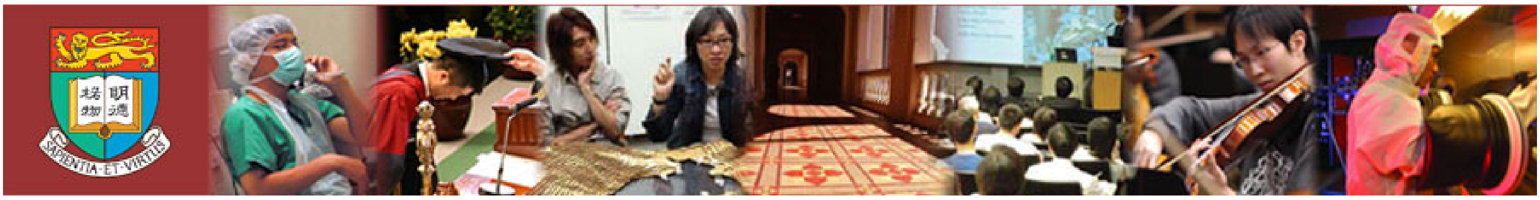

Load-Balanced Optical Switch for High-Speed Router Design

Bing Hu, *Member, IEEE*, and Kwan L. Yeung, *Senior Member, IEEE*

***Abstract*—A hybrid electro-optic router is attractive, where packet buffering and table lookup are carried out in electrical domain and switching is done optically. In this paper, we propose a load-balanced optical switch (LBOS) fabric for a hybrid router. LBOS comprises N linecards connected by an N-wavelength WDM fiber ring. Each linecard i is configured to receive on channel λ_i. To send a packet, it can select and transmit on an idle channel based on where the packet goes. The packet remains in the optical domain all the way from input linecard to output linecard. Meanwhile, the loading in the ring network is perfectly balanced by spreading the packets for different destinations to use different wavelengths, and the packets for the same destination to use different time slots. With the pipelined operation of the LBOS, we show that LBOS can yield close-to-100% throughput performance. To address the ring-fairness problem under the inadmissible traffic patterns, an efficient throughput-fair scheduler is devised. To efficiently support multicast traffic, a simple multicast scheduler is also proposed. Finally, the linecard placement problem is investigated for further cutting down the average packet delay.**

***Index Terms*—Fiber ring network, high-speed router, load-balanced optical switch.**

I. Introduction

IMPLEMENTING an all-optical router is still far from being practical because of the immature technologies in optical processing and buffering. In this paper, we focus on designing hybrid electro-optic routers, where packet buffering and table lookup [1] are carried out in electrical domain, and switching is done optically [2], [3].

There are various efforts in designing an efficient optical switch. In [4], a hybrid electro-optic implementation of a load-balanced electronic switch [5] is reported, where optical MEMS is used as middle-stage switch modules in the three-stage Clos network architecture. Due to the hybrid nature, all-optical packet transmission from an input linecard to an output linecard is not possible. Recently, Fastnet [6], an optical switch fabric comprising N switch linecards connected by two counter-rotating WDM fiber rings, is proposed. The notion of counter-rotating WDM fiber rings originally appears in designing metro networks [7]–[12]. In Fasnet, one ring is used for transmission, while the other is for reception. A linecard is attached to both rings for proper sending and receiving. The two rings are connected via a folding point (e.g., a pre-determined linecard). Only a special input port (called master input) can generate a frame header (called locomotive). Other input ports are restricted to put their packets at the end of a frame only after a frame header passes by. An input linecard first sends a packet onto the transmission ring. Then the packet passes through the folding point to reach the reception ring. It travels along the reception ring until reaching the destination linecard. We can see that in an N -linecard Fasnet, a packet needs to travel between 1 to 2 N hops for reaching its destination linecard. The average distance between any pair of linecards is $(1+2N)/2$ hops. Although Fasnet allows all-optical packet transmission from one linecard to another, its delay-throughput performance is limited due to the large average distance between linecards.

Other interesting work on designing efficient ring networks can be found in [13]–[16]. In [13], a bidirectional WDM ring network called HORNET is proposed. In HORNET, a network node attempts to balance its traffic using the available bandwidth in both directions of the ring. A control-channel-based protocol enables the nodes to share the bandwidth of the network while preventing collisions. To ensure fairness among nodes, a fair scheduler is designed based on DQDB [14], and at the cost of lower system throughput. In [15], [16], an Optical-header Processing and Access Control System (OPACS) is proposed for time-slotted ring network, where optical packet header is time-division-multiplexed with the packet payload. The optical headers across all parallel wavelengths can be received, modified, and retransmitted by a wavelength—time conversion technique. Although OPACS provides good throughput and fairness performance, its implementation complexity is high.

In this paper, we focus on designing an optical switch that allows packets to be sent all-optically from one linecard to another. A new load-balanced optical switch (LBOS) is proposed, where N linecards are connected by an N -wavelength WDM fiber ring. Unlike Fasnet, the single fiber ring is used for both transmission and reception. Each linecard i is configured to receive on channel λ_i. To send a packet, it can select and transmit on an idle channel based on where the packet goes. Under admissible traffic patterns, LBOS provides close-to-100% throughput by evenly spreading packets for different destinations in both wavelength and time domains. To address the ring-fairness problem under inadmissible traffic patterns, a throughput-fair scheduler is proposed. To effectively carry multicast traffic, a simple multicast scheduler is designed. To further cut down average packet delay, a way to reconfigure the relative positions of linecards on the ring is also designed.

Manuscript received November 27, 2009; revised April 09, 2010; accepted April 19, 2010. Date of publication May 20, 2010; date of current version June 28, 2010. This work was supported in part by the Cisco Research Center, the Seed Funding Programme for Basic Research of The University of Hong Kong (No. 10400906), and National Science Fund for Distinguished Young Scholars (No. 60525110).

B. Hu is with the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China (e-mail: binghu@zju.edu.cn).

K. L. Yeung is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China (e-mail: kyeung@eee.hku.hk).

Digital Object Identifier 10.1109/JLT.2010.2050677

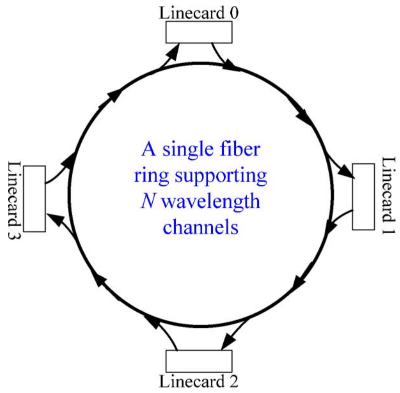

Fig. 1. A 4 × 4 load balanced optical switch.

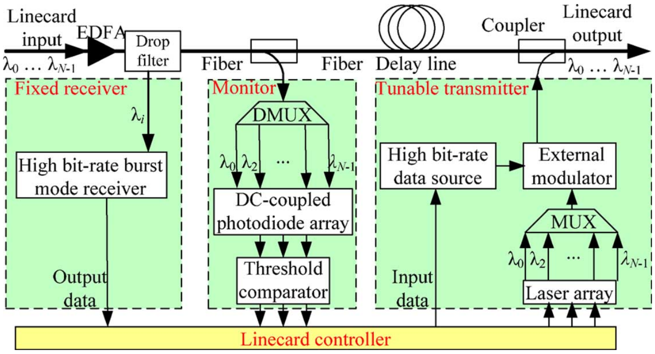

Fig. 2. Internal structure of linecard i.

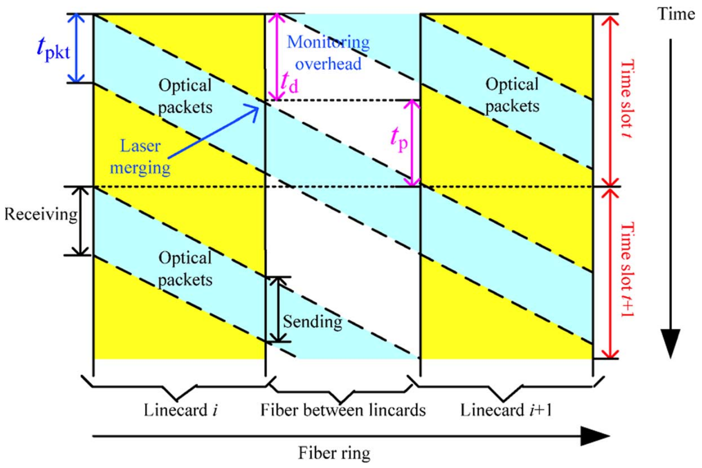

Fig. 3. Non-pipelined operation of LBOS.

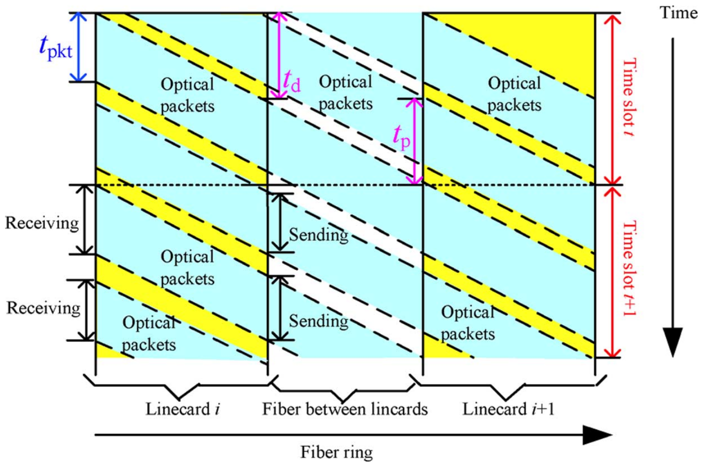

Fig. 4. Pipelined operation of LBOS.

The rest of the paper is organized as follows. In the next section, the design and operation of LBOS are detailed. In Section III, a throughout-fair scheduler is presented. In Section IV, we extend LBOS to support multicast traffic as well as linecard placement. Simulation results are presented in Section V and we conclude the paper in Section VI. In Appendix A, we further show that the LBOS is an optical counterpart of an efficient load-balanced electronic switch.

II. Load Balanced Optical Switch

A. Switch Architecure

LBOS is targeted at all-optical switching of a packet from one linecard to another. As depicted in Fig. 1, LBOS consists of N linecards connected by an N-wavelength WDM fiber ring. Each linecard i has two ports, input i and output i. Output i is configured to receive (only) on its dedicated wavelength channel λ_i. To send a packet to linecard j, input i needs to transmit the packet onto channel λ_j when λ_j is idle.

The internal structure of linecard i is similar to that used by Fasnet [6], and is shown in Fig. 2. For simplicity, the electrical virtual output queues (VOQs) at each input are not shown. A linecard i has three major modules: a receiver on channel λ_i, a "tunable" transmitter and a wavelength monitor. In Fig. 2, the EDFA (Erbium Doped optical Fiber Amplifier) is used to compensate for the optical signal loss en route. A filter drops wavelength λ_i from the fiber and passes all other channels to a splitter. The dropped λ_i enters the high bit-rate burst mode receiver for receiving. The splitter taps out a fraction of light and feeds it to the monitor module. The remaining signals in the fiber will go through a FDL (Fiber Delay Line) of $t_{\rm d}$ seconds, where $t_{\rm d}$ is the time required for the monitor to identify an idle channel and the transmitter to start sending a packet onto the identified channel.

The light entered the monitor module is demultiplexed into $N-1$ separate λ's for detection by the dc-coupled photodiode array. This is followed by a threshold comparator for identifying idle channels. Among all the idle channels, the linecard controller selects the longest (electrical) VOQ(i,j), and its head of line packet is sent using the transmitter module. The transmitter module consists of a fixed laser array, where laser λ_j is for sending packets destined to linecard j. (A single fast tunable laser can be used if that is more cost effective.) Finally, the transmitted packet is merged back to the fiber ring for going to the next linecard.

B. Switch Operations

Let the amount of time required to send a packet be $t_{\rm pkt}$ seconds. The duration of a time slot becomes $t_{\rm d}+t_{\rm p}$ seconds, where $t_{\rm d}$ is the propagation delay of the FDL in Fig. 2 and $t_{\rm p}$ is the propagation delay of the fiber connecting to the next linecard along the ring. Assume the whole system is synchronized, and in each time slot, at most one packet can be transmitted and/or received by each linecard. For the proper operation of the switch,

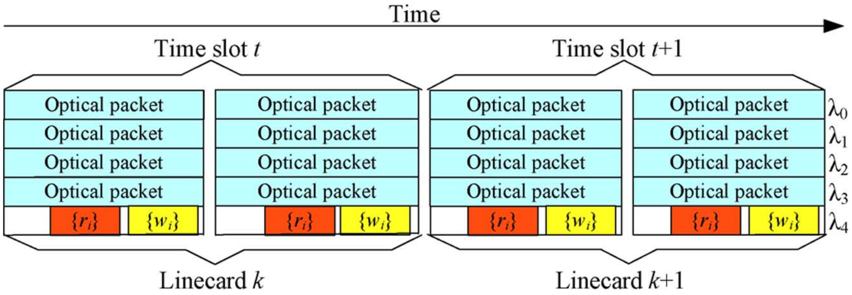

Fig. 5. Control channel λ_N, which carries $\{w_i\}$ and $\{r_i\}$.

we must have $t_{\mathrm{d}} \geq t_{\mathrm{pkt}}$ and $t_{\mathrm{p}} \geq t_{\mathrm{pkt}}$, as depicted in Fig. 3. Notably, linecard i starts to receive a packet at the beginning of slot t and it takes t_{pkt} seconds to receive the entire packet. Meanwhile, the monitor identifies all the idle channels, and a packet is sent onto the idle channel that has the longest VOQ size. The sent packet is added back to the ring at t_d seconds after the beginning of the current slot. It takes another t_{p} seconds for the first bit of the packet to arrive at linecard $i+1$. This marks the end of time slot t and the beginning of slot $t+1$.

From Fig. 3, we can see that a packet sent by linecard i will arrive at linecard j after $(j-i)$ mod N time slots. We can also see that in each time slot, the transmitter is idle in the first t_{d} seconds, whereas the receiver and monitor are idle for the last t_{p} seconds. As only a single packet is sent/received in each slot, the efficiency of (non-pipelined) LBOS is $t_{\mathrm{pkt}}/(t_{\mathrm{d}}+t_{\mathrm{p}})$, or at most 50% $(t_{\mathrm{p}}=t_{\mathrm{d}}=t_{\mathrm{pkt}})$. To avoid such under-utilization, the transmitter, receiver and monitor can be used for pipelined packet sending, receiving and scheduling, as shown in Fig. 4. Specifically, in the first half of a time slot, the transmitter can send a packet scheduled in the second half of the previous time slot. In the second half of a time slot, the receiver can receive an additional incoming packet, and the monitor can schedule another packet for sending in the first half of the next time slot. In other words, up to two packets can be received and transmitted in each time slot.

Indeed, with or without pipelined operations, LBOS can effectively balance the loading in the ring by spreading packets going to different destinations to use different wavelengths (i.e., space/wavelength domain load balancing), and packets going to the same destination to use different time slots (i.e., time domain load balancing). In Appendix A, we further show that LBOS is an optical counterpart of the load-balanced electronic switch architecture in [17].

III. A Max-Min Fair Scheduling Algorithm

For admissible traffic patterns, as long as the switch is stable, all packets can arrive at the outputs with bounded delays. In this case, fairness in throughput is not an issue. For an inadmissible traffic pattern with over-subscribed outputs, LBOS will suffer from the ring-fairness problem, i.e., an up-stream input can throttle a down-stream input while sending packets to the same over-subscribed output. A max-min fair scheduler (LBOS-F) is thus designed in this section.

A. LBOS-F

The basic idea is to carry out resource reservation for flows with VOQs exceeding a pre-determined threshold of T packets. To do so, an optical control channel (λ_N) is required for conveying reservation requests and grants. Accordingly, an extra low-speed (and thus inexpensive) transceiver on channel λ_N is required at each linecard. Assume pipelined LBOS is used. The operation of λ_N is illustrated in Fig. 5. In each packet duration, λ_N carries two vectors, an *overload vector* $\{w_i\}$ and a *reservation vector* $\{r_i\}(i=0,1\ldots N-1)$. When $w_i=l$, linecard i has more than or equal to T packets destined for linecard l. Otherwise $w_i=-1$. When $r_i=m$, channel λ_i (of the current packet duration) is reserved for linecard m (for sending a packet to linecard i). Otherwise, $r_i=-1$.

At each packet duration, linecard k drops λ_N, examines and updates the values of $\{w_i\}$ and $\{r_i\}$ based on the following rules:

- *Sending reservation/overload request*. For any linecard k, among its VOQs $\geq T$, select VOQ(k,l) based on a round robin (RR) scheduler and set the overload vector as $w_k=l$.
- *Determining the winner*. Linecard k examines the received $\{w_i\}$. If $w_i \neq k$ for all i's, set $r_k=-1$ in $\{r_i\}$ to indicate no reservation on λ_k. If there are some $w_i=k$, then using the RR scheduler to select $w_j=k$, and set $r_k=j$. This indicates that λ_k (of the current packet duration) is reserved by linecard j. Then update $w_i=k$ to $w_i=-1$ for all $i=0,1\ldots N-1$.
- *Sending a packet*. Linecard k examines the received reservation vector $\{r_i\}$. If there is any $r_i=m$, where $m>0$ and $m \neq k$, channel λ_i is not available (i.e., reserved by linecard m). If $r_i=k$, the head of line (HOL) packet from VOQ(k,i) is sent. Otherwise, send the HOL packet from the longest VOQ and with the corresponding idle channel.

Note that the delay between a linecard generating a request and knowing the result is N time slots (one round trip time on the ring network). With the pipelined LBOS, each linecard can send two packets per time slot. If the pre-determined threshold $T<2N$ packets, when a linecard knows its reservation is successful, the corresponding VOQ may already become empty because the backlogged packets have been exhausted while waiting for the result to arrive. This produces a wasted slot. If $T \geq 2N$, it is guaranteed that there will be *at least* one packet in the queue for making use of the reserved slot. However, having a large T would adversely affect the packet delay performance. Therefore, we set $T=2N$ in our LBOS-F.

B. Max-Min Fairness Criterion

In the following, we show that LBOS-F can satisfy the max-min fairness criterion. We first borrow two definitions from [18].

Definition 1: The allocation vector $\{a_i\}$ is said to be feasible if and only if:

- Each entity receives an allocation greater than or equal to zero; that is, for all $i, a_i \geq 0$.
- The total allocated resource is less or equal to the available resource U; that is, $\sum a_i \leq$ U.

Definition 2: For the demand vector $\{b_i\}$, the allocation vector $\{a_i\}$ is said to be max-min fair if:

1) It is feasible.
2) No entity receives an allocation greater than its demand; that is, for all $i, a_i \leq b_i$.
3) For all i, the allocation of entity i cannot be increased while satisfying the above two conditions and without reducing the allocation of some other entity j for which $a_j \leq a_i$.

As long as an algorithm meets the three conditions above, it satisfies the max-min fairness criterion. Note that in our LBOS-F algorithm, the demand vector $\{b_i\}$ is the traffic load from input i to an over-subscribed output j. Let the capacity of output j be U, i.e., the available resource is U. Assume LBOS-F allocates U to each input i with allocation $a_i(i = 0, 1 \ldots N-1)$. Obviously, $a_i \geq 0$ and $\sum a_i \leq \mathrm{U}(i = 0, 1 \ldots N-1)$. So a_i is *feasible* (condition 1). By setting the threshold for generating a reservation request at T =2 N, LBOS-F will not waste any reserved slot. So for all $i, a_i \leq b_i$ can be ensured (condition 2).

In the following, we focus on condition 3, where we increase some bandwidth allocation a_i and see how this would affect other inputs. Assume the switch has been "warmed up". Let c_i be the number of times that input i's $\mathrm{VOQ}(i, j)$ exceeds threshold T during L time slots. We have

$$c_i \leq L \quad \text{for all } i \ (i = 0, 1 \ldots N-1). \tag{1}$$

If c_i of input i is larger than c_k of input k, according to LBOS-F input i will generate more reservation requests and thus get a larger share of output j's bandwidth (output j is over-subscribed). That is

$$a_i \geq a_k, \text{ if } c_i \geq c_k \tag{2}$$

By conditioning on the value of c_i, two cases are considered:

- $c_i < L$: In one or more time slots, the length of $\mathrm{VOQ}(i, j)$ is less than threshold T. Then traffic load b_i is satisfied by allocation a_i, i.e.,

$$\lim_{L \to \infty} c_i / L = b_i = a_i$$

  Therefore, a_i cannot be further increased because a_i is conformed to condition 2.
- $c_i = L$: The length of $\mathrm{VOQ}(i, j)$ is always longer than threshold T. This indicates that traffic load b_i cannot be satisfied by allocation a_i because output j is over-subscribed:

$$\sum a_i = U \tag{3}$$

  From (1), we have:

$$c_i \geq c_k \quad \text{for all } k \ (k = 0, 1 \ldots N-1)$$

  Combine it with (2), we get:

$$a_i \geq a_k, \quad \text{for all } k \ (k = 0, 1 \ldots N-1) \tag{4}$$

  To increase a_i, we have to reduce some $a_k(k = 0, 1 \ldots N-1)$ due to (3). Then we reduce the allocation to some input k for $a_i \geq a_k$ (4), which proves that condition 3 is ensured.

Combining the proof for all the three conditions in Definition 2, we proved that LBOS-F satisfies the max-min fair criterion.

IV. Extensions and Refinements

A. Supporting Multicast

The transmitter module in Fig. 2 consists of a fixed layer array. The lasers are turned on by direct current injection when a packet is to be sent. Data bits are then "written" inside a channel by an external modulator. Laser array facilitates multicasting, where bits can be written *simultaneously* by the external modulator on multiple wavelengths (where the corresponding lasers have been turned on for carrying a multicast packet). In this way, packet "replication" is done in optical domain, where bandwidth efficiency is less critical.

To support multicast, we modify the scheduling algorithm at each linecard as follows. In addition to N unicast $\mathrm{VOQ}(i, k)$'s, we add a FIFO queue for multicast traffic. In each time slot, based on the channel status detected by the wavelength monitor (and the reservation status if LBOS-F is used), linecard i selects a packet for sending among its $N + 1$ local queues. Priority is given to multicast traffic by always examining the multicast queue first. This is because the multicast queue suffers from severe HOL blocking, and sending a multicast packet is more cost-effective. If the HOL multicast packet's fan-out set (i.e., the set of targeted destinations) overlaps with the set of idle channels, replicate and send the multicast packet onto the overlapped wavelengths. Then update the fan-out set by excluding those have been sent. If the updated fan-out set is empty, the multicast packet is removed from the queue. If there are no backlogged multicast packets *or* none of them can be selected (due to zero-overlap between idle channels and the packet's fan-out set), we select a unicast packet for sending using the basic LQF scheduler. To further reduce the HOL blocking experienced by the multicast VOQ, we can add additional multicast queues to each input port.

B. Cutting Down Average Delay by Linecard Placement

In LBOS, the delay experienced by a packet is the summation of the queuing delay at the input linecard and the propagation delay between linecards. Since linecards are connected by a ring network, the inter-linecard distance and thus the propagation delay is predetermined. Let $b_{i,j}$ be the packet arrival rate for $\mathrm{flow}(i, j)$, and $h_{i,j}$ be the propagation delay from linecards i

to j. For a given traffic rate matrix $\{b_{i,j}\}$, the traffic weighted average propagation delay is:

$$H = \left(\sum_j \sum_i b_{i,j} \times h_{i,j} \right) \Big/ \left(\sum_j \sum_i b_{i,j} \right). \quad (5)$$

We have $0 \le b_{i,j} \le 1$ and $0 \le h_{i,j} \le N-1$ for $\forall i,j \in [0, N-1]$. Note that flow(i,i) does not enter the ring, and thus $h_{i,i} = 0$.

Assume $b_{0,3} = 1$ is the only flow in Fig. 1. From (5), we have $H = 3$ slots. If we swap the positions of linecards 0 and 2, H becomes 1 and the propagation delay is minimized. It is shown [19] that finding the optimal linecard placement for minimizing H has the same complexity as the classic traveling salesman problem. Based on the following notations, we formulate the linecard placement problem as an ILP (Integer Linear Programming) problem.

- x_i: the propagation delay of flow$(0,i)$ packets, where $0 \le x_i \le N-1$, for $\forall i \in [0, N-1]$.
- $f_{i,j}$: binary variable and $j > i$ for $\forall i,j \in [0, N-1]$. If $f_{i,j} = 1$, it means $x_i > x_j$ and if $f_{i,j} = 0$, then $x_i < x_j$.

Minimizing (6), shown at the bottom of the page, subject to the following ring topology constraints:

$$x_0 = 0 \quad (7)$$

$$1 \le x_i \le N-1 \quad \text{for } \forall i \in [1, N-1] \quad (8)$$

$$x_i - x_j - N f_{i,j} \ge 1 - N, j > i \quad \text{for } \forall i,j \in [0, N-1] \quad (9)$$

$$x_j - x_i + N f_{i,j} \ge 1, j > i \quad \text{for } \forall i,j \in [0, N-1] \quad (10)$$

Notably, constraints (9) and (10) above are to ensure $x_i \ne x_j$ if $i \ne j$.

In practice, the linecard placement pattern is changed only if there is a significant enough change in traffic matrix. We can implement a LBOS using an OXC (Optical cross-Connect), as shown in Fig. 6. Less expensive OXC with milliseconds reconfiguration delay can be used if the reconfiguration takes place not very frequently.

V. Performance Evaluations

In this section, we study the performance of our proposed LBOS under three types of traffic patterns, admissible, inadmissible (i.e., with over-subscribed outputs) and multicast. For comparison, Fasnet [6], which has a similar hardware complexity as LBOS, is implemented. In simulating Fasnet, we adopt the best parameter settings reported in [6], i.e., $\text{fairness quota} = 100$ and $\text{maximum accumulated quota} = 500$. To be fair, the same simulation environments as Fasnet are adopted for all our simulations, i.e., propagation delay between adjacent linecards is 100 ns ($t_\text{p} = 100$ ns) and each linecard introduces a delay of 100 ns ($t_\text{d} = 100$ ns). Accordingly, the duration of a time slot in LBOS is 200 ns. We further define a time unit to be 100 ns, or half of a time slot. We assume packets can only arrive at the beginning of each time unit. Without pipelined operation, LBOS can send at most one packet in every two time units (Fig. 3). With pipelined LBOS (Fig. 4), one packet can be sent in each time unit.

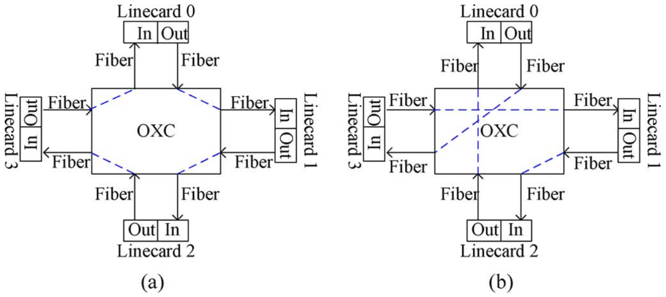

Fig. 6. Two possible linecard placement patterns using OXC: (a) $\{0-1-2-3\}$ and (b) $\{0-3-1-2\}$.

We also implement (a) iSLIP algorithm [20] (with a single iteration), which serves as a benchmark for single-stage input-queued switch, and (b) output-queued switch, which serves as a lower bound. Although we only present simulation results for switch with size $N = 32$ linecards below, the same conclusions and observations apply for other switch sizes.

