Baldur: A Power-Efficient and Scalable Network Using All-Optical Switches

Mohammad Reza Jokar[1], Junyi Qiu[2], Frederic T. Chong[1], Lynford L. Goddard[2], John M. Dallesasse[2], Milton Feng[2], and Yanjing Li[1]

[1]Department of Computer Science, University of Chicago
[2]Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign
Email: jokar@uchicago.edu, qiu12@illinois.edu, chong@cs.uchicago.edu,
{lgoddard, jdallesa, mfeng}@illinois.edu, yanjingl@uchicago.edu,

***Abstract*—We present the first all-optical network, Baldur, to enable power-efficient and high-speed communications in future exascale computing systems. The essence of Baldur is its ability to perform packet routing on-the-fly in the optical domain using an emerging technology called the transistor laser (TL), which presents interesting opportunities and challenges at the system level. Optical packet switching readily eliminates many inefficiencies associated with the crossings between optical and electrical domains. However, TL gates consume high power at the current technology node, which makes TL-based buffering and optical clock recovery impractical. Consequently, we must adopt novel (bufferless and clock-less) architecture and design approaches that are substantially different from those used in current networks.**

At the architecture level, we support a bufferless design by turning to techniques that have fallen out of favor for current networks. Baldur uses a low-radix, multi-stage network with a simple routing algorithm that drops packets to handle congestion, and we further incorporate path multiplicity and randomness to minimize packet drops. This design also minimizes the number of TL gates needed in each switch. At the logic design level, a non-conventional, length-based data encoding scheme is used to eliminate the need for clock recovery.

We thoroughly validate and evaluate Baldur using a circuit simulator and a network simulator. Our results show that Baldur achieves up to 3,000X lower average latency while consuming 3.2X-26.4X less power than various state-of-the-art networks under a wide variety of traffic patterns and real workloads, for the scale of 1,024 server nodes. Baldur is also highly scalable, since its power per node stays relatively constant as we increase the network size to over 1 million server nodes, which corresponds to 14.6X-31.0X power improvements compared to state-of-the-art networks at this scale.

***Keywords*-optical computing; all-optical network; exascale computing; datacenter network;**

I. Introduction

Power-efficient, high-speed, and scalable networks play a critical role in exascale high-performance computing (HPC) systems. In this paper, we present the first complete design of an all-optical network, Baldur, which is built using all-optical switches to perform network packet routing completely in the optical domain. Baldur consumes less power while achieving significantly higher network performance compared to state-of-the-art networks, and is highly scalable to over 1 million server nodes to enable exascale computing.

In large-scale networks, optical links are the main communication fabric because they are significantly more efficient than electrical links [1], [2]. Current networks use optical links but electrical switches, which leads to many inefficiencies due to the need to cross between optical and electrical domains. Optical switching proposals exist in the literature, but they still rely on electrical packet header processing [3], [4], [5]. In contrast, Baldur controls optical switches entirely in the optical domain. This is highly desirable as it removes significant overheads such as packet buffering, optical-to-electrical (O-E) conversions, and electrical-to-optical (E-O) conversions, etc., and enables on-the-fly packet processing and switching.

In Baldur, optical processing and switching is made possible by a new device technology called the transistor laser (TL) [6], [7], [8], [9], which can be used to construct optical logic gates [10]. TL gate prototypes have already been fabricated [11], which validates the TL as a feasible and promising technology.

However, with opportunities the TL also brings along unique constraints and challenges. Therefore, novel architecture and design approaches are essential. There are two key challenges of an all-optical design: (1) currently there is no clear path to dense optical memories [12], [13]; (2) a TL gate at the current technology node consumes >100X higher power than a 32 nm CMOS gate; therefore, optical clock recovery and complex logic functions are impractical. These technology constraints lead us to several unorthodox architectural and design decisions that are largely unfavorable for electrical networks but turn out to be highly suitable for TL networks, which include: (1) multi-stage networks with randomized connections between network stages [14] to achieve high scalability and immunity to worst-case traffic patterns; (2) low-radix (e.g., 2x2) switches to minimize design complexity; (3) packet drops and re-transmissions to handle network congestion; and (4) path multiplicity inside the network switches [14], [15] to minimize packet drop rate.

Moreover, to enable practical and efficient implementation of these architectural ideas, we design our 2x2 optical switch based on a non-conventional asynchronous data encoding scheme. This design consists of 1,112 TL gates only, and consumes 96.6X less power than a 2x2 electrical switch.

We thoroughly evaluate our Baldur network and show that it has several major advantages. First, Baldur provides 3.2X-26.4X power improvements and up to 3,000X performance improvement compared to state-of-the-art electrical networks

2378-203X/20/$31.00 ©2020 IEEE
DOI 10.1109/HPCA47549.2020.00022

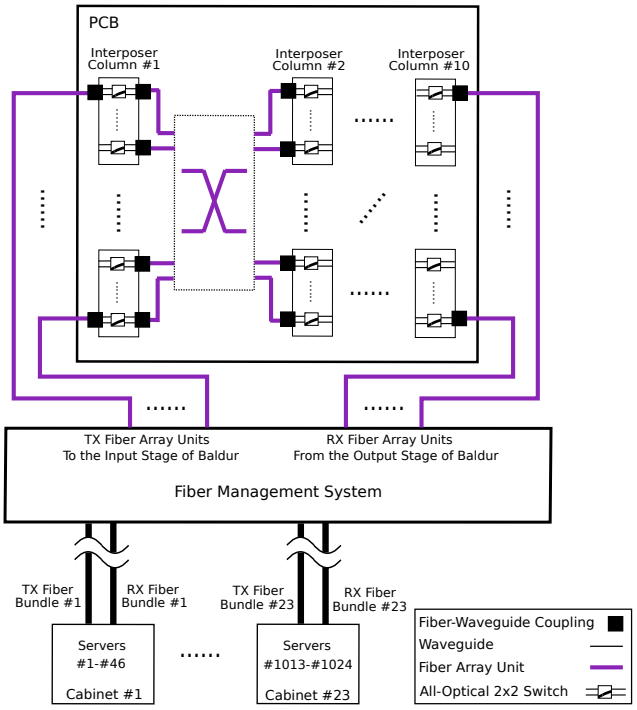


Figure 1: Baldur Network with 1,024 Server Nodes.

(including dragonfly [16], fat-tree [17], and multi-butterfly [18]) under various representative traffic patterns and real workloads. Second, Baldur's architectural property makes it immune to worst-case traffic patterns [19]. Third, Baldur is highly scalable, unlike the widely-deployed dragonfly and fat-tree networks whose scalability is limited by switch radix. Finally, Baldur eliminates the need to manage a hierarchy of network switches – for example, as shown in Figure 1, Baldur connects 1,024 server nodes using only a single 1,024-port switch that fits in one cabinet.

Our key architectural contribution is that, through in-depth analysis and exploration, we obtain the fundamental understanding of various tradeoffs between network design parameters and the TL technology. Our research spans device, circuit/logic, and architecture/system layers to create the first complete design of an all-optical network that is optimized for the TL, which achieves dramatic improvements vs. existing networks even in the presence of various limitations/constraints of the current TL technology node. The major efforts of this work include:

- We characterize TL gates using a detailed device-level simulator.
- We design an all-optical TL switch, and fully validate its correctness and reliability by performing detailed circuit simulations.
- We introduce the Baldur network, a novel architecture optimized for our TL switches.
- We thoroughly evaluate the performance, power, cost, and scalability of Baldur, and quantitatively compare it with state-of-the-art networks to demonstrate its major benefits.

The rest of the paper is organized as follows. Sec. II provides the motivation for this work. Sec. III provides the background on TL gates and device-level simulation results. Sec. IV presents the architecture and design of Baldur. We show our results in Sec. V and Sec. VI. Sec. VII presents related work. Finally, Sec. VIII concludes the paper.

II. Motivation

Exascale computing systems are projected to have 100,000 10 teraFLOP or 1,000,000 1 teraFLOP server nodes, and each node is required to support very high (e.g., 200 GBps) network bandwidth [20]. Thus, networks with high power efficiency, high bandwidth, and low latency are essential in order to satisfy the high communication demand between such large numbers of nodes.

A. Limitations of Existing Networks

In today's HPC and datacenter environments, the networks typically consist of electrical switches (to perform header processing and packet switching), electrical links for short-distance communication, and optical links for long-distance communication. Such networks are referred to as *electrical networks* in this paper. Dragonfly [16] and fat-tree [17] networks are two representative electrical networks that are widely deployed. However, their scalability is limited by the switch radix. In both networks, as the number of server nodes increases, the radix of network switches also increases. However, it is not practical to build a single >64-radix switch with high bandwidth per port mostly due to power constraints [21], [22]. It is also not practical to build >64-radix switches using multiple low-radix switches [22], since that also increases network power, cost, and latency significantly (e.g., based on our analysis, a 128K-node fat-tree network built using 80-radix switches consumes 6.4X more power per node compared to a 1,024-node fat-tree network built using 16-radix switches). Given that the radix is no more than 64, fat-tree and dragonfly networks are limited to 66K and 263K nodes, respectively [16], [17].

Another class of well-known network architectures is the stage-based networks. An example of such networks is multi-butterfly with randomized connections between network stages, which is highly scalable and immune to worst-case traffic patterns [18]. However, electrical multi-stage networks are expensive/impractical in large scales due to significant O-E/E-O/SerDes overheads. For example, based on our analysis, a radix-2 multi-butterfly with multiplicity of 4 consumes 223.5 W per node at the scale of 1,024 nodes – 6X higher than fat-tree – and 41.7% of the power is attributed to O-E/E-O conversions and SerDes units.

In the literature, networks that utilize optical switching elements have also been proposed [3], [4], [5]. Arrayed Waveguide Grating Routers (AWGRs) are one of the most popular examples [3], [5], [23], and they allow multiple input ports to send packets to one output port simultaneously using different wavelengths. However, in these networks, header processing is still performed in the electrical domain, which not only increases packet latency, but also incurs many of the same inefficiencies as electrical networks (i.e., significant

packet buffering and O-E/E-O/SerDes overheads). Moreover, the scalability of AWGR-based networks is also limited by the switch radix. Using typical 32-radix AWGRs, the network size is limited to 128K nodes [24]. Table I summarizes the limitations of these existing networks.

B. Opportunities and Challenges of the TL All-Optical Network

To the best of our knowledge, no practical technology existed to perform optical computing in the past. As the TL emerges as a key technology capable of high-speed optical computing, it provides a well-grounded basis to create all-optical networks, where *both header processing and packet switching are performed in the optical domain*. This is highly desirable in HPC and datacenter networks because:

- All-optical processing and switching eliminates the overheads associated with O-E/E-O conversions and SerDes units, which leads to significant reduction in power.
- It also eliminates the high power overheads associated with packet buffering (including memories, virtual channel allocation, and control logic [25], [26], [27]).
- Eliminating the aforementioned overheads allows us to efficiently realize the benefits (high scalability and immunity to worst-case traffic patterns) of stage-based networks.
- An all-optical network enables on-the-fly packet processing and switching, which reduces network latency significantly.

Despite its high potentials, properties associated with the current TL technology constrain network design choices. First, optical buffering is not practical and consumes high power [12], [13]. For example, using TL latches [10] as optical buffers can lead to a >1,000X power consumption increase in Baldur. Therefore, we do not consider optical buffers an option. Second, due to the high power consumption of TL gates which makes optical clock recovery impractical, as well as the need to perform ultra-fast switching, the routing algorithm must be simple and clock-less. Hence, complex adaptive/oblivious routing techniques are infeasible.

C. Architectural Implications and Novelty of TL Networks

The opportunities and challenges associated with the TL technology impose several key architectural implications as summarized in Table II, which lead to drastically different design decisions compared to current networks.

First, while multi-stage topologies (with randomized connections between network stages [14]) are expensive/impractical in electrical networks as discussed in Sec. II-A, they turn out to be highly efficient and optimized for TLs because all buffering/E-O/O-E/SerDes overheads are eliminated by TL's high-speed optical computing capability. Second, while many electrical networks (such as dragonfly and fat-tree) favor high-radix and low-diameter designs to minimize packet latency and power overheads associated with O-E/E-O conversions and SerDes, low-radix (e.g., 2x2) switches are required for TL networks. This is because the routing algorithms must be simple enough so that they can be implemented using TL gates without incurring significant power costs, but switching power/complexity increases exponentially as the radix increases. Moreover, the diameter in TL networks can be high because the network latency is very low and there are no E-O/O-E/SerDes overheads. Third, in TL networks, since there are no optical buffers, a packet must be dropped and re-transmitted if network congestion occurs. While dropping/re-transmitting packets generally leads to high packet latency in electrical networks, its performance impact on a TL network is much smaller because the TL provides high-speed, in-flight switching which leads to ultra-low network latency. Lastly, we leverage previous theoretical results and incorporate path multiplicity inside the TL switches to minimize packet drop rate. Path multiplicity provides more than one possible path to deliver a packet from the switch input port to the designated output destination [14], [15]. This idea incurs high cost for electrical networks, but is adequate for TL networks – given that TL networks must be constructed using low-radix switches, and that there is no SerDes/E-O/O-E overheads, the benefits obtained from a reduced packet drop rate are well worth the complexity/cost associated with path multiplicity.

Table I: Limitations of Existing Networks.

	Electrical Multi-butterfly [18]	Dragonfly / Fat-tree [16], [17]	AWGR Networks [24]
Performance	Limited by slow electrical header processing & switching		Limited by slow electrical header processing although they perform optical switching
Scalability	Limited by high power	Limited by switch radix (66K-263K nodes)	

In summary, although the architectural ideas in Table II have largely grown out of favor for current electrical networks due to various practical concerns, with the new TL technology which presents drastically different tradeoffs, we judiciously revisit these ideas out of the large architectural design space and incorporate them to create the Baldur network, which successfully achieves superior results.

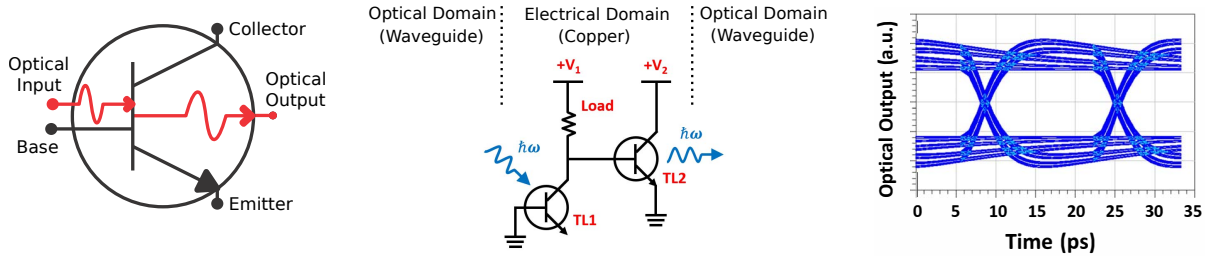
III. The Transistor Laser Technology

In this section, we provide background information on the transistor lasers (TLs), the key technology enabler for Baldur. The TL device is an InGaP/GaAs heterojunction bipolar transistor (HBT) with the addition of quantum-wells for photon generation and optical cavity for coherent light output. As shown in Figure 2(a), a TL has three electrical ports and two optical ports. It can act as a transistor, a direct-modulated laser, and a photodetector, depending on various current and voltage bias conditions. When an electrical current is applied to the base, a proportional electrical current is generated at the collector port, like a transistor. If the collector-emitter voltage is higher than a threshold, a coherent optical output is also generated and the TL works as a laser. When the base-collector junction sees an optical input, a photocurrent is generated and the TL functions as a photodetector.

Individual high-speed and energy-efficient TL devices have been fabricated [31], [32]. In this paper, we focus on using the TL devices to construct *optical logic gates*, i.e., logic

Table II: The Differences in Architectural Design Considerations between Electrical and TL Networks.

	Electrical Networks	TL Networks
Multi-stage Networks	Impractical (large SerDes/O-E/E-O/switching overheads)	Efficient (no SerDes/O-E/E-O and low-cost switching)
Low Radix	Undesirable (due to high diameter which also leads to large SerDes/O-E/E-O overheads)	The only viable option and the cost/overhead is low
Packet Drop	Undesirable (due to high re-transmission overhead)	The only viable option (since there are no optical buffers) and the re-transmission overhead is low
Path Multiplicity	high cost (large SerDes/O-E/E-O/switching overheads)	Low cost especially for low radix



(a) (b) (c)

Figure 2: (a) The Transistor Laser. (b) The Optical TL Inverter. (c) The Eye Diagram of a TL Inverter Operating at 60 Gbps.

Table III: TL Device and Circuit Parameters.

Device Parameters	Junction Capacitance: 100 fF [28] Spon. Recombination Lifetime: 37 ps [28], [29] Photon Lifetime: 2.72 ps [28], [29] Wavelength: 980 nm[1] TL Threshold Current: 0.1 mA[2] Bias Current: 0.2 mA[2]
Circuit Parameters (Typical Condition)	Voltage Supplies: 1.32 V (+V1), 0.6 V (+V2) Load Resistor: 5 ohm Base Current Modulation: 0.2 mA Collector Tunneling Modulation: 17 μA PD Junction Capacitance: 100 fF Average PD Current: 0.1 mA

[1] The TLs can produce different wavelengths by using materials with different bandgaps to support wavelength division multiplexing (WDM).
[2] These values are on-par with micro-cavity VCSELs [30].

Table IV: Device-Level Simulation Results for TL Gates (same results apply for inverter, NAND, NOR, AND, and OR gates).

Area (μm^2)	Rise/Fall Time (ps)	Delay (ps)	Power (mW)	Data Rate (Gbps)
25	7.3	1.93	0.406	60

gates whose inputs and outputs are all optical signals. Figure 2(b) shows the schematic of a TL inverter gate. Here, TL1 functions as a photodetector and a pull-down current source, and TL2 functions as a laser that provides optical output. TL1 generates photocurrent in response to its optical input and lowers the voltage at the base terminal of TL2, thereby modulating TL2's optical output. This basic TL inverter gate structure is the basis for logic gates with multiple inputs. For example, by adding photodetectors either in parallel or in series in the pull-down branch, NOR and NAND gates can be constructed. Similarly, replacing the left branch of the TL inverter with a pull-up source allows the construction of AND and OR gates. Moreover, the optical power of a signal will not deteriorate as it passes through the TL gate because the TL gate restores signal strength. In addition to logic gates, an optical latch can be constructed using two cross-coupled TL NOR gates [10].

The preliminary round of TL gate fabrication has been successfully completed [11]. We are currently targeting a TL technology node which may be fabricated with reasonable cost and high yield rate in the near future, and the detailed simulation results (obtained using the Keysight Advanced Design System (ADS) software) are reported below. We are in the process of optimizing the fabrication flow of the TL gates to validate these simulation results, as well as scaling the TL technology further to continue to improve latency/power.

Table III shows the device and circuit parameters used in the simulation of a TL inverter gate, and the simulation results are summarized in Table IV. An interesting property in TL logic gates is that the same power and speed results apply to multi-input NAND, NOR, AND, and OR gates as well. This is because, although a multi-input TL gate requires multiple TLs at the optical input, only one TL is required at the output, and the TL at the output is the speed/power-limiting element. The power consumption and speed will be the same as long as the average photocurrent is maintained at the same level across different gates, regardless of the number of inputs. Therefore, all TL logic gates consume 0.406 mW at 60 Gbps[1], or 6.77 fJ/bit (a TL latch consists of 2 cross-coupled TL NOR gates [10], so it consumes double the power). However, additional inputs impose additional waveguide routing and coupling complexity. Therefore, in our design we limit the number of inputs to no more than 2.

We also include the simulated eye diagram of a TL inverter gate operating at 60 Gbps in Figure 2(c), which shows sufficient eye opening that indicates good signal integrity and reliable operation.

By connecting TL logic gates using waveguides, optical circuits that perform various logic functions can be constructed. In addition to waveguides, the following *passive* components are also used in our TL switch design: (1) *splitters* to split

[1]Note that, static power is the dominant component in the power of TL gates. Thus, unlike CMOS gates, the power of TL gates is roughly constant for different data rates and activity factors.

one optical signal into multiple signals [33], [34], providing circuit fanouts; (2) *combiners* to combine multiple optical signals into one [34]; (3) *waveguide delay elements* to delay the propagation of optical signals for a short amount of time in the order of 100ps [35], [36].

To fabricate TL chips, since the TL's epitaxial structure is very similar to GaAs HBTs with the addition of a Distributed Bragg Reflector (DBR), the standard HBT process flow on GaAs substrates may be adopted with minimal modifications, which include: (1) forming an oxide aperture to provide optical and current confinement [37], [38]; (2) using an Inductively-Coupled Plasma (ICP) etch to process form mesas in the semiconductor DBR region (similar to how DBRs are fabricated in VCSELs). This is how our TL gate prototypes were processed. To form even larger circuits, TL chips can be bonded onto an optical interposer where waveguides and all passive circuit elements reside through hybrid integration [39], similar to [40].

IV. The All-Optical Baldur Network

In this section, we present the details of our Baldur network[2], which consists of 2x2 TL switches to implement a bufferless and clock-less multi-butterfly topology. We expect Baldur to achieve similar results with other multi-stage topologies (e.g., Benes [41], Omega [42], etc.) because many multi-stage networks are largely isomorphic [43].

A. TL Switch Design Challenges

The basic building blocks of Baldur are a group of 2x2 optical switches. Using the TL to design these switches presents unique challenges and thus requires new approaches that are fundamentally different from synchronous electrical logic design. First, typical data encoding schemes such as 8b/10b and 64b/66b [44] require clock and data recovery (CDR) units to first recover the clock signal that is used to generate the packet data, and then recover the actual packet data using each cycle of the recovered clock as a delimiter for each bit. CDR units cannot be efficiently implemented using TL gates, since the power of electrical CDR units is already very high and will be further exacerbated (by >100X) due to the high power of TL gates at the current technology node. The only other option, which is to use electrical CDR units to recover optical data, deteriorates the benefits of all-optical switching. Therefore, a new way to decode data without a clock is required. Second, the lack of an optical clock as well as clocked sequential elements means that the TL switch design must be asynchronous.

B. Our Clock-less Data Encoding/Decoding Approach

To address the challenges discussed in the previous section, we developed a variant of the Digital Pulse Interval Width Modulation (DPIWM) scheme [45], [46] to perform *length-based encoding*, as it allows data decoding without a clock. Note that, our length-based encoding scheme only needs to be applied for the routing bits in the packet, as shown in

[2]Baldur is the god of light in Norse mythology, which is a well-suited name for the first all-optical network.

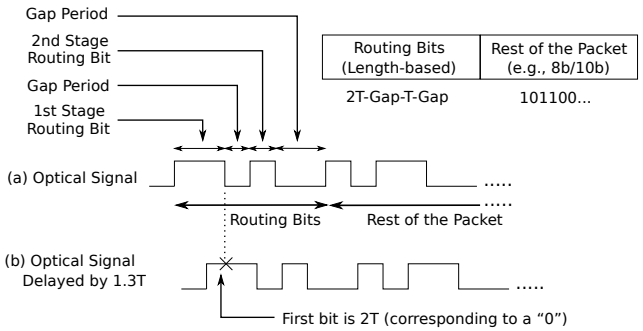


Figure 3: Packet Format, Encoding, and First Bit Decoding in Baldur.

Figure 3. In Baldur, there is 1 routing bit for each stage of the multi-butterfly network. All routing bits are represented by the presence of light, where binary values are encoded into the lengths of optical signals: logic "0" is encoded as two bit periods ($2T$) and logic "1" is encoded as one bit period (T), where T is the clock cycle time. Moreover, the routing bits are separated by "gap periods" represented by the absence of light, such that the total length of each routing bit plus the following gap period is $3T$. This specific uniform length of each routing bit is derived based on our analysis to simplify design logic and also enhance circuit reliability.

Our length-based encoding scheme introduces very minimal bandwidth overhead. For example, if there are 8 routing bits and the rest of the packet is 512 bytes, this scheme would introduce 0.34% overhead compared to the typical 8b/10b encoding.

C. Details of the TL Switch Design

Given network packets whose routing bits are encoded using the length-based scheme, our all-optical TL switch is responsible for obtaining the designated output port of the packet, and either routing the packet to the correct output port or dropping it if the network is congested. Figure 4(a) shows the block diagram of the switch, which includes two main components: the *switch fabric* and the *header processing unit*.

In the **switch fabric**, an input packet is first split into two by an optical splitter (SP0 or SP1): one is sent to the header processing unit to obtain the packet's destination and arbitrate for the corresponding output port, and the other is connected to TL gate AND0 or AND1. Each of the AND gates modifies an incoming packet by AND'ing it with the output of a mask off latch from the header processing unit, so that the first routing bit is masked off. As a result, the next routing bit becomes the first bit at the next stage – in other words, the first routing bit always specifies the destination in the current stage. Masking the routing bits this way simplifies switching logic and enables modular design.

The output of AND0/AND1 is delayed using waveguide delays WD0/WD1 until the arbitration for the designated output port for the corresponding packet is finished. Based on our simulation experiments, the delay of WD0 and WD1 is set to 132 ps. The outputs of WD0 and WD1 are then split

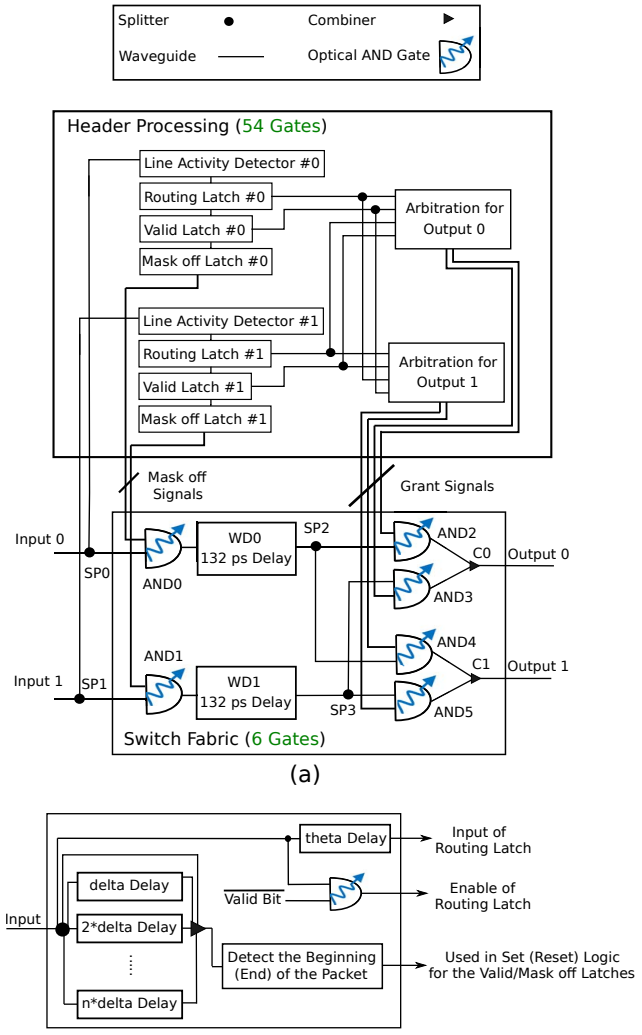


Figure 4: (a) Overview of the All-Optical 2x2 Switch with Multiplicity of 1, consisting of only 60 TL gates. (b) Details of the Line Activity Detector Block in (a).

again using SP2 and SP3, and connected to optical AND gates AND2-AND5 so they can reach either output port depending on the routing decision from the header processing unit. Finally, the outputs of AND2 and AND3 are combined using optical combiner C0, which performs the OR function, to form a multiplexer. Similarly, AND4, AND5, and C1 together form another multiplexer. Four grant signals from the heading processing unit, one for each input/output combinations, act as the select signals for the two multiplexers to forward or drop packets according to arbitration results.

The **header processing unit** is composed of control latches, line activity detectors, and arbitration units.

Control Latches: There are three types of control latches in our design. A *routing latch* stores the decoded routing bit of an incoming packet, and a corresponding *valid latch* indicates if the bit stored in the routing latch is currently valid. Moreover, a *mask off latch* outputs a signal to the switch fabric to mask off the first routing bit in the packet, as discussed above. The values of all these latches are driven by the line activity detectors.

Line Activity Detector: The line activity detector (Figure 4(b)) serves two main functions.

First, it detects the beginning and the end of each packet by continuously detecting the presence of light. In the current design, we assume that 8b/10b encoding is used for the non-routing bits of a packet, which is not allowed to contain more than 5 consecutive 0's[3]. Given this constraint and to achieve a low error probability of 10^{-9} in the presence of variations and timing jitters (see Sec. IV-F), in Baldur we specify that the absence of light for a period longer than $6T$ means that there is no in-flight packet currently. To detect the presence of light, the input signal is split into multiple paths that connect to multiple delay elements with different delay values, and the outputs of all delay elements along with the original input are connected using an optical combiner as shown in Figure 4(b). The output of the combiner is "1" if and only if at least one of its inputs is "1"; therefore, it becomes "1" as soon as the presence of light is detected at the beginning of a packet, and stays "1" until the end of the packet, which is $6T$ time period after the falling edge of the last "1" bit in the packet. The "0" to "1" and "1" to "0" transitions in the combiner output are used to signify the beginning and the end of a packet, respectively. To detect these transitions, we delay the combiner output by a short amount of time ($0.5T$ in our implementation), and compare the delayed signal with the original signal: if the delayed signal is "0" and the original signal is "1", then there is a "0" to "1" transition; and vice versa for a "1" to "0" transition.

Both the valid and mask off latches are set at $2.5T$ time period after the beginning of a packet, and are reset at the end of the packet[4]. This way, their values become "1" during the gap period that corresponds to the first routing bit, and remain "1" until the end of the packet. This is the intended behavior since the routing bit is indeed valid during the entire duration when the valid bit is "1". Moreover, the mask off bit is "1" before the beginning of the second bit, so only the first bit of the packet is masked off.

Second, the line activity detector decodes the routing bit of its input packet for the current stage – i.e., the first bit of the packet, and stores it in a routing latch. Figure 3 illustrates how the first bit of a packet is obtained in our design: we delay the input signal by $1.3T$ and measure the delayed signal at the falling edge of the first bit (i.e., before its gap period). If the value is 1, it means that the length of the first bit is $2T$ (corresponding to a "0"); otherwise the length is T (corresponding to a "1"). Figure 4(b) shows how this procedure is implemented in the line activity detector to

[3]We use 8b/10b as an example. Our design can be easily modified to work with other encoding schemes.

[4]In Baldur with multiplicity of 1, the behavior of the valid and mask off latches is the same. However, this is not the case when multiplicity (m) is greater than 1, because for each input, m valid latches are required for m paths, but the number of the mask off and routing latches stays 1 regardless of the multiplicity value.

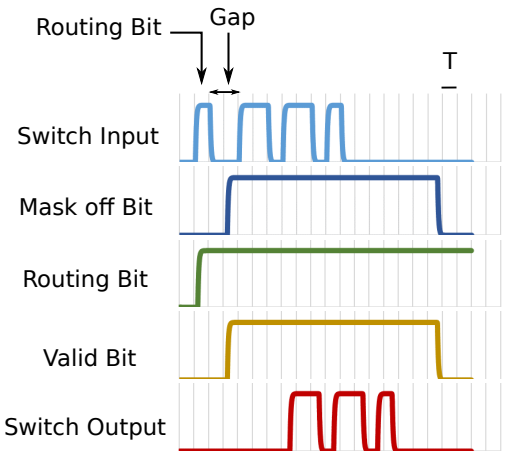


Figure 5: HSPICE Simulation Results.

Table V: Path Multiplicity and Drop Rate Results in Baldur.

Multiplicity	1	2	3	4	5
Gates per Switch	64	300	642	1,112	1,710
Switch Latency (ns)	0.14	0.49	0.94	1.5	2.25
Drop Rate (%)	65.3	21.5	3.2	0.3	0.02

generate the appropriate data input and enable signals for a routing latch.

Arbitration unit: After the routing bit of a packet is stored in the routing latch and the valid bit becomes "1", the packet is ready to participate in the arbitration process, and the arbitration result (i.e., the grant signals) is sent to the switch fabric to forward the packet to its designated output port if the packet wins the arbitration (the packet will be dropped otherwise). The design of our arbitration unit is a 2x2 asynchronous arbiter built using a latch and two threshold NOT gates, similar to [47]. It guarantees that for each output, at most one packet wins the arbitration at any given time.

D. Circuit Simulation Results

We model TL gates using the results reported in Table IV, and simulate the 2x2 TL switch using Synopsys's HSPICE tool. Figure 5 depicts the simulation waveform for a scenario where the designated output port is available when a packet arrives at the input. As shown in the waveform, the routing bit is stored in the routing latch before the falling edge of the routing bit. Also, both the valid and mask off bits become "1" during the gap period of the first routing bit and stays "1" until the end of the packet. Therefore, the packet traverses from its input to the designated output port as expected. The waveform also shows that the first routing bit in the packet is correctly masked off before the packet arrives at the switch output.

E. Minimizing and Handling Packet Drops

We enhance our design by increasing path multiplicity and randomness to decrease the number of packet drops substantially. Multiplicity of m in a 2x2 switch is achieved by having $2m$ output ports (m output ports per each output direction), and $2m$ input ports (m input ports per each input direction). Thus, multiplicity of greater than 1 provides more than one path to deliver a packet to the designated output direction. To implement a 2x2 switch with multiplicity of m, we augment the TL switch with multiplicity of 1 (as shown in Figure 4) such that: (1) $2m$ input packets can be processed independently (which is achieved by increasing the number of paths in the switch fabric and the number of control latches, line activity detectors, and arbitration units in the header processing unit); and (2) an input packet can be transmitted successfully as long as at least one of the m paths is available (which is achieved by checking the availability of each path sequentially in the arbitration units).

We evaluate the design complexity, latency, and packet drop rate for Baldur with different path multiplicity values in a 1,024-node network, and the results are shown in Table V (drop rates are obtained using the CODES simulator for the transpose traffic pattern under 0.7 input load; see methodology details in Sec. V-A). We can see that, even with a high multiplicity of 4 or 5, the latency/area/power costs are still minimal, but the packet drop rate is reduced substantially. Therefore, incorporating path multiplicity results in a good trade-off for Baldur. Moreover, if we view a radix-2 multi-stage network as a sorting network that narrows a packet's possible destination by a factor of 2 at each stage, the effectiveness of multiplicity can be improved by connecting the switch outputs that belong to the same direction to random switches in the appropriate sorting group in the next stage. This randomization leads to a theoretical property called "expansion" [14]. Networks with expansion, including Baldur, are immune to worst-case permutations [19].

For large-scale networks, detailed network simulations cannot be easily performed; therefore, we determine the multiplicity value required for Baldur to achieve a low (less than 1%) packet drop rate as follows: given a network scale (i.e., the number of server nodes), we consider the worst-case scenario where one packet per server node is injected to the network and all the packets arrive at the first stage of the network at the same time, and simulate this scenario using an in-house tool to obtain the packet drop rate corresponding to different multiplicity values for various traffic patterns. Our results show that multiplicity of 4 is required for a 1,024-node network (which is consistent with our detailed simulation results reported in Table V), and multiplicity of 5 is sufficient for networks with over 1 million nodes.

In addition to incorporating multiplicity, we also implement the binary exponential backoff (BEB) mechanism [48] to throttle the transmitter nodes when the network starts to experience an increasing number of packet drops to further reduce drop rate.

In the rare cases that packets are dropped, packet re-transmissions are handled by the server nodes. When a transmitter node sends a packet, it waits for an ACK from the receiver. If it does not receive the ACK after a timeout period based on its own local timer (because either the packet itself or the ACK is dropped), the transmitter node re-transmits the packet. To support packet re-transmission, a small buffer is required for each node to hold all outgoing packets that have

not been ACK'ed. Based on our simulation results (details in Sec. V), since the packet drop rate is less than 1%, a buffer of 536 KB per node is sufficient for a wide variety of traffic patterns under a heavy input load of 0.7, and we use 1 MB buffers in our design to allow abundant margins. This re-transmission mechanism may be implemented in either software or hardware, depending on the protocols supported at the server nodes.

F. Reliability of Baldur

Reliability is a critical consideration in network switches. In TL switches, optical signal timing and amplitude are two major factors that can affect the correctness of the switch. However, since a TL gate naturally restores signal strength as discussed in Sec. III, we focus on validating that TL switches can achieve reliable operations in the presence of timing variations and jitters.

Specifically, we consider 10% variation in the delay and rise/fall time of TL gates and 1 ps variation in the waveguide delay elements, and manually verify that, in the presence of these variations, our design can tolerate up to *0.42T* change (in either direction) in the bit length of any routing bit. Given this result, if we model timing jitter (in ps) at each transition of the packet signal as a random variable that follows a Gaussian distribution with $\mu = 0$ [49] and a variance of 1.53, the major error scenarios in the TL switch design, which are listed below, would only occur at a low error probability of 10^{-9}: (1) the length of a routing bit was originally *2T* (*T*), but it is incorrectly stored as *T* (*2T*); (2) the valid bit becomes high (low) when the routing bit is invalid (valid); (3) the mask off bit is latched incorrectly; (4) the line activity detector fails to detect the presence/absence of network packets correctly. This analysis shows that our TL switch design is equipped with adequate design margins to perform reliable operations.

However, in the case that an error is detected in Baldur, diagnosis support is provided so that the error can be isolated to a single 2x2 TL switch, and appropriate repair actions can be taken. In Baldur with multiplicity of 1, the path that each packet traverses to reach its destination is deterministic. Thus, a faulty switch can be identified using a sufficiently-large number of packets. In Baldur with multiplicity of greater than 1, each switch can be configured such that only one output port is enabled at a time to achieve deterministic routing (so the testing procedure is similar to that for Baldur with multiplicity of 1). This is done by incorporating additional test signals (driven by the server nodes) in Baldur to block all output ports except one in all of the 2x2 TL switches.

G. Physical Link Interfaces and Packaging

The Baldur network is physically built using a 2D array of optical interposers [40] integrated on one or more printed circuit boards (PCBs) as shown in Figure 1. To simplify the connectivity requirements between different interposers, we constrain each interposer column to contain only 1 stage of the multi-butterfly network. Connections inside each interposer are formed using waveguides, and interposers in adjacent columns are connected using fiber array units (FAUs) (e.g., [50]). To connect server nodes to the Baldur network, optical fibers with LC connectors can be used. Since there are a large number of input/output ports in Baldur, a fiber management system is included, where rack-mount fiber enclosures and cassettes (RFEC) [51] are responsible for connecting the fibers from/to server nodes with dense FAUs, which are then coupled into the first/last column of the optical interposers. At the server nodes, either existing optical transceivers or TL-based lasers/photodetectors may be used, as long as the wavelength supported is consistent for the whole network.

The number of cabinets required to hold the entire Baldur network is 1 for the 1,024-node scale, and 752 (which is 3.2% of the total number of cabinets) for the 1M-node scale, assuming that the standard 60.96 cm x 45.72 cm PCBs, 32 mm x 10 mm interposers, and commercially-available FAUs, fiber management systems, and cabinets are used. These results are calculated under both fiber pitch (127 μm [50]) and power/thermal (no more than 85 kW per cabinet [1]) constraints, with the fiber pitch being the limiting factor (e.g., if the 85 kW peak power per cabinet is the only constraint, then only 176 cabinets are needed for the 1M-node scale). Fiber pitch may reduce in the future (as this is an active research area [52]), which will in turn reduce the hardware requirements for Baldur as well. Moreover, note that the TL gates occupy only a very small portion of the interposer area (e.g., <10% for a 1,024-node network with multiplicity of 4), leaving abundant space for waveguides and other passive optical elements. This result suggests that the area of Baldur is insensitive to the area of TL gates.

V. Performance Evaluation

A. Methodology

We evaluate the performance of a 1,024-node Baldur network with multiplicity of 4 (which achieves a packet drop rate of <1% as discussed in Sec. IV-E) using the CODES toolkit to perform packet-level network simulations [53]. In our evaluation, we quantitatively compare Baldur with the following representative networks: (1) an electrical multi-butterfly network with multiplicity of 4; (2) a dragonfly network constructed using the most optimized architecture recommended in [16]; (3) a 3-level fat-tree network with full bisection bandwidth as proposed in [17]; (4) an ideal network with infinite bandwidth and a flat packet latency of 200 ns. Our simulation configurations and parameters are summarized in Table VI.

In terms of workloads, we simulate a wide variety of synthetic traffic patterns as well as real HPC workload traces:

- *Random_Permutation:* server nodes are paired for packet transmission using a random permutation.
- *Transpose:* a server node with binary address $a_n a_{n-1}..a_{\frac{n}{2}}a_{\frac{n}{2}-1}...a_1a_0$ sends packets to the server node with address $a_{\frac{n}{2}-1}...a_1a_0a_na_{n-1}..a_{\frac{n}{2}}$.
- *Bisection:* half of the server nodes are paired with the other half for packet transmission using a random permutation.
- *Group_Permutation:* first, in dragonfly, the groups are paired using a random permutation, and each server node

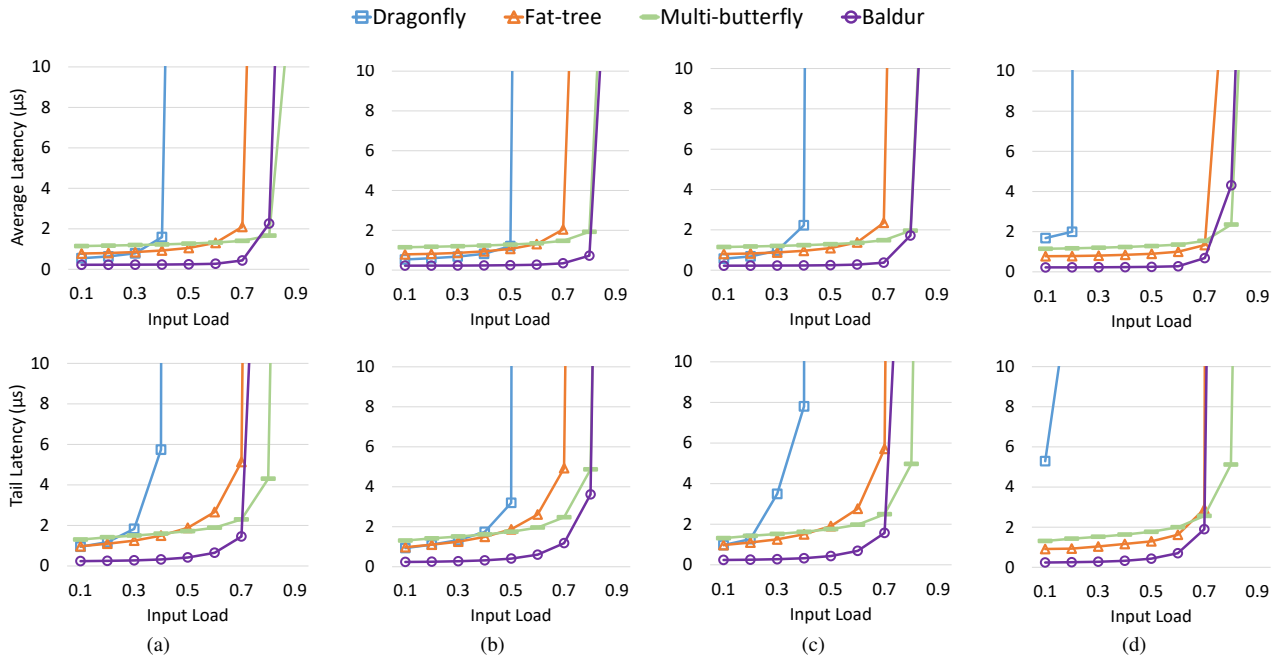


Figure 6: Average (Top) and Tail (Bottom) Packet Latency Results. (a) Random_Permutation. (b) Transpose. (c) Bisection. and (d) Group_Permutation.

Table VI: Simulation Configurations and Parameters. Link delay values are derived based on the length of the links, as well as optical fiber and copper wire propagation delay values.

Baldur		
Multiplicity	Switch Latency (ns)	Link Delay (ns)
4	1.5 (based on Table V)	100
Electrical Multi-butterfly (24 KB buffer per port, 3 virtual channels, radix 2)		
Multiplicity	Switch Latency (ns)	Link Delay (ns)
4	90 [54]	100
Dragonfly (24 KB buffer per port, 3 virtual channels)		
Routing Policy	Switch Latency (ns)	Link Delay (ns)
Adaptive [16]	90 [54]	Intra-group: 10 Inter-group: 100
Fat-tree (24 KB buffer per port, 3 virtual channels)		
Routing Policy	Switch Latency (ns)	Link Delay (ns)
Adaptive [55]	90 [54]	Level1: 10 Level2: 50 Level3: 100
Ideal (Infinite Bandwidth; Packet Latency = 200 ns)		

sends packets to another randomly-chosen node in the partner group. Then, the same transmitter/receiver node pairs are applied to all other networks.

- *Hotspot:* all server nodes send packets to one specific destination node.
- *Ping_Pong1:* server nodes are randomly paired. Each node sends a packet to its partner node, wait until it receives a packet from the partner node, and then immediately sends the next packet.
- *Ping_Pong2:* similar to ping_pong1 but with a different pairing algorithm. First, in dragonfly, the nodes from one group (e.g., group A) are paired with nodes from another group (e.g., group B). Then, the same transmitter/receiver node pairs are applied to all other networks.
- Four HPC workloads from the Design Forward project [56]. The communication traces are collected using the DUMPI framework [57].

In our simulation experiments, we vary the input load, which is defined as the percentage of time that the transmitter is busy transmitting packets. For each synthetic traffic pattern and input load value, each server node injects 10,000 packets to the network. For all synthetic traffic patterns except the two ping_pong patterns (in which packets are sent back-and-forth sequentially between paired nodes without any idle periods), the time interval between two consecutive packets is determined based on an exponential distribution, and the mean value of the time interval is defined as:

$$mean\_value\_of\_time\_interval = (packet\_size)/(input\_load * link\_data\_rate) \quad (1)$$

packet_size is set to 512 bytes [53] and *link_data_rate* is set to 25 Gbps (which is the maximum data rate per lane in current standards) in our experiments.

B. Performance Results

In this section, we present the network performance results obtained from our simulation experiments. Note that, the results for Baldur account for all latency overheