the hotspot port. Packets going to the hotspot port undergo a higher contention rate in the switch using the DLB or the MLB than they do in the switch adopting the SLB. The reason is that the DLB and the MLB may occupy all wavegroups almost all the time after the hotspot port saturates, and almost all fresh incoming packets are directed to the loopback buffer. For non-hotspot ports, the transmission for most delayed packets can still start before receiving the entire packet. For the switch using DLB with one transmitter per queue, most delayed packets going to non-congested ports undergo a large delay after the hotspot port saturates due to the HOL blocking; the contention probability for non-congested ports becomes flat at the same time because the flow control limits the amount of traffic injected into the network.

Figure 11 shows the occupancies of different loopback buffers. The 99% buffer size is measured as a threshold, where 99% of the time the number of packets stored in the buffer is smaller than this threshold. The buffer size is measured at the time that a packet arrives at the loopback buffer. Figure 11(a) shows the occupancies of different loopback buffers under uniform random traffic. The results shown in Fig. 11(a) indicate that the optical hybrid switch actually requires a small buffer space, even when the network load is heavy, and that most of the time the average number of packets stored for each destination is less than 1 under uniform random traffic. This again confirms that the transmission for most delayed packets can start before their last bits are received. It is no wonder that the SLB must store a few more packets compared to the DLB and the MLB, since it can use only one wavelength to send out delayed packets for one output. Figure 11(b) shows the occupancies of different loopback buffers under hotspot traffic. The buffer size becomes almost flat after saturation, implying that most delayed packets try to reach the same output. As the SLB can use only one wavelength to drain out delayed packets, the buffer size of the SLB increases more quickly and saturates earlier than do the DLB and the MLB as the load increases. The DLB with one transmitter for each queue stores more delayed packets than do other loopback buffers after saturation, due to the HOL blocking.

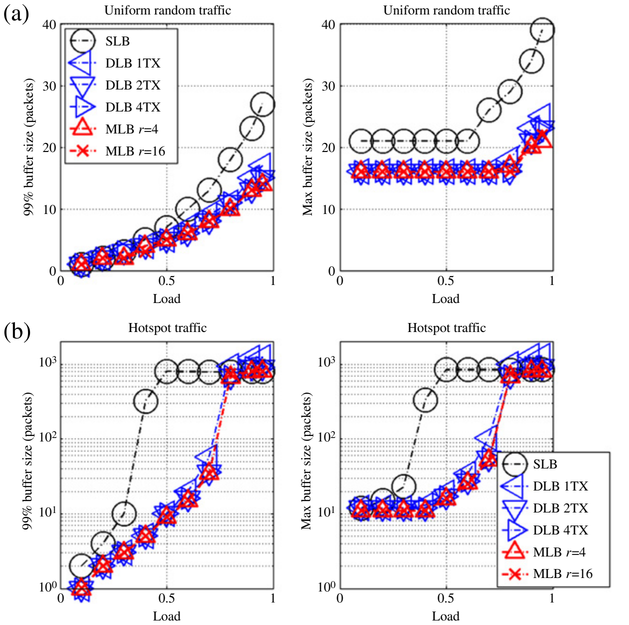

Fig. 11. (Color online) The occupancy comparison for the different loopback buffer structures: (a) under uniform random traffic; (b) under hotspot traffic.

The results shown in Figs. 7 to 11 validate the effectiveness of the proposed flow control scheme. The congestion occurring on a particular port will not affect the other ports, except for the DLB with one transmitter per queue. Again, as one input can generate at most one delayed packet for the loopback buffer, congestion may occur in only a few output ports, not all of the ports throughout the switch.

In summary, the optical hybrid switch adopting the proposed DLB and MLB can provide better performance compared with the switch using the SLB, as the DLB and the MLB use more AWGR ports for transmitting delayed packets. Therefore, delayed packets can be sent out on different wavelengths, even for the same destination, and the delayed packets do not make requests to the same contention group when they make those requests as fresh incoming packets. In addition to the gain in performance, the DLB and the MLB require much less memory I/O bandwidth than does the SLB. Although the MLB and the DLB with multiple transmitters per queue perform similarly, the MLB occupies fewer AWGR ports and requires fewer tunable transmitters, thus making it more attractive when we consider performance, scalability, cost, and complexity.

V. Performance Evaluation With Two-Level Flow Control

As discussed above, we want to eliminate the assumption that the OCA RX output rate is the same as the OCA RX aggregated input rate. Therefore, flow control between the OCA RX and the switch control plane is essential in resolving the rate mismatching that occurs when the OCA RX output rate is smaller than the OCA RX aggregated input rate. Therefore, the optical hybrid switch must adopt a two-level flow control scheme to prevent packet drop, as shown in Fig. 12: the loopback buffer flow control that has been discussed in Section IV, and the OCA flow control that will be discussed in this section. We will also discuss the interaction between the two flow controls.

The OCA flow control is built upon on–off flow control. When the number of packets that have accumulated at the OCA RX of one output exceeds the pre-defined high threshold, a flow control message will be sent to the switch control plane to ask for the suspension of all transmissions from the switch to the output, except for the loopback buffer flow control messages. When this happens, the switch directs all incoming packets destined for that output to the loopback buffer. The transmissions can be resumed when the buffer size drops below the low threshold. Since the OCA RX and the OCA TX are placed adjacent to each other, it is not difficult to pass the flow control message from the OCA RX to the OCA TX. The OCA TX can then send the flow control information on the wavelength used for transmitting optical labels, so that the flow control information can be received and processed at the control plane. We can use one bit in the optical label to denote the flow control status. In the period when the OCA flow control is ON, the flow control bit at all optical labels will be marked to ‘1’; otherwise, it will be marked to ‘0.’ Between optical labels, we can use two types of common symbols to indicate whether the OCA flow control is ON or OFF. Therefore, we can assume that the total time for the flow control information being sent, received, and processed will be less than the time required to transmit one normal packet. Since one OCA RX may receive k packets concurrently, the buffer at one OCA RX must be able to store $k + High\_Threshold_{OCA\_RX}$ packets to prevent buffer overflow. We set $High\_Threshold_{OCA\_RX}$ to N and $Low\_Threshold_{OCA\_RX}$ to $N/2$ in simulations.

When the OCA flow control is triggered, the loopback buffer can still send flow control messages if necessary. To prevent the OCA RX from dropping flow control messages due to buffer overflow, we can allocate a dedicated space at each OCA RX to store the loopback buffer flow control messages. The loopback buffer flow control messages for one particular end node will arrive at the OCA RX in sequence on the same wavelength, no matter which type of loopback buffer is adopted. The loopback buffer flow control message received at one OCA RX will be inserted in the head of the queue, so that it can be sent to the end node as soon as possible. Since the OCA RX output rate may be smaller than the OCA RX aggregated input rate, the loopback buffer flow control message may also undergo some latency at the OCA RX. If the old flow control message is still waiting for transmission at the head of the queue when a new one is received, we can replace the old message with the new one so that only one flow control message carrying the latest information waits at the head of the queue. The largest delay that a flow control message may undergo at the OCA RX is equal to the packet transmission latency. To accommodate this extra latency, the required buffer size for the SLB, and each memory unit for the DLB and the MLB, must increase to $3*N+High\_Threshold_{SLB}$, $3+High\_Threshold_{DLB}$, and $3*r+High\_Threshold_{MLB}$ packets, respectively.

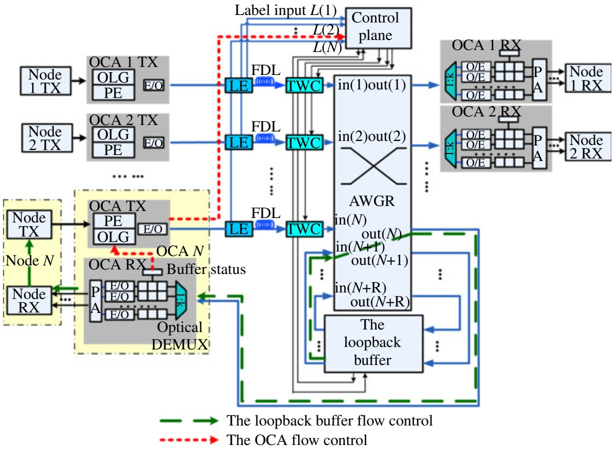

Fig. 12. (Color online) The proposed two-level flow control for the optical hybrid switch. OLG: optical label generator; PE: packet encapsulation; LE: label extractor; FDL: fiber delay line; PA: packet aggregation; O/E: optical-to-electrical converter; E/O: electrical-to-optical converter; TX: transmitter; RX: receiver; $L(i)$: label from node i.