A. Admissible Uniform Traffic

Admissible uniform traffic is generated as follows. At every time unit for each input, a packet arrives with probability p (input load) and destines to each output with equal probability. From Fig. 7, we can see that non-pipelined LBOS can only obtain up to 50% throughput. For pipelined LBOS, close-to-100% throughput can be obtained. Note that the delay performance is the total delay a packet experienced at input port and en route. For LBOS, the average propagation delay is 32 *time units* or 16 *time slots* (i.e., $N/2$ under uniform traffic with switch size N). From Fig. 7, we can see that the delay performance of LBOS is dominated by the propagation delay. For Fasnet, the average propagation delay is already 64 *time units* or 32 *time slots* (i.e., $(1 + 2N)/2$), leaving alone (a) the extra time required by a Fasnet linecard to detect the free compartment in the "train" of frames, and (b) the extra queuing delay at input ports. Therefore, compared with Fasnet our LBOS gives significantly smaller delay. When $p = 0.6$, Fasnet experiences a delay of

$$H = \frac{\sum\limits_{j>i} \sum\limits_i b_{i,j}[(x_j - x_i) + N f_{i,j}] + \sum\limits_{j<i} \sum\limits_i b_{i,j}[(x_j - x_i) + N(1 - f_{i,j})]}{\sum\limits_j \sum\limits_i b_{i,j}} \quad (6)$$

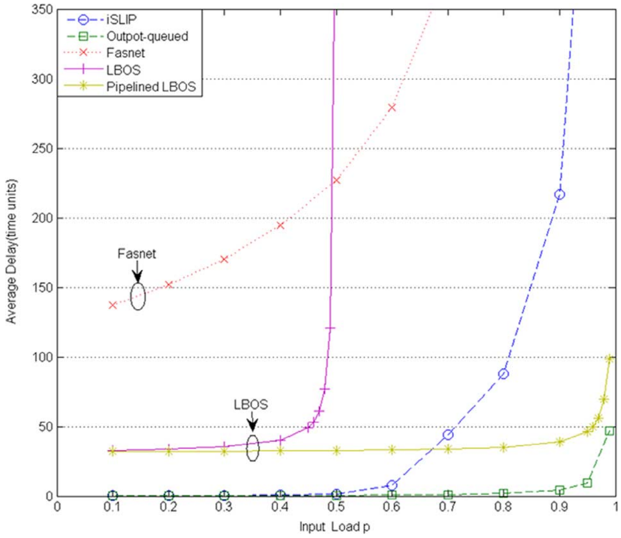

Fig. 7. Delay versus input load, under uniform traffic.

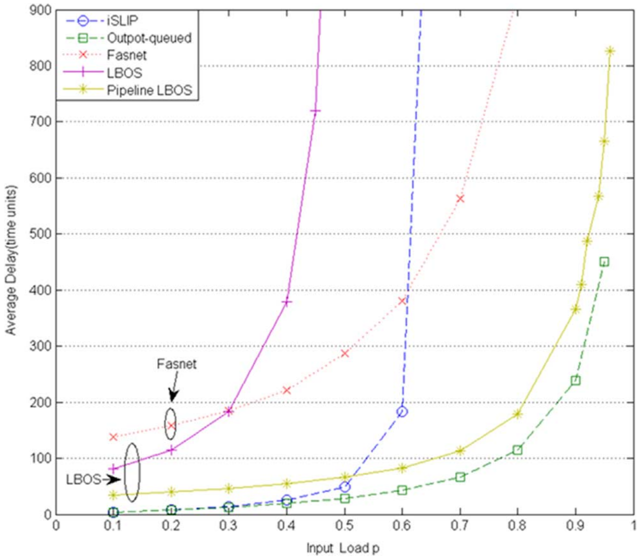

Fig. 8. Delay versus input load, under uniform bursty traffic.

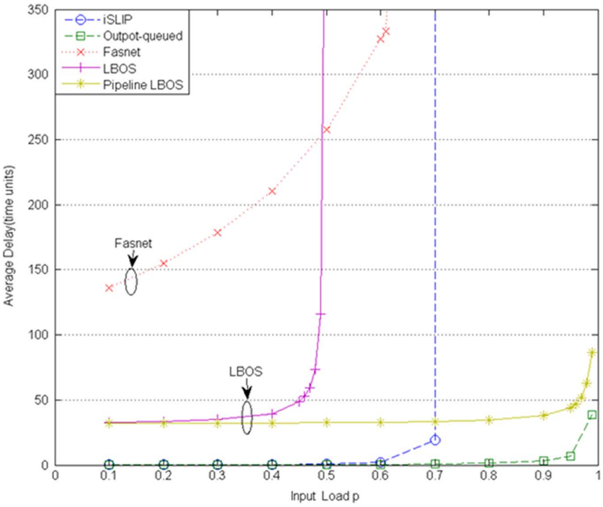

Fig. 9. Delay versus input load, under hot-spot traffic.

279.4 time units, and pipelined LBOS only 33.1, cutting down the delay by more than eight times.

From now on, we shall only focus on the pipelined LBOS and will drop "pipelined" for simplicity.

B. Admissible Uniform Bursty Traffic

Bursty arrivals are modeled by the ON/OFF traffic model, which is a special instance of the two-state Markov-modulated Bernoulli process. In the ON state, a packet arrival is generated in every time unit. In the OFF state, there are no packet arrivals. Packets of the same burst have the same output and the output for each burst is uniformly distributed. Given the average input load of p and average burst size s, the state transition probabilities from OFF to ON is $p/[s(1-p)]$ and from ON to OFF is $1/s$. Without loss of generality, we set burst size $s = 30$ packets. From Fig. 8, we can see delay builds up quickly with input load. For bursty traffic, the input port delay tends to dominate the total delay. At $p = 0.7$, with Fasnet packets experience an average delay of 563.8 time units, whereas for pipelined LBOS is just 112.9.

C. Admissible Hot-Spot Traffic

We assume packets arriving at each input port in each time unit follow the same independent Bernoulli process with probability p. Hot-spots are generated as follows. For input port i, packet goes to output $i + N/2$ with probability 0.5, and goes to other outputs with same probability $1/[2(N-2)]$. From Fig. 9, again we can see that LBOS consistently outperforms Fasnet and delivers close-to-100% throughput.

D. Inadmissible Traffic

We next compare the performance of LBOF, LBOS-F and Fasnet [6] under two inadmissible traffic patterns. (For admissible traffic patterns, LBOS-F generates the same performance as pipelined LBOS and thus not shown in Figs. 7–9.) We first adopt the inadmissible server-client traffic model in [11]. At each time unit for every input, a packet arrives with probability p. Linecards are partitioned into two types: a server (i.e., linecard 0) and $N-1$ clients. The server transmits packets with equal probability to all clients. Each client transmits 1/3 of its traffic toward the server and 2/3 to the other clients with equal probability. When $N = 32$, the amount of traffic going to the server is given by

$$z = p(N-1)/3 = 31p/3.$$

Fig. 10 shows the bandwidth share of three representative flows, (1,0), (24,0) and (31,0), at the server/linecard 0. From Fig. 10, we can see that as the loading at linecard 0 increases (which becomes inadmissible when $z > 1$), with LBOS flows (31,0) and (24,0) are quickly throttled by flow (1,0), due to the ring-fairness problem. With LBOS-F, the three flows equally share the oversubscribed server bandwidth. Although Fasnet also ensures fair resource sharing, the average throughput for each flow is smaller than LBOS-F. This is because LBOS-F ensures close-to-100% maximum throughput, whereas Fasnet cannot.

