

LIONS: An AWGR-Based Low-Latency Optical Switch for High-Performance Computing and Data Centers

Yawei Yin, *Member, IEEE*, Roberto Proietti, *Member, IEEE*, Xiaohui Ye, *Student Member, IEEE*, Christopher J. Nitta, *Member, IEEE*, Venkatesh Akella, and S. J. B. Yoo, *Fellow, IEEE*

Abstract—This paper discusses the architecture of an arrayed waveguide grating router (AWGR)-based low-latency interconnect optical network switch called LIONS, and its different loopback buffering schemes. A proof of concept is demonstrated with a 4×4 experimental testbed. A simulator was developed to model the LIONS architecture and was validated by comparing experimentally obtained statistics such as average end-to-end latency with the results produced by the simulator. Considering the complexity and cost in implementing loopback buffers in LIONS, we propose an all-optical negative acknowledgement (AO-NACK) architecture in order to remove the need for loopback buffers. Simulation results for LIONS with AO-NACK architecture and distributed loopback buffer architecture are compared with the performance of the flattened butterfly electrical switching network.

Index Terms—Buffers in switches, optical interconnections, optical switches, optics in computing.

I. INTRODUCTION

WHILE the advances in CMOS technologies continue to support Moore's Law [1] in increasing the number of transistors in a die, microprocessor performance is expected to continue to increase with parallel processing of many cores on-chip. Amdahl's law [2] suggests that a parallel computing system with balanced processing, memory, and communications performs best across most applications. A balanced system with 100 TeraFLOPs/second computing speed and 100 TB memory would need 100 TB/s (800 Tb/s) bisection bandwidth. The future high-performance computing (HPC) systems and Data Centers implemented with multicore processors will require terabytes/second of bandwidth for processor to processor and processor to memory. It is expected to be increasingly difficult to meet the scalable, high-bandwidth density and low-latency communication requirements of these future large data centers and terascale computing systems using conventional electrical interconnects.

Optical interconnects exploiting the inherent wavelength division multiplexing parallelism could overcome those limitations. In the past few years, some efforts have been made in designing optical switching architecture by considering the particular challenges faced by the networks supporting datacenters and high-performance computers. Among all the proposed optical switching architectures in literature, the optical shared memory supercomputer interconnect system (OSMOSIS) [3], [4], data vortex [5], and low-latency interconnect optical network switch (LIONS, previously named as DOS) [6] are three pioneering architectures that have attracted a fair amount of attention. The simulation comparison in [6] shows that LIONS provides low-latency and high-throughput switching and does not saturate even at very high ($\sim 90\%$) input load. In addition, the average latency and throughput of LIONS do not change dramatically as the port number increases, which is scalable beyond that of OSMOSIS or data vortex.

This paper discusses the architecture of LIONS together with its different loopback buffering schemes. Specifically, this paper presents the hardware demonstration of a 4×4 LIONS testbed, and compares the experimental results with simulation results to verify the correctness of the simulator developed to model the LIONS architecture. The experimental and simulation studies indicate that the loopback buffers in the LIONS architecture will be the main limiting factor for scalability. In order to overcome this limitation, we propose the all-optical negative acknowledgement (AO-NACK) architecture. Moreover, the synthetic traffics as well as giga-updates per second (GUPS) benchmarks are used to evaluate the performance of AO-NACK architecture in comparison with traditional LIONS with loopback buffers and the electrical switching network using a flattened butterfly (FBF) topology.

The remainder of this paper is organized as follows. Section II describes the comprehensive architecture of LIONS together with the different loopback buffer strategies. Section III presents the proof of concept testbed demonstration of a 4×4 LIONS with shared loopback buffer (SLB) architecture, and compares the experimentally collected statistics with the simulation results in order to verify the accuracy of our simulator. Section IV introduces the LIONS with AO-NACK architecture which removes the scalability limitation of the loopback buffer. The results of performance simulations comparing the different architectures on both synthetic and benchmark workloads are also presented here. Section V concludes this paper.

Manuscript received June 1, 2012; revised July 10, 2012; accepted July 11, 2012. Date of publication July 17, 2012; date of current version April 3, 2013. This work was supported by the Department of Defense under Contract H88230-08-C-0202 and Google Research Awards.

The authors are with the Departments of Electrical and Computer Engineering and Computer Science, University of California at Davis, Davis, CA 95616 USA (e-mail: yyin@ucdavis.edu; rproietti@ucdavis.edu; xye@ucdavis.edu; cjnitta@ucdavis.edu; akella@ucdavis.edu; yoo@ece.ucdavis.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSTQE.2012.2209174

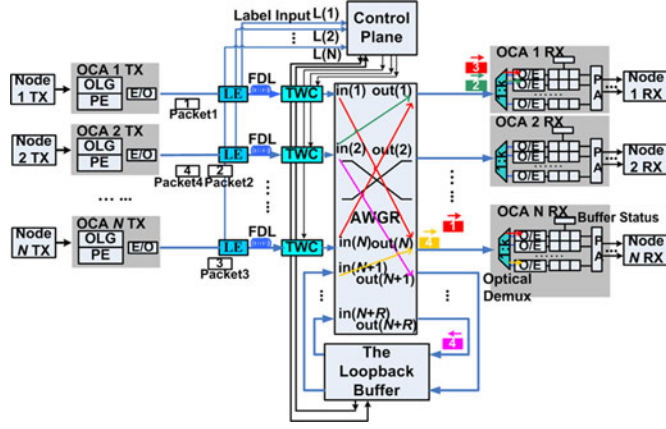


Fig. 1. LIONS architecture.

II. LIONS WITH DIFFERENT LOOPBACK BUFFERING SCHEMES

It is well known that traditional electrical switch output queuing is capable of achieving lower switch latency and higher throughput than input queuing [7] if the necessary speedup ($\times N$) is incorporated in the output queue. However, when the line rate and the switch radix increase, output queuing becomes difficult to implement (primarily because the output queues with high aggregated bandwidth are difficult to realize). Therefore, electrical switch designs focus on complicated input queuing structures, such as virtual output queue (VOQ), and associated multistage arbitration schemes. Arrayed waveguide grating router (AWGR) switching fabric does not suffer from this limitation as they have the unique strength of wavelength parallelism, which allows optical wavelengths to cross over and propagate in parallel.

Fig. 1 shows the general architecture of LIONS, which has at its core an AWGR, tunable wavelength converters (TWCs), an electrical control plane (CP), electrical loopback buffers, label extractors (LEs), and fiber delay lines. Between the switch and each end node, there is an optical channel adapter that serves as the media interface [6]. In particular, 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 this requires a total of N^2 receivers for the whole switch. 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 an optical switch, the store-and-forward mechanism cannot be applied due to the lack of feasible optical buffers. In the proposed LIONS switch, the electronic buffers are placed in the loopback path, referred to here as the loopback buffer. Note that the LIONS switch uses a *forward-store* strategy, as opposed to the store-and-forward strategy employed in an electrical switch. Only the contended packets that fail to get grants from the arbiter are stored. The loopback buffer in LIONS plays an important role in contention resolution. In this section, we compare the three proposed loopback buffer architectures in LIONS, referred

to as the SLB, the distributed loopback buffer (DLB), and the mixed loopback buffer (MLB).

A. SLB

The SLB has the structure shown in Fig. 2(a). A $1:N$ optical DEMUX and N receivers are necessary, because the SLB may receive delayed packets from different inputs concurrently on different wavelengths. A $N:1$ 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 $1:N$ 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 packet destined for the same output should be sent out serially. The transmission for a delayed packet can 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.

B. DLB

The loopback buffer will become more scalable if the queues can be organized based on the input ports instead of on the output ports. 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. However, the buffer controller design will become more complicated, as contention may occur among queues for different inputs. Moreover, the head-of-line (HOL) blocking may occur, and end-to-end latency may increase.

Instead of using VOQs with complicated arbitration to alleviate the effect of the HOL blocking, we can also exploit the intrinsic wavelength parallelism to do so and keep the arbitration relatively simple. We can deploy multiple transmitters for each queue, thus making it capable of sending multiple packets to different switch outputs on different wavelengths concurrently. Considering scalability and cost, only a fixed number of transmitters should be deployed for each queue. The packets that are

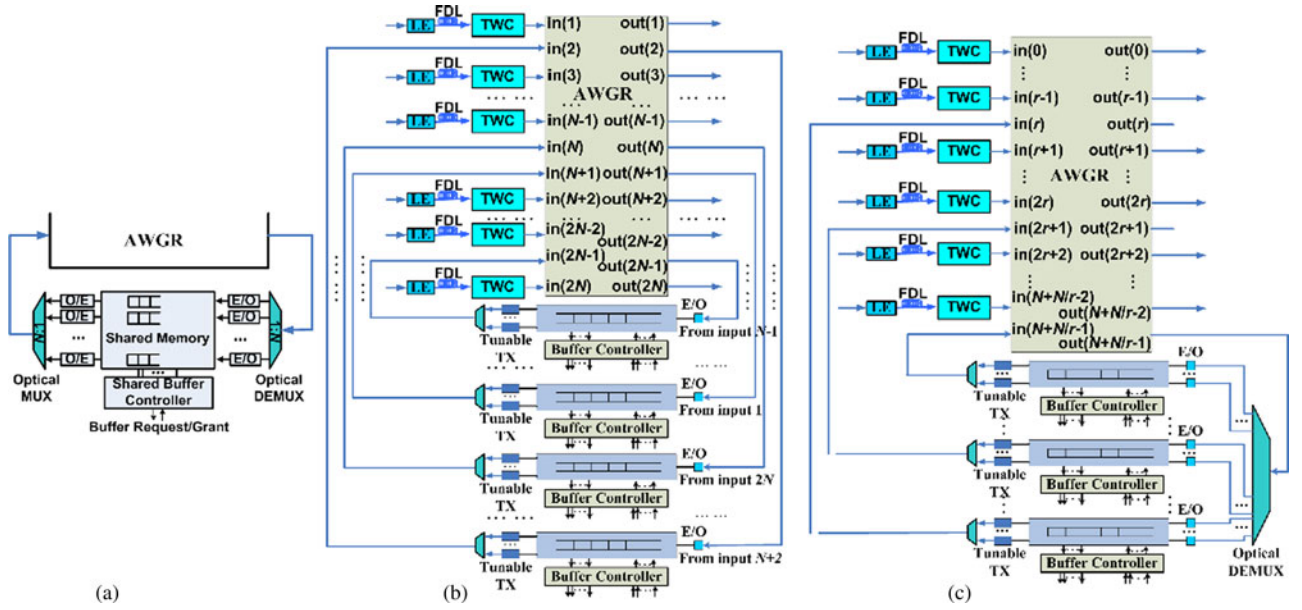


Fig. 2. LIONS with (a) SLB, (b) DLB, and (c) MLB.

waiting for transmission at the head of each VOQ should gain the grant of the transmitters first, and then the one who wins one transmitter can make the request to the arbiters in the CP. In this case, with multiple transmitters, multiple VOQs in each queue can make more than one request at a time to the CP arbiter, and thus make more efficient use of the resources and avoid HOL blocking.

The DLB uses tunable transmitters, because packets stored at one queue may go to different AWGR outputs. Therefore, each queue in the DLB should be connected to a separate AWGR input. Then a $2N \times 2N$ AWGR is needed to connect to N end nodes and the DLB, as shown in Fig. 2(b). The blocked packets from a single AWGR input port are all directed to a certain AWGR output that is connected to the dedicated queue in DLB. Each input queue in DLB corresponds to an AWGR input port. 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 to the input i and connect its corresponding DLB queue to the input j ($0 \leq i, j \leq 2N$). Assuming there are at least two contention groups, then if i and j satisfy the relation $j = \text{mod}(i + N + 1, 2N)$, they 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 comes out of the queue in DLB.

C. MLB

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. 2(c). 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 and one output if the switch connects to N end nodes. Note that in principle, the MLB could also use N/r output ports on AWGR, but that would require N/r of $1 : r$ optical DEMUX at the input of each queue in MLB. 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. All the couplers used in LIONS are assumed to be 2:1 couplers for simplicity. Therefore, for MLB with r tunable transmitters at each queue, $(r - 1)$ couplers are needed for each queue, and since there are N/r queues in total, $(r - 1) \times (N/r)$ couplers are required in total. 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 CP. 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

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 DEMUX	One 1: N optical DEMUX	One 1: N optical DEMUX	None
Number of Optical MUX	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 Memory	1	N/r	N

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 $\{j | \text{mod}(j, r + 1) = 0, 0 \leq j < N + N/r\}$ 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 an input set of $\{i | i = \text{mod}(j + N/2 + N/(2 \times r) + h, N + N/r), 1 \leq h \leq r\}$, so that the set of inputs and the input connected to the corresponding queue always belong to different contention groups. When r is a fixed small number, although the memory used in the MLB requires more I/O bandwidth than the memory used in the DLB does, the former requires much less I/O bandwidth than the memory used in the SLB does. Overall, the MLB solution is a tradeoff between the DLB solution and SLB solution in terms of required I/O bandwidth.

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.

The performance evaluation of LIONS with three kinds of loopback buffers are presented in [8]. Overall, LIONS 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 buffered packets. Therefore, the buffered 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 as when they first arrived at the switch. In addition to the gain in performance, the DLB and the MLB require much less memory

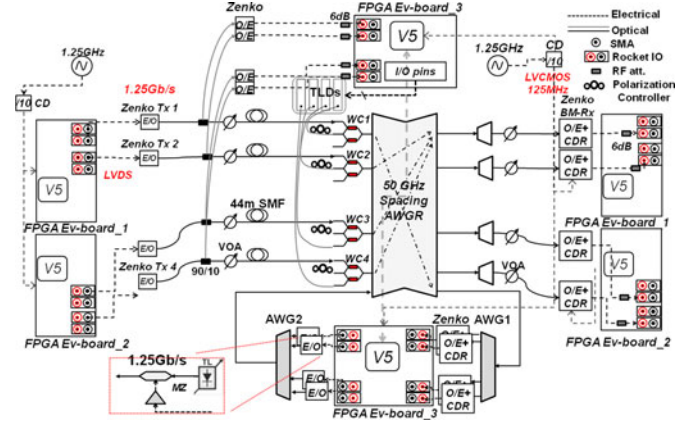


Fig. 3. 4×4 LIONS testbed with SLB.

I/O bandwidth than does the SLB. Although the DLB with multiple transmitters per queue performs a little bit better than MLB, the MLB occupies fewer AWGR ports and requires fewer tunable transmitters.

III. HARDWARE DEMO OF LIONS AND EXPERIMENTAL VERIFICATION OF THE SIMULATOR

Although the performance of LIONS with DLB and MLB is much better than with SLB, the complexity and cost of these two architectures restrain them from a laboratory-scale implementation. Moreover, other than studying the feasibility, the major purpose of implementing LIONS is to verify the correctness of the simulator we developed. No matter which architecture of the loopback buffers is chosen, an accurately modeled LIONS in simulator should have similar or even identical performance with the actually implemented testbed. As far as one of the LIONS architectures can be modeled correctly in a reasonable scale, we will be able to verify that the library of different modules in LIONS has been established accurately in the simulator. Therefore, the projection of LIONS to higher port count or the projection to other buffer architectures becomes reasonably easy, since it is only a matter of adjusting the parameters or adjusting the buffer arrangements in the simulator. In light of this, we choose the most feasible SLB architecture to implement among all the three architectures of LIONS loopback buffers.

The 4×4 LIONS testbed is depicted in Fig. 3. As shown, a 32×32 50-GHz spacing AWGR constitutes the core of the switch architecture. It also includes wavelength converters (WCs) based on cross-phase modulation in a semiconductor optical amplifier Mach-Zehnder interferometer. Each WC accepts one continuous wave input signal from a tunable laser diode (TLD) board. The TLD guarantees nanosecond switching time over the C band with a wavelength accuracy of 0.02 nm. By reading the 5-bit parallel control signals coming from the field-programmable gate array (FPGA)-based CP, each TLD board tunes its wavelength according to a routing table stored on a complex programmable logic device chip mounted on the board itself. An optical path is established between each AWGR input and output on a packet basis, according to the destination address carried by each packet label.

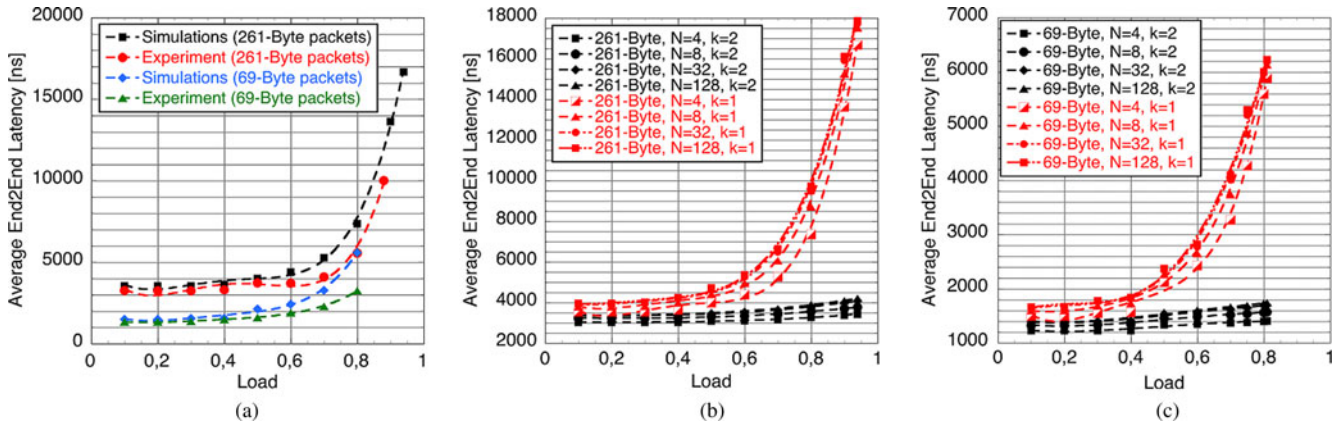


Fig. 4. (a) Comparison of simulation results with experimental results. (b) Projection of simulation results to high port count with large packet size and $k = 1, 2$. (c) Projection of simulation results to high port count with small packet size and $k = 1, 2$.

The CP and loopback buffers are implemented using a Xilinx Virtex 5 FPGA ML523 Characterization Platform, which is capable of instantiating eight high-speed RocketIO GTP tile transceivers, is connected to 16 pairs of differential SMA connectors. Four pairs of the transceivers are used as the four CP channels, each of which receives the labels from its LE (i.e., the 90/10 splitter). The labels are encapsulated in the packet headers in the time domain for simplicity. Note that the labels can be wavelength multiplexed to the payload in order to completely separate control and data plane, which is done in the simulator. The CP reads the label and generates the 5 bit control signals to TLD boards after arbitration. The contended packets that fail to win arbitrations are directed to the inputs of the loopback buffer. The I/O ports of loopback buffers are implemented using another four pairs of RocketIO transceivers. The Zenko burst mode transceivers are widely used in the testbed beside the RocketIO interfaces of end-hosts, CP, and loopback buffers, and the burst mode clock and data recovery (BM-CDR) modules are used at the receivers connected to the output ports of the AWGR, since the WC will turn the optical power on and off while switching.

The LIONS testbed uses two Virtex 5 FPGA platforms to emulate the multiprocessor parallel computing system. Four of MicroBlaze Soft Processor Cores [9] were instantiated on Virtex 5 FPGA with MPI interfaces capable of doing remote direct memory access operations. The generated data are first written to the BRAM block on FPGA and then moved into the RocketIO transmitter output queue using direct memory access (DMA) operation. Then, the packets are encapsulated and deserialized by RocketIO at the 1.25-Gb/s output line rate. On the Rx side, the received data packets are directly moved from the input queue to the DDR2 SDRAM memory on board using DMA operation.

The end-to-end latency is one of the important performance metrics to the switch. A synthetic traffic model is used in the testbed. The data streams at each host are encapsulated into fixed size packets with uniform random destination address. Each packet is with 5 B header (2 B preamble, 1 B destination address, 1 B source address, and 1 B packet length). Different offered load can be achieved by changing the guard time between packets. Note that a minimum guard time of 17 B has to be guaranteed due

to the hardware constraints (i.e., worst case TLD tuning time, burst mode receiver settling time, and comma alignment delay in SERDES). Since the traffic is uniform randomly distributed, the end-to-end latency statistics can be collected at any of the output ports. In the experiment, only host 2 was used to collect data.

Fig. 4(a) shows the comparison of statistic results from both simulations and experiments. The black and red lines show the 4×4 experimental and simulation data, respectively, with $k = 1$ and packet size of 256 B plus 5 B header, while the blue and gray lines show the same results with packet size of 64 B. Here, k means that the number of parallel wavelengths can be received by the same host simultaneously from one output port of the switch [6]. As shown, the comparison of the results shows a close match between the experimental data and the simulation data, which verifies the correctness and accuracy of the simulator we developed. The other curves in Fig. 4(b) and (c) show the projection of the results to high port count, and also to $k = 2$ case. As depicted, the increase of the LIONS radix does not significantly affect the end-to-end latency, while $k = 2$ can dramatically reduce it since it reduces the contention probability at each output port.

The testbed implementation was limited by the available off-the-shelf components. Therefore, the line rate of the testbed was limited because the commercially available Zenko BM-CDR modules are running at 1.25 Gb/s. Despite the low line rate of the demo, the match between the simulation and experiment is still meaningful considering that the simulation at 10 Gb/s or higher line rate will only change the packet transmission time. It is important to note in our architecture that the CP can operate at a lower data rate because the label associated with each packet can be modulated on a separate wavelength. So, even with an FPGA, 10-Gb/s data rate can be supported with 1.25-Gb/s label rate and an application-specified integrated circuit-based CP can be used to go beyond that. The arbitration process stays the same no matter what line rate is used in the simulation. The scalability of the arbitrations in LIONS switch has been addressed in our previous papers. In particular, Ye *et al.* [6] showed that the arbitration logic (which is the bottleneck to scalability to larger number of ports) can be implemented in a distributed manner and scales beyond the traditional electrical switch.

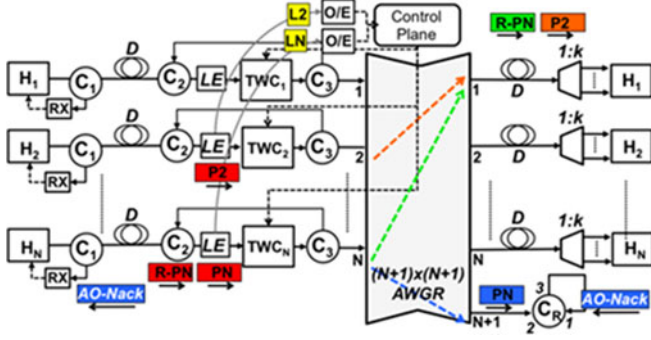


Fig. 5. Interconnect architecture using AO-NACK. H: host; C: optical circulator; LE: label extractor; TWC: tunable wavelength converter; AWGR: array waveguide grating router; RX: AO-NACK receiver.

IV. BUFFERLESS LIONS WITH AO-NACK

Although LIONS can support low-latency switching under high input loads, its loopback buffers require complex and fair amount of memory and TX/RX components running at the line rate, which becomes a bottleneck for implementation. The AO-NACK technique is capable of eliminating this bottleneck by promptly notifying the end nodes whenever one of their packets cannot reach the desired output due to contention.

Fig. 5 shows an architecture using the AO-NACK technique. N hosts connect to an $(N + 1) \times (N + 1)$ AWGR by means of a fiber of length D , which represents the host-switch distance. TWCs used at each of the N input ports perform the switching function in the optical domain. Each input port is also equipped with two optical circulators (OCs) to separate the on-the-fly packets (the packets traveling toward the AWGR input ports) from the counterpropagating (traveling backwards) AO-NACKs. LEs separate the low-speed label signals from the high-speed payloads and send the labels to the low-speed electronics CP. After O/E conversion, the CP processes the label and sends the control signals to the TWCs to switch the packets according to their destination. As explained in detail in [6], optical parallelism in AWGR can be used to reduce contention probability— k inputs can reach the same output port using different wavelengths. Then, a $1 : k$ optical demux at each host receiver side separates the different signals traveling simultaneously on the same fiber. For simplicity, let us assume that $k = 1$. If two packets (e.g., P2 and PN) from different inputs are contending for the same output (output1), the CP switches one packet (e.g., P2) to the desired output, while the other packet (PN) is switched to the $N + 1$ port (reflective port). An OC used as shown in Fig. 5 reflects the packet (PN) back to its sender (H_N). An OC at the host site ($C1$) extracts the counterpropagating packet, which now acts as AO-NACK. A dedicated receiver is then used to detect the AO-NACK and trigger the retransmission. If $L/2D \geq 1$, where L is the packet length (in meters), the AO-NACK reaches the sender while the transmission for the related packet is still happening or it has just finished. In this case, a simple edge detector is sufficient to detect the AO-NACK since there is no ambiguity about which packet the AO-NACK refers to. If $L/2D < 1$, the received AO-NACK is related to a packet of which the transmission is completed. Since there

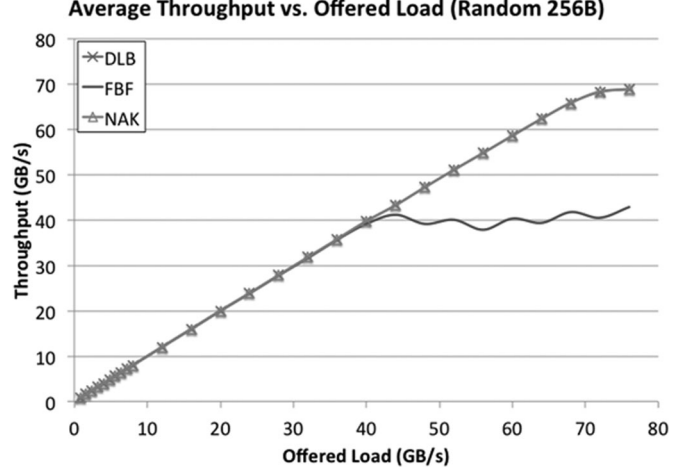


Fig. 6. DLB, FBF, and NAK network throughput versus offered load for 256 B packets on uniform random traffic.

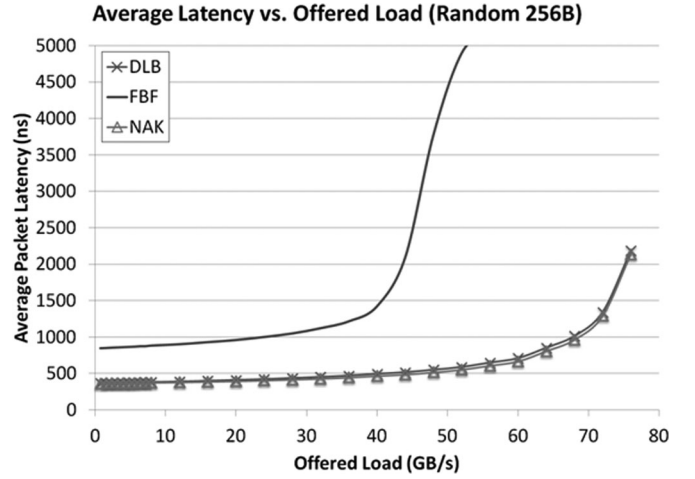


Fig. 7. DLB, FBF, and NAK network latency versus offered load for 256 B packets on uniform random traffic.

may be several on-the-fly packets, an edge detector can be still used, but the sender needs to use a time stamp for each on-the-fly packet. If the counter expires (the time counter value can be fixed since the AO-NACK arrival time is deterministic), the sender can then assume that the associated packet has reached the desired output. Otherwise, packet retransmission is triggered. Another solution could consist of including in the packet header an on-the-fly packet sequence number field of a few bits and then receiving and reading only the first few bytes of the AO-NACK messages. In this case, it is necessary that the AO-NACK technique preserves the packet content. Note that the passive nature of AWGR and OC (C_R) guarantees that this technique can intrinsically reflect multiple packets simultaneously without any crosstalk effect. This aspect, together with the fact that an AO-NACK cannot contend with other packets or other AO-NACKs, makes this technique robust.

A proof of concept demonstration of AO-NACK technique can be found in [10], where a host-switch distance of ≈ 20 m and a packet length of 204.8 ns are used. The notification of

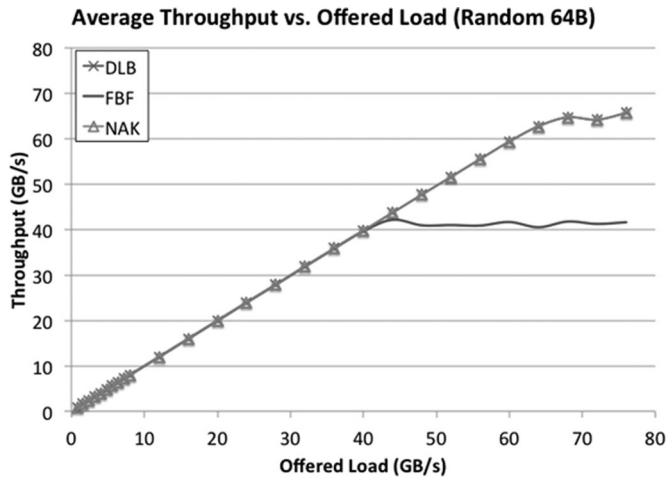


Fig. 8. DLB, FBF, and NAK network throughput versus offered load for 64 B packets on uniform random traffic.

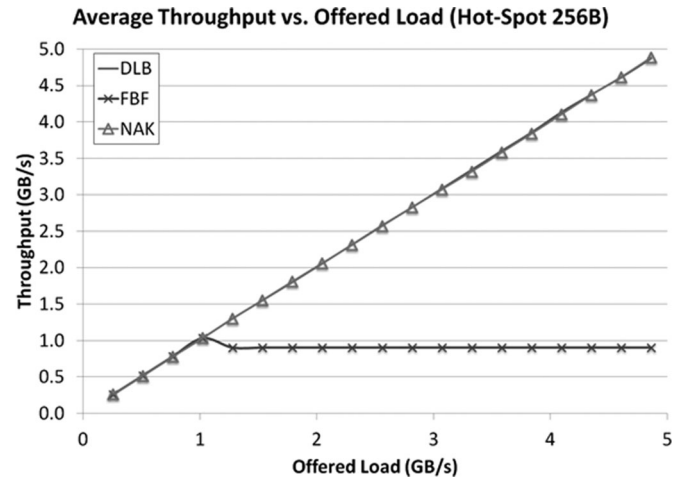


Fig. 10. DLB, FBF, and NAK network throughput versus offered load for 256 B packets on hot-spot traffic.

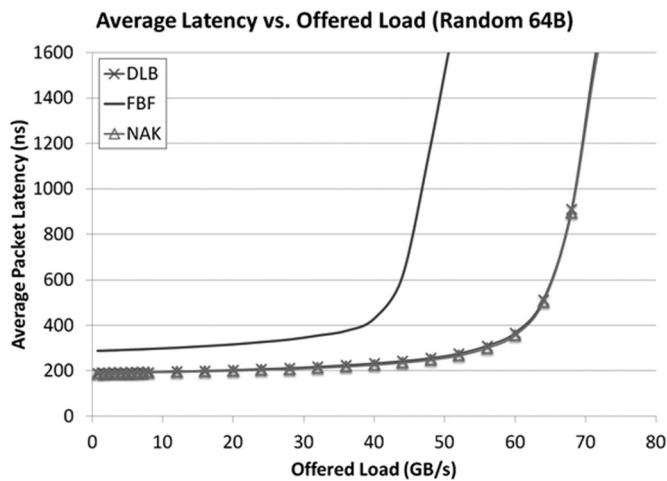


Fig. 9. DLB, FBF, and NAK network latency versus offered load for 64 B packets on uniform random traffic.

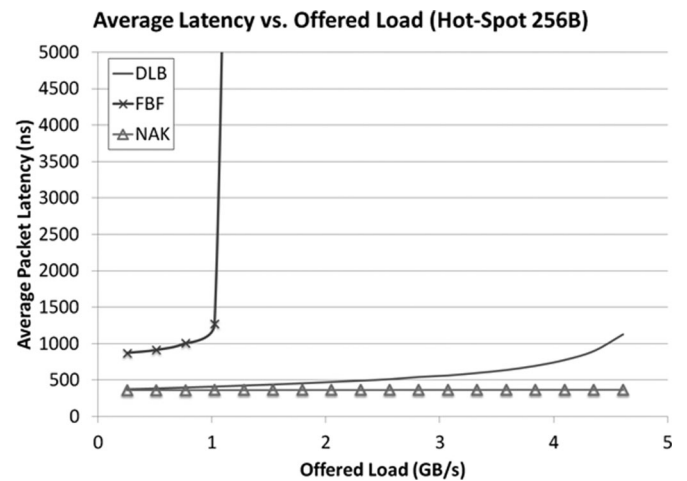


Fig. 11. DLB, FBF, and NAK network latency versus offered load for 256 B packets on hot-spot traffic.

AO-NACKs messages and successful packet retransmission is demonstrated with error-free operation at 10 and 40 Gb/s.

In order to evaluate the performance of AO-NACK architecture, the simulator has been developed to correctly model the DLB, AO-NACK, and the FBF architecture [11]. The DLB architecture was chosen since it represents the best performance among all the three loopback buffer architectures. The simulated FBF consists of a 4×4 grid of routers with four nodes connected to each router in a $20 \text{ m} \times 20 \text{ m}$ square. Each node is connected to every other node in each row and column. The radius of each router is 10. The distance from each node to its “local” router is 1.75 m, with each router 5 m from its nearest neighbor. In the DLB and AO-NACK architecture, the distance between each node and the centralized switch is assumed to be 10 m. Note that $k = 4$ means that the receiver bandwidth for each AO-NACK node is four times that of the FB nodes—this may appear to be an unfair comparison; however, remarkably, the FBF network has 2.5 times as many links as the AO-NACK [12].

Both synthetic traffic model and benchmarks are used in the simulator to evaluate the performance of LIONS, and the results are compared with the FBF switching network.

The synthetic traffic simulations consisted of uniform random and hot-spot traffic with packet sizes of 256 and 64 B. Figs. 6–Fig. 9 show the throughput and latency for uniform random traffic with 256 and 64 B packets.