In summary, the optical hybrid switch will utilize a two-level flow control scheme. One is the OCA flow control, which resolves the OCA RX input and output rate mismatching and prevents buffer overflow at the OCA RX. The other is the loopback buffer flow control, which controls the amount of traffic injected into the network to ensure that no packet will be dropped if it has been in the network. The triggering of the OCA flow control increases the probability of the loopback buffer flow control being triggered, but not vice versa. If the OCA flow control is ON for a while, the packet will accumulate in the loopback buffer, and the loopback buffer flow control will be triggered later. The loopback buffer flow control limits the number of packets injected into the network, thus alleviating, and not exacerbating, the congestion at the OCA RX.

The simulations conducted in this section use the same configurations as those in Section IV, except for the OCA RX output data rate and the hotspot parameter. In this section, we assume that the output data rate for the OCA RX is the same as the data rate on each wavelength, so that the input and output rate ratio for the OCA RX is k. We change the hotspot parameter from 0.04 to 0.02 so that the hotspot port will not saturate at a very low input load. Because the simulations

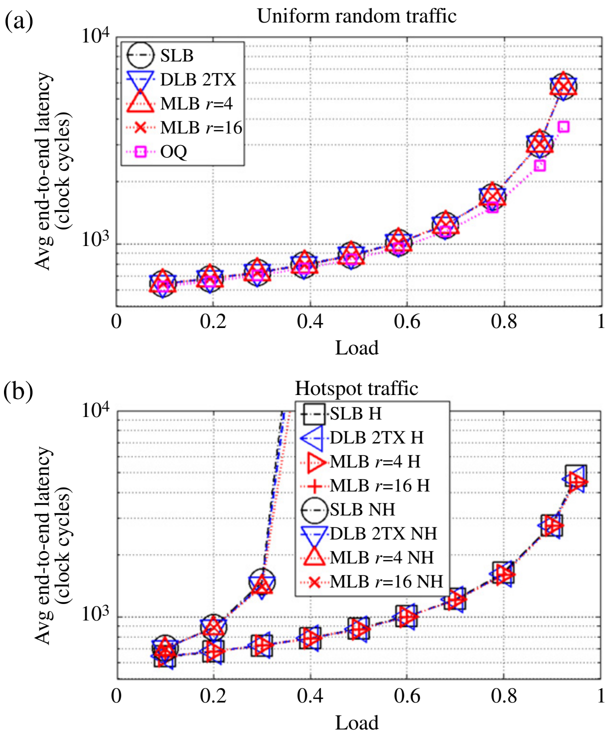

Fig. 13. (Color online) The end-to-end latency comparison: (a) under uniform random traffic, OQ is the ideal output queue, $k = N$; (b) under hotspot traffic.

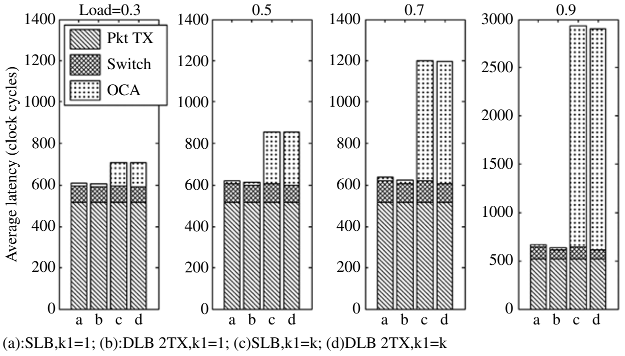

Fig. 14. The end-to-end latency breakdown comparison of the SLB and the DLB for different OCA RX output rates under uniform random traffic. Pkt TX: packet transmission latency; Switch: the delay at the switch, including delay at the loopback buffer; OCA: the delay at the OCA TX and OCA RX, a: SLB, $k1 = 1$; b: DLB 2TX, $k1 = 1$; c: SLB, $k1 = k$; d: DLB 2TX, $k1 = k$.

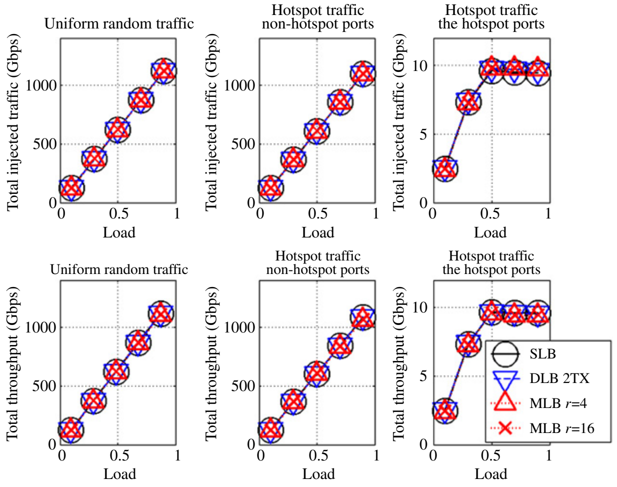

Fig. 15. (Color online) The injected traffic and the throughput comparisons under uniform random traffic and hotspot traffic.

in Section IV show that the performance difference between the DLB with 2 transmitters per queue and the DLB with 4 transmitters per queue is small, we will show only the performance of the DLB with 2 transmitters per queue for the simulations in this section.

Figure 13 shows a comparison of the average end-to-end latencies. We exclude the propagation delay that is determined solely by the distance between the switch and end nodes. Figure 13(a) shows a comparison of the latencies under uniform random traffic. There is little difference among the results for the switches using different loopback buffers, even at high input load. The switches using different loopback buffers all perform slightly worse than does the switch with the ideal output queue. Figure 13(b) shows the latencies under hotspot traffic for the hotspot destination and non-hotspot destinations, respectively. Since the end node now has a reduced capability for consuming data, the hotspot port begins to saturate at a light input load. Again, the latencies at non-hotspot ports are similar to those under uniform random traffic.

Figure 14 shows the average end-to-end latency breakdown for the switch with different OCA output data rates for the SLB and the DLB under uniform random traffic. The parameter $k1$ denotes the input and output rate ratio for the OCA RX. When the output rate of the OCA RX is smaller than the OCA RX aggregated input rate, packets will undergo more latency at the OCA RX, thus significantly increasing the end-to-end latency; the latency at the OCA RX increases more quickly than does the switching latency as the load increases.

Figure 15 shows the injected traffic and the throughput comparison for optical hybrid switches adopting different loopback buffers under uniform random traffic and hotspot traffic. All configurations perform similarly under different traffic. Under uniform random traffic, the throughput increases as the load increases since no port saturates. Under hotspot traffic, the throughput for non-hotspot ports increases as the load increases, and it is not influenced by the congestion occurring at the hotspot port. Again, the throughput is the same as the injected traffic for all cases, confirming that no packet is dropped in the switch. In contrast to Fig. 8, the throughput of the hotspot port is the same for the switches using different loopback buffers, and it becomes flat after the hotspot port saturates. The difference between the SLB and other loopback buffers vanishes due to the OCA RX output rate reduction.

Figure 16 shows the contention probability breakdown comparison for delayed packets under uniform random traffic. The transmission for most delayed packets starts before their last bits arrive. Although Fig. 16 shows trends similar to those

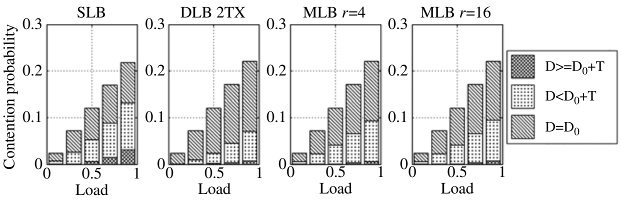

Fig. 16. The contention probability breakdown for different loopback buffers under uniform random traffic.

in Fig. 9, unlike Fig. 7(a), Fig. 13(a) shows that the switches using different loopback buffers all perform similarly. This similarity indicates that, when the OCA RX output rate is the same as that on each wavelength, the contention at the OCA RX, which cannot be alleviated by output queuing, becomes a greater factor than the contention at the switch in limiting the switch performance.

Figure 17 shows a comparison of the contention probability breakdown for the delayed packets under hotspot traffic. For non-hotspot ports, the transmission for most delayed packets can start before their last bits arrive at the buffer. The DLB and the MLB allow more delayed packets to be transmitted before the entire packet is received than does the SLB. Since packets spend much more time at the OCA RX, the total latencies for the switches using different loopback buffers are similar, as shown in Fig. 13(b). For the hotspot port, most delayed packets undergo large delays after the hotspot port saturates. The loopback buffer flow control guarantees that the performance of non-hotspot ports will not be affected by congestion occurring at the hotspot destination.

Figure 18(a) shows the occupancies of different loopback buffers under uniform random traffic. Despite the output rate reduction at the OCA RX, the number of delayed packets stored at the loopback buffer still increases slowly as the load increases, and, most of the time, the average number of packets stored for each destination is less than 1 under uniform random traffic, which is similar to that shown in Fig. 11(a). Figure 18(b) shows the occupancies of different loopback buffers under hotspot traffic. The buffer size becomes almost flat after the hotspot port saturates, implying that most delayed packets have the same destination address.

Figure 19 shows the trigger frequencies of the OCA flow control and the loopback buffer flow control under hotspot traffic. The OCA flow control can be easily triggered when the hotspot output saturates. The triggering frequency of the loopback buffer flow control is much lower than that of the OCA flow control (which is desirable since the loopback buffer flow control has a more significant impact on system performance), because the loopback buffer flow control limits the actual traffic injection rate for the system. The results also show that the switch using the MLB triggers more flow control than does the switch using the SLB and that the switch using the DLB triggers even more flow control than does the switch using the MLB, thus indicating that the more distributed the loopback buffer is, the more flow control the loopback buffer requires.

In summary, the simulations shown in this section confirm that the OCA flow control and the loopback buffer flow control work as expected, ensuring that no packet drop occurs due to contention and buffer overflow, resolving data rate mismatching due to output queuing, and insulating the congestion.

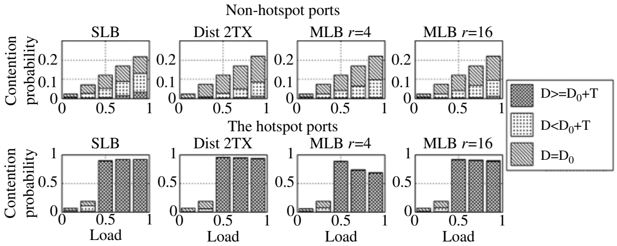

Fig. 17. The contention probability breakdown for different loopback buffers under hotspot traffic.

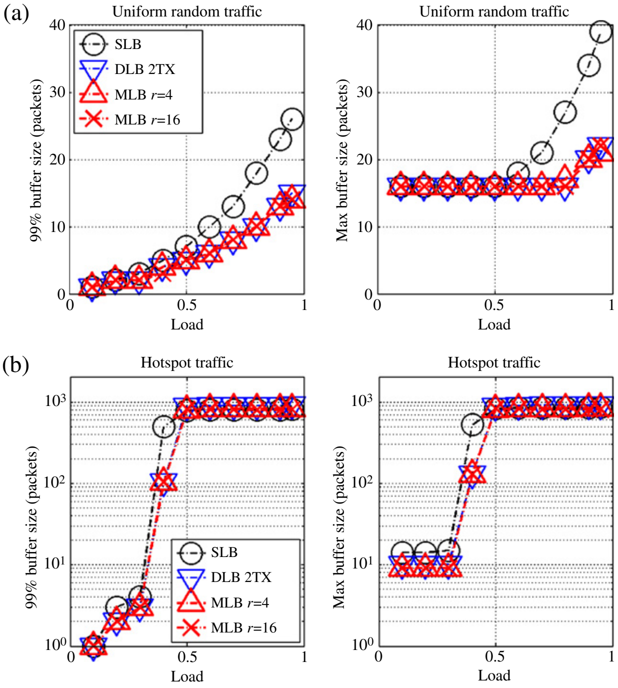

Fig. 18. (Color online) The occupancy comparison of different loopback buffers: (a) under uniform random traffic; (b) under hotspot traffic.

VI. Conclusion

In this paper, we have proposed two loopback buffer architectures for an optical switch. We described the advantages and disadvantages of different loopback buffer architectures and compared their performance through simulation. Given the performance, the scalability, the cost, and the complexity of the required electronic and optical devices, the MLB appears to be a better choice for the optical switch than are the DLB and the SLB. We also proposed a two-level flow control scheme. The loopback buffer flow control can eliminate packet drop at the switch and prevent the congestion at the hotspot port from affecting the performance of the non-hotspot ports. The OCA flow control prevents potential packet drop due to the data rate mismatching that is caused by output queuing. The simulation

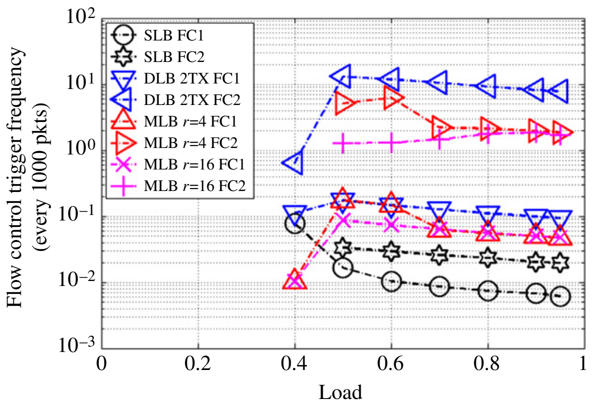

Fig. 19. (Color online) Flow control trigger frequencies under hotspot traffic (FC1: the OCA flow control; FC2: the loopback buffer flow control).

results validate the effectiveness of the proposed two-level flow control scheme.

Acknowledgment

This work was supported in part by the U.S. Department of Defense under contract #H88230-08-C-0202.

References

[1] L. A. Barroso and U. Hölzle, "Introduction," in *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Morgan & Claypool, 2009, pp. 1–11.

[2] X. Ye, P. Mejia, Y. Yin, R. Proietti, S. J. B. Yoo, and V. Akella, "DOS—A scalable optical switch for datacenters," in *Proc. ACM/IEEE Symp. Architectures for Networking and Communications Systems*, 2010, pp. 1–12.

[3] "Data center architecture overview," in *Cisco Data Center Infrastructure 2.5 Design Guide*. Cisco, 2007, pp. 7–16 [Online]. Available: http://www.cisco.com/univercd/cc/td/doc/solution/dcidg21.pdf.

[4] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proc. ACM SIGCOMM 2008 Conf. Data Communication*, 2008, pp. 63–74.

[5] "Architectural overview," in *InfiniBand Architecture Specification Volume 1, Release 1.0*. 2001, pp. 61–63 [Online]. Available: http://www.infinibandta.org/specs.

[6] N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat, "Helios: A hybrid electrical/optical switch architecture for modular data centers," in *Proc. ACM SIGCOMM 2010 Conf. Data Communication*, 2010, pp. 339–350.

[7] G. Wang, D. G. Andersen, M. Kaminsky, K. Papagiannaki, T. S. E. Ng, M. Kozuch, and M. Ryan, "c-Through: Part-time optics in data centers," in *Proc. ACM SIGCOMM 2010 Conf. Data Communication*, 2010, pp. 327–338.

[8] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable dragonfly topology," in *Proc. 35th Int. Symp. Computer Architecture*, 2008, pp. 77–88.

[9] B. Webb and A. Louri, "A class of highly scalable optical crossbar-connected interconnection networks (SOCNs) for parallel computing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 11, no. 5, pp. 444–458, 2000.

[10] A. Louri and A. Kodi, "An optical interconnection network and a modified snooping protocol for the design of large-scale symmetric multiprocessors (SMPs)," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 11, pp. 1093–1104, 2004.

[11] R. Chamberlain, M. Franklin, and C. Baw, "Gemini: An optical interconnection network for parallel processing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 13, no. 10, pp. 1038–1055, 2002.

[12] C. Minkenberg, F. Abel, P. Muller, R. Krishnamurthy, M. Gusat, P. Dill, I. Iliadis, R. Luijten, R. R. Hemenway, R. Grzybowski, and E. Schiattarella, "Designing a crossbar scheduler for HPC applications," *IEEE Micro*, vol. 26, no. 3, pp. 58–71, 2006.

[13] R. Hemenway, R. R. Grzybowski, C. Minkenberg, and R. Luijten, "Optical-packet-switched interconnect for supercomputer applications," *J. Optical Netw.*, vol. 3, pp. 900–913, 2004.

[14] C. Hawkins, B. A. Small, D. S. Wills, and K. Bergman, "The data vortex, an all optical path multicomputer interconnection network," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 3, pp. 409–420, 2007.

[15] O. Liboiron-Ladouceur, A. Shacham, B. A. Small, B. G. Lee, H. Wang, C. P. Lai, A. Biberman, and K. Bergman, "The data vortex optical packet switched interconnection network," *J. Lightwave Technol.*, vol. 26, no. 13, pp. 1777–1789, 2008.

[16] K. Bergman, D. Keezer, and S. Wills, "Design, demonstration and evaluation of an all optical processor memory-interconnection network for petaflop supercomputing," in *ACS Interconnects Workshop*, 2010, p. 16 [Online]. Available: http://lightwave.ee.columbia.edu/?s=research\&p=high-performance_computing_systems#dv.

[17] H. Yang and S. J. B. Yoo, "Combined input and output all-optical variable buffered switch architecture for future optical routers," *IEEE Photon. Technol. Lett.*, vol. 17, pp. 1292–1294, 2005.

[18] S. J. B. Yoo, "Optical packet and burst switching technologies for the future photonic Internet," *J. Lightwave Technol.*, vol. 24, pp. 4468–4492, 2006.

[19] S. Bregni, A. Pattavina, and G. Vegetti, "Architectures and performance of AWG-based optical switching nodes for IP networks," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 7, pp. 1113–1121, 2003.

[20] M. Maier, M. Scheutzow, and M. Reisslein, "The arrayed-waveguide grating-based single-hop WDM network: An architecture for efficient multicasting," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 9, pp. 1414–1432, 2003.

[21] W. D. Zhong and R. S. Tucker, "Wavelength routing-based photonic packet buffers and their applications in photonic packet switching systems," *J. Lightwave Technol.*, vol. 16, no. 10, pp. 1737–1745, 1998.

[22] D. Banerjee, J. Frank, and B. Mukherjee, "Passive optical network architecture based on waveguide grating routers," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 7, pp. 1040–1050, 1998.

[23] M. C. Chia, D. K. Hunter, I. Andonovic, P. Ball, I. Wright, S. P. Ferguson, K. M. Guild, and M. J. O'Mahony, "Packet loss and delay performance of feedback and feed-forward arrayed-waveguide gratings-based optical packet switches with WDM inputs-outputs," *J. Lightwave Technol.*, vol. 19, no. 9, pp. 1241–1254, 2011.

[24] Z. Zhang and Y. Yang, "Performance analysis of optical packet switches enhanced with electronic buffering," in *Proc. IEEE Int. Symp. Parallel & Distributed Processing*, 2009, pp. 1–9.

[25] Z. Guo, Z. Zhang, and Y. Yang, "Performance modeling of hybrid optical packet switches with shared buffer," in *Proc. 30th IEEE Int. Conf. Computer Communications*, 2011, pp. 1530–1538.

[26] L. Liu, Z. Zhang, and Y. Yang, "Packet scheduling in a low-latency optical switch with wavelength division multiplexing and electronic buffer," in *Proc. 30th IEEE Int. Conf. Computer Communications*, 2011, pp. 1692–1700.

[27] S. J. B. Yoo, "Wavelength conversion technologies for WDM network applications," *J. Lightwave Technol.*, vol. 14, no. 6, pp. 955–966, 1996.