We also simulate an attack-traffic scenario, where output/linecard 0 is gradually dominated by traffic coming from input/

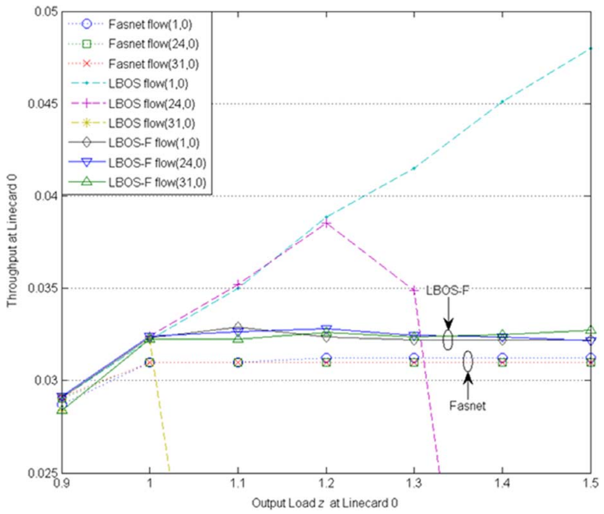

Fig. 10. Linecard 0's throughput versus its output load z, under server-client traffic.

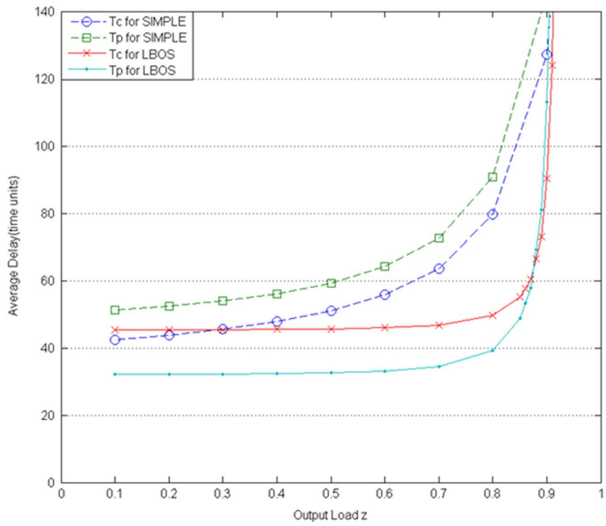

Fig. 12. Delay versus output load, with uniform multicast traffic.

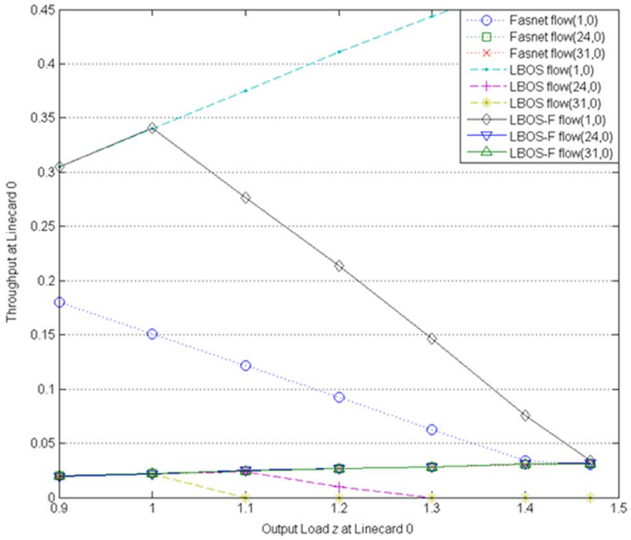

Fig. 11. Linecard 0's throughput versus its output load z, under attack traffic.

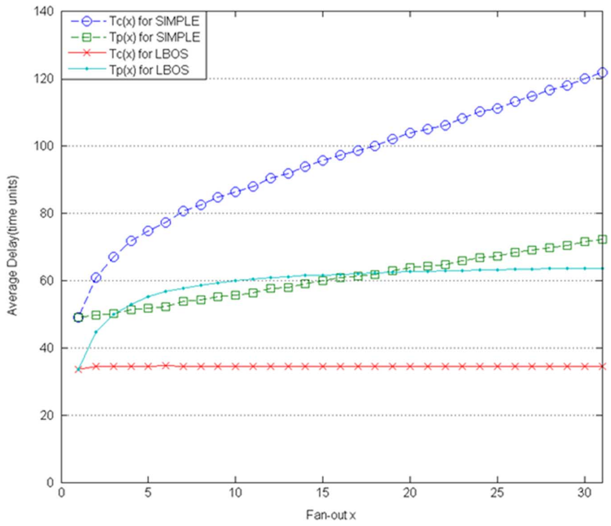

Fig. 13. Delay versus fan-out, with uniform multicast traffic at $z = 0.7$.

linecard 1. The detailed traffic model is as follows. At each time slot for each input port, a packet arrives with probability p. For input 1, an arrived packet goes to output 0 with probability 0.5 (we call it an attack-flow), and the remaining 0.5 probability is equally shared by all other outputs. For any other input i, an arrived packet goes to the $N-1$ outputs (excluding output i) with equal probability. Therefore, at the over-subscribed output 0 and with $N = 32$, the output load z is:

$$z = 0.5p + p \cdot (N-2)/(N-1) = p \cdot 91/62$$

From Fig. 11, as output load z increases, with LBOS the throughputs of flow(31,0) (marked by $*$) and flow(24,0) (marked by $+$) quickly drop to 0, while the throughput for the attack-flow(1,0) (marked by $\bullet$) increases linearly. When LBOS-F and Fasnet are used, the attack-flow(1,0) is regulated/reduced, due to the max-min fair allocation nature of the two algorithms. Specifically, the attack-flow(1,0) can only make use of the excess bandwidth from flows with smaller traffic demands, e.g., flow$(i,0)s$ $(i = 2, 3 \ldots 31)$. From Fig. 11, we can also see that the throughput of flow(1,0) using LBOS-F is significantly higher than that of Fasnet. Again, this is because LBOS-F can achieve higher overall switch throughput than Fasnet.

E. *Multicast Traffic*

We compare the multicast LBOS scheduler (with a single multicast queue per input) in Section IV.A with a simple scheduling algorithm (SIMPLE). In SIMPLE, a multicast packet is replicated to become unicast packets upon its arrival, where each unciast packet joins its own unicast $\text{VOQ}_1(i,k)$ if output k is in the fan-out set. The rest of the operation is the same as the unicast LBOS.

In our simulations, we try to distinguish between the average delay experienced by *all copies* (T_c) of a multicast packet and the average delay experienced by the *last-copy* (T_p) of a multicast packet. Notably, T_p corresponds to the worst-case delay and provides us some insight on the delay variation among different copies of a multicast packet. For all the multicast packets with

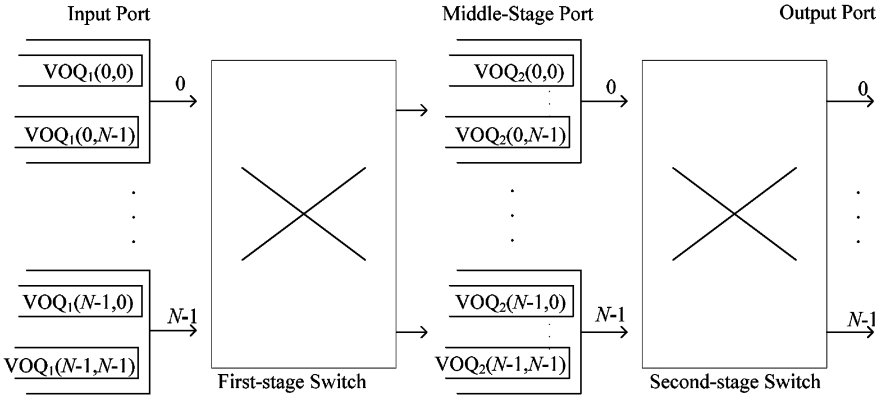

Fig. 14. Load-balanced electronic switch.

fan-out $x, T_c(x)$ and $T_p(x)$ denote their average delay and average last-copy delay respectively. Comparing $T_c(x)$ and $T_p(x)$ for different x's, we can examine the fairness performance in handling packets with different fan-outs.

Multicast traffic is generated as follows. At every time unit for each input, a packet arrives with probability p. If a packet arrives, it has equal probability of being unicast or multicast. If the packet is unicast, it destines to each output with equal probability. If the packet is multicast, its fan-out x is randomly selected between [2, 31] (while excluding the traffic from input i to output i), and the identity of each output in the fan-out set is also randomly selected. Fig. 12 shows the packet delay performance against output load z, where

$$z = p[0.5 + 0.5(2 + 31)/2] = 8.75p. \qquad (11)$$

To ensure the multicast traffic in our simulations is always admissible, we must have $z \leq 1$ (or $p \leq 1/8.75$).

From Fig. 12, we can see that for output load $z < 0.9$, our (multicast) LBOS provides a lower average packet delay than SIMPLE. At $z = 0.8$, LBOS cuts down both the overall average delay (T_c) and the average last-copy delay (T_p) by about two times. Fig. 13 shows the delay performance against different fan-outs, while fixing $z = 0.7$. With LBOS, we can see that $T_c(x)$, the average delay for packets with fan-out x, remains constant at 33 units, whereas $T_p(x)$, the average last-copy delay for packets with fan-out x, increases slowly with x. This shows that LBOS is generally fair among multicast packets with different copies. On the contrary, with SIMPLE algorithm, both $T_c(x)$ and $T_p(x)$ increase rapidly with fan-out x.

F. Performance for Linecard Placement

We randomly generate twenty 16 × 16 admissible traffic matrices. For each matrix, the average propagation delay is calculated using (5) and the average of the 20 matrices is found to be $H = 16.1$ time units. With the optimized linecard placement (by solving the ILP in (6)–(10)), we can get an average propagation delay of 14.1 time units. A saving of 12.3% in propagation delay is obtained. We then carry out simulations to get the average packet delay (by also taking the input port queuing delay into account) for each scenario. We found that without linecard placement, the average delay is 25.9, and with linecard placement, the delay drops to 22.9.

VI. Conclusion

In this paper, we proposed a very simple-yet-effective load-balanced optical switch (LBOS) for designing hybrid electro-optical high-speed router. LBOS comprises N linecards connected by an N-wavelength WDM fiber ring. Each linecard i is configured to receive on channel λ_i. To send a packet, it can select and transmit on an idle channel based on where the packet goes. We showed that the excellent delay-throughput performance yielded by LBOS is due to its capability in balancing the switch load in both time and wavelength domains. To further enhance its performance, LBOS was extended to support pipelined packet sending, receiving and scheduling. A throughput-fair scheduler was proposed for solving the ring-fairness problem of LBOS under inadmissible traffic patterns. Finally, we also extended the proposed LBOS to support efficient multicast as well as linecard placement.

Appendix A

Consider the basic LBOS operation in Fig. 3. If we treat the fiber ring as a FDL, then the ring network "buffers" a packet from linecard i to j for exactly $(j - i) \bmod N$ time slots. Since one round trip time along the ring is N time slots, a specific wavelength channel on the ring can buffer up to N in-flight optical packets. With N wavelengths, the ring can buffer up to N^2 packets. With the above buffering notion in mind, we show that LBOS is an optical counterpart of the load-balanced electronic switch (LBES) in [17].

In fact, the first LBES was proposed in [5]. A LBES consists of two stages of switch fabrics, as shown in Fig. 14. The first fabric converts the non-uniform traffic into uniform (by spreading packets going to different outputs over different middle-stage ports, and packets going to the same output over different time slots), and the second switch fabric delivers packets to their correct outputs. In Fig. 14, $\mathrm{VOQ}_1(i, k)$ represents the VOQ at input i with packets destined for output k, and $\mathrm{VOQ}_2(j, k)$ denotes the VOQ at middle-stage port j with packets destined for output k. Each switch fabric is configured according to a pre-determined and periodic sequence of switch configurations (which removes the need for a centralized scheduler). A possible sequence of configurations is shown in Fig. 15, where at time slot t input i, middle-stage port j and output k are connected according to the following pattern:

$$j = (i - t) \bmod N, k = (j + t) \bmod N.$$

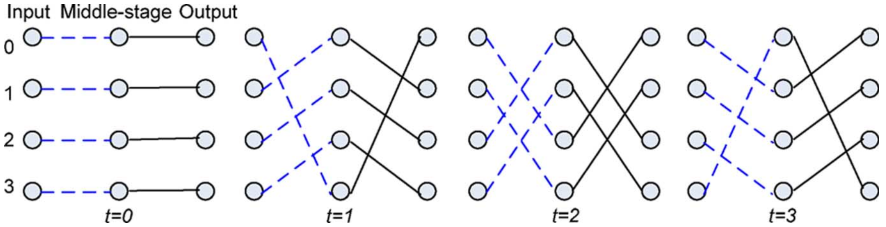

Fig. 15. A joint sequence of configurations for LBES in Fig. 14.

A LBES faces the packet out-of-order problem, as packets belonging to the same flow traverse through different middle-stage ports and experience different amounts of middle-stage port delay. It is shown [17] that with a single packet buffer at each $\mathrm{VOQ}_2(j,k)$, the sequence of switch configurations in Fig. 15, and a feedback-based local scheduler, LBES can overcome the packet out-of-order problem while still yielding close-to-100% maximum throughput. But the scheduler in [17] requires a dedicated feedback packet to be sent from each middle-stage port to its connected output port in every time slot, for reporting its $\mathrm{VOQ}_2(j,k)$'s occupancy status. In [21], the need for dedicated feedback packets is removed by smartly piggybacking an occupancy vector on each data packet sent. But occupancy vector still consumes bandwidth.

It is interesting to point out that our LBOS is an optical counterpart of the LBES in [17], while not incurring any feedback overhead (neither dedicated feedback packets nor piggybacked occupancy vectors). In LBOS, optical packets are "buffered" as they propagate along the fiber ring in different wavelengths, which mimics the buffering services offered by the middle-stage $\mathrm{VOQ}_2(j,k)$'s in Fig. 14. In a specific time slot, the channel status (i.e., idle or not) of all the wavelengths passing by, which is equivalent to the occupancy vector of $\mathrm{VOQ}_2(j,k)$, will be conveniently detected by the wavelength monitor on each linecard—the need for dedicated feedback packets/vectors is thus removed. In fact, a one-to-one mapping between every instance of sequence in Fig. 15 and the corresponding operation on the ring network in Fig. 1 can be found.

References

[1] D. Pao, N. H. Liu, A. Wu, K. L. Yeung, and K. S. Chan, "Efficient hardware architecture for fast IP address lookup," *IEE Proc. Computers & Digital Tech.*, vol. 150, no. 1, pp. 43–52, Jan. 2003.

[2] B. Wu, K. L. Yeung, M. Hamdi, and X. Li, "Minimizing internal speedup for performance guaranteed switches with optical fabrics," *IEEE/ACM Trans. Netw.*, vol. 17, no. 2, pp. 632–645, 2009.

[3] B. Wu, K. L. Yeung, P. H. Ho, and X. Jiang, "Minimum delay scheduling for performance guaranteed switches with optical fabrics," *J. Lightw. Technol.*, vol. 27, no. 16, pp. 3453–3465, Aug. 2009.

[4] I. Keslassy, S. T. Chuang, K. Yu, D. Miller, M. Horowitz, O. Solgaard, and N. McKeown, "Scaling the internet routers using optics," presented at the ACM SIGCOMM'03, Karlsruhe, Germany, Aug. 2003.

[5] C. S. Chang, D. S. Lee, and Y. S. Jou, "Load balanced Birkhoff-von Neumann switches, part I: One-stage buffering," *Comput. Commun.*, vol. 25, pp. 611–622, 2002.

[6] A. Bianco, E. Carta, D. Cuda, J. M. Finochietto, and F. Neri, "A distributed scheduling algorithm for an optical switching fabric," presented at the IEEE ICC 2008, Beijing, China, May 2008.

[7] A. Carena, V. D. Feo, J. Finochietto, R. Gaudino, F. Neri, C. Piglione, and P. Poggiolini, "RINGO: An experimental WDM optical packet network for metro applications," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 8, pp. 1561–1571, Oct. 2004.

[8] A. Bianco, J. M. Finochietto, G. Giarratana, F. Neri, and C. Piglione, "Measurement-based reconfiguration in optical ring metro networks," *J. Lightw. Technol.*, vol. 23, no. 10, pp. 3156–3166, Oct. 2005.

[9] A. Antonino, A. Bianco, A. Bianciotto, V. D. Feo, J. M. Finochietto, R. Gaudino, and F. Neri, "Wonder: A resilient WDM packet for metro applications," *Opt. Switching Netw.*, vol. 5, no. 1, pp. 19–28, Mar. 2008.

[10] A. Bianco, D. Cuda, J. M. Finochietto, F. Neri, and M. Valcarenghi, "Wonder: A pon over a folded bus," presented at the IEEE GLOBECOM 2008, New Orleans, LA, Nov. 2008.

[11] A. Bianco, D. Cuda, J. M. Finochietto, and F. Neri, "Multi-metaring protocol: Fairness in optical packet ring networks," presented at the IEEE ICC 2007, Glasgow, Scotland, May 2007.

[12] A. Bianco, D. Cuda, J. M. Finochietto, F. Neri, and C. Piglione, "Multi-fasnet protocol: Short-term fairness control in WDM slotted MANs," presented at the IEEE ICC 2006, Paris, France, May 2006.

[13] I. M. White, M. S. Rogge, K. Shrikhande, and L. G. Kazovsky, "A summary of the HORNET project: A next-generation metropolitan area network," *IEEE J. Sel. Area Commun.*, vol. 21, no. 9, pp. 1478–1494, Nov. 2003.

[14] *Distributed Queue Dual Bus (DQDB) Subnetwork of a Metropolitan Area Network (MAN)*, IEEE Standard 802.6, Dec. 1990.

[15] M. C. Yuang, I. F. Chao, Y. M. Lin, B. C. Lo, P. L. Tien, and S. S. W. Lee, "A high-performance optical access and control system for packet-switched WDM metro ring networks," presented at the IEEE GLOBECOM 2008, New Orleans, LA, Nov. 2008.

[16] M. C. Yuang, Y. M. Lin, and Y. S. Wang, "A novel optical header processing and access control system for a packet-switched WDM metro ring network," *J. Lightw. Technol.*, vol. 27, no. 21, pp. 4907–4915, Nov. 2009.

[17] H. I. Lee, B. C. Lee, and S. W. Seo, "A load balancing scheme for two-stage switches maintaining packet sequence," presented at the IEEE ICC 2006, Istanbul, Turkey, Jun. 2006.

[18] M. Hosaagrahara and H. Sethu, "Max-min fair scheduling in input-queued switches," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 4, Apr. 2008.

[19] K. L. Yeung and T. S. P. Yum, "Node placement optimization in shuffleNets," *IEEE/ACM Trans. Netw.*, vol. 6, no. 3, pp. 319–324, Jun. 1998.

[20] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," *IEEE/ACM Trans. Netw.*, vol. 7, no. 2, pp. 188–201, Apr. 1999.

[21] B. Hu and K. L. Yeung, "Feedback-based scheduling for load-balanced two-stage switches," *IEEE/ACM Trans.* [Online]. Available: http://www.eee.hku.hk/~kyeung/Ton.pdf, accepted for publication

Bing Hu (S'06–M'10) received the B.Eng. and M.Phil. degrees in communication engineering from University of Electronic Science and Technology of China in 2002 and 2005, respectively. He received the Ph.D. degree in the Department of Electrical and Electronic Engineering, The University of Hong Kong, in 2009.

He is currently an Assistant Professor in the Department of Information Science and Electronic Engineering, Zhejiang University. His research interests include next-generation Internet, high speed packet switch/router design and all-optical networks.

Kwan L. Yeung (SM'99) was born in 1969. He received the B.Eng. and Ph.D. degrees in information engineering from The Chinese University of Hong Kong in 1992 and 1995, respectively.

He joined the Department of Electrical and Electronic Engineering, The University of Hong Kong in July 2000, where he is currently an Associate Professor. His research interests include next-generation Internet, packet switch/router design, all-optical networks and wireless data networks.