The performance of DLB and the NAK is almost identical for all synthetic traffic patterns—the difference between the two is almost indistinguishable in the figures. The DLB and NAK architectures greatly outperform the FBF for larger packet sizes, but as the packet size decreases the difference between the AWGR-based networks and the FBF becomes less apparent. The diminished performance of the AWGR-based networks is primarily due to the time required to tune the wavelengths.

The throughput and latency results for hot-spot traffic are shown in Figs. 10–Fig. 13. Note that the offered load is limited to 5 GB/s since the networks are theoretically limited to four

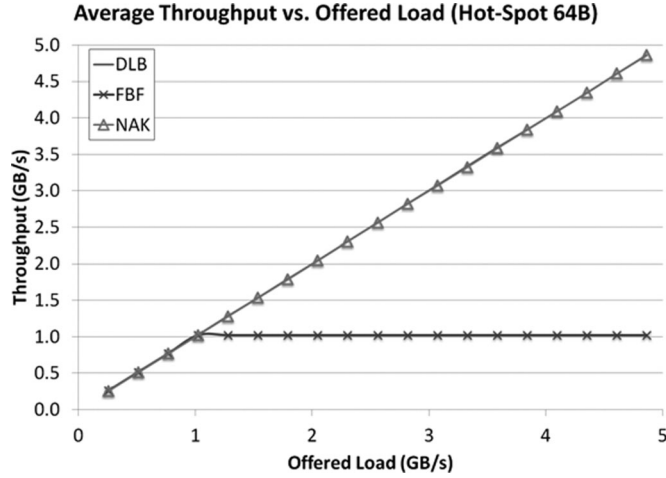


Fig. 12. DLB, FBF, and NAK network throughput versus offered load for 64 B packets on hot-spot traffic.

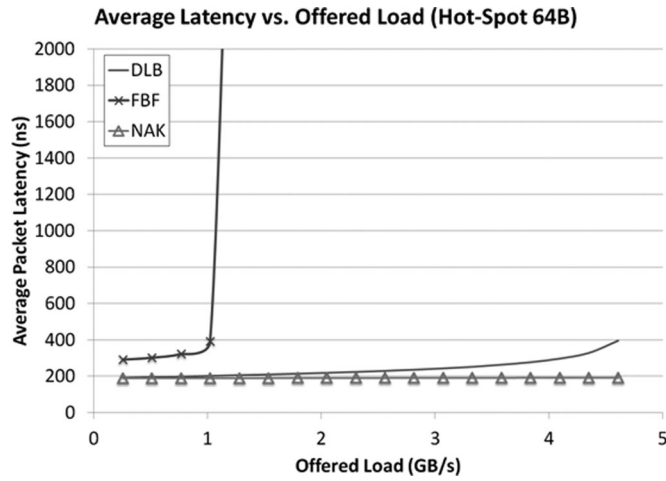


Fig. 13. DLB, FBF, and NAK network latency versus offered load for 64 B packets on hot-spot traffic.

10-Gb/s receivers. The maximum throughput of the AWGR-based networks is four times that of the FBF since K was set to four for these simulations. The AWGR networks are fairly insensitive to the packet size on the hot-spot traffic pattern due to the relatively low total load being offered.

The GUPS benchmark is of particular interest in high-performance computation and typical of in-memory database applications that implements *transactional* nature of query processing. Each “update” requires a node to read a random memory location, modify the value, and then write back to the same memory location. The GUPS benchmarking for the three networks simulated a 64-node shared memory system with a 64-bit address space. The updates were of 64-bit data values and 1024 outstanding requests were allowed per node. Each nonlocal update required a 64-bit read request, 128-bit read reply (64-bit address and 64-bit data), and a 128-bit write (64-bit address and 64-bit data). The simulations were run allowing outstanding requests to be aggregated into larger packets and also without aggregation.

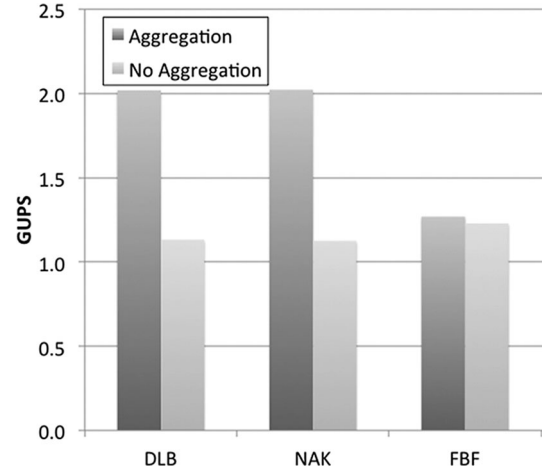


Fig. 14. DLB, NAK, and FBF in GUPS benchmark.

The LIONS networks performed poorly without aggregation due to the small average packet size and the tuning time discussed previously. The LIONS networks greatly outperformed the FBF when aggregation was allowed, and actually approached the theoretical maximum GUPS for the network configuration, as shown in Fig. 14. These results show that the tuning time for the AWGR-based networks is critical to overall system performance, and that data aggregation can dramatically improve performance.

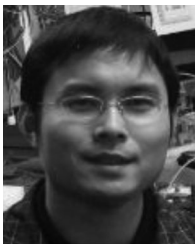
V. CONCLUSION

This paper discusses the architecture of LIONS together with its different loopback buffer schemes. A proof of concept demonstration of a 4×4 LIONS testbed is presented. A simulator is developed to model the LIONS architecture and was calibrated and verified by the experimental results. Although LIONS with DLB performs, in theory, the best in terms of latency and throughput, the AO-NACK architecture can eliminate the complex and costly loopback buffers while maintaining the same level of performance as DLB. The LIONS outperforms FBF in synthetic traffic, whereas it outperforms FBF in GUPS benchmark when aggregation is allowed. As the next step, we are currently developing the interface for the FPGA boards and 10-Gb/s BM-CDRs for the testbed.

REFERENCES

- [1] G. E. Moore, “Cramming more components onto integrated circuits,” *Electronics*, vol. 38, pp. 114–117, 1965.
- [2] G. Amdahl, “Validity of the single processor approach to achieving large-scale computing capabilities,” in *Proc. AFIPS Conf.*, 1967, pp. 483–485.
- [3] 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, May–Jun. 2006.
- [4] R. Hemenway, R. R. Grzybowski, C. Minkenberg, and R. Luijten, “Optical-packet-switched interconnect for supercomputer applications,” *J. Opt. Netw.*, vol. 3, pp. 900–913, 2004.
- [5] 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. Lightw. Technol.*, vol. 26, no. 13, pp. 1777–1789, Jul. 2008.

- [6] 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. Archit. Netw. Commun. Syst. (ANCS)*, Oct. 2010, pp. 1–12.
- [7] C. Shang-Tse, A. Goel, N. McKeown, and B. Prabhakar, "Matching output queueing with a combined input/output-queued switch," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 6, pp. 1030–1039, Jun. 1999.
- [8] X. Ye, R. Proietti, Y. Yin, S. J. B. Yoo, and V. Akella, "Buffering and flow control in optical switches for high performance computing," *J. Opt. Commun. Netw.*, vol. 3, pp. A59–A72, Aug. 2011.
- [9] M. Hubner, K. Paulsson, and J. Becker, "Parallel and flexible multiprocessor system-on-chip for adaptive automotive applications based on Xilinx MicroBlaze soft-cores," presented at the 19th IEEE Int. Parallel Distrib. Process. Symp., Washington, DC, Apr. 2005.
- [10] R. Proietti, Y. Yin, R. Yu, X. Ye, C. J. Nitta, V. Akella, and S. J. B. Yoo, "All-optical physical layer NACK in AWGR-based optical interconnects," *IEEE Photon. Technol. Lett.*, vol. 24, no. 5, pp. 410–412, Mar. 2012.
- [11] J. Kim, W. J. Dally, and D. Abts, "Flattened butterfly: A cost-efficient topology for high-radix networks," in *Proc. 34th Annu. Int. Symp. Comput. Archit.*, 2007, pp. 126–137.
- [12] R. Proietti, C. J. Nitta, X. Ye, Y. Yin, V. Akella, and S. J. B. Yoo, "Performance of AWGR-based optical interconnects with contention resolution based on all-optical NACKs," presented at the Opt. Fiber Commun. Conf., Los Angeles, CA, Mar. 2012.



Yawei Yin (M'11) received the B.S. degree in applied physics from the National University of Defense Technology, Changsha, China, in 2004, and the Ph.D. degree in electrical engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2009.

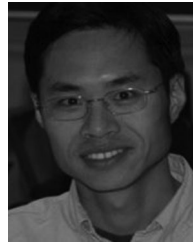
He is currently a Postdoctoral Researcher with the Next Generation Networking Systems Laboratory, University of California at Davis, Davis. His research interests include optical switching technologies and architectures for supercomputing and data

center applications, as well as flexible bandwidth elastic optical networking algorithms, simulations, and experiments.



Roberto Proietti (M'11) received the M.S. degree in telecommunications engineering from the University of Pisa, Pisa, Italy, in 2004, and the Ph.D. degree in electrical engineering from Scuola Superiore Sant'Anna, Pisa, in 2009.

He is currently a Postdoctoral Researcher with the Next Generation Networking Systems Laboratory, University of California, Davis. His research interests include optical switching technologies and architectures for supercomputing and data center applications, high spectrum efficiency coherent transmission systems, and elastic optical networking.



Xiaohui Ye (S'07) received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 2002 and 2005, respectively. He is currently working toward the Ph.D. degree at the Next Generation Networking Systems Laboratory, University of California, Davis.

His research interests include various topics related to heterogeneous networks, including optical switching technologies and architectures for supercomputing and data center applications.



Christopher J. Nitta (S'06–M'11) received the Ph.D. degree in computer science from the University of California, Davis, in 2011.

He is currently a Postdoctoral Researcher and Lecturer at the University of California, Davis. His research interests include network-on-chip technologies, embedded system and real-time operating system design, and hybrid electric vehicle control.



Venkatesh Akella received the Ph.D. degree in computer science from the University of Utah, Salt Lake City, in 1992.

Since 1992, he has been a Professor of electrical and computer engineering at the University of California, Davis. His current research encompasses various aspects of embedded systems and computer architecture with special emphasis on embedded software, hardware/software codesign, and low-power system design.

Dr. Akella is a member of Association for Computing Machinery and is the recipient of the National Science Foundation CAREER Award.



S. J. B. Yoo (S'82–M'84–SM'97–F'07) received the B.S. (with distinction), M.S., and Ph.D. (with minor in physics) degrees from Stanford University, Stanford, CA, in 1984, 1986, and 1991, respectively, all in electrical engineering.

He is currently a Professor of electrical engineering at the University of California, Davis (UC Davis). His research at UC Davis includes optical switching devices, systems, and networking technologies for the future computing and communications. Prior to joining UC Davis in 1999, he was a Senior Research

Scientist at Bellcore, leading technical efforts in optical networking research and systems integration. He participated in the advanced technology demonstration network/multiwavelength optical networking testbed integration and a number of standardization activities including GR-2918-CORE, GR-2918-ILR, GR-1377-CORE, and GR-1377-ILR on dense wavelength-division multiplexing and OC-192 systems.

Dr. Yoo is a Fellow of the Optical Society of America, and a recipient of the Defense Advanced Research Projects Agency Award for Sustained Excellence in 1997, the Bellcore CEO Award in 1998, the Outstanding Mid-Career Research Award (UC Davis) in 2004, and the Outstanding Senior Research Award (UC Davis) in 2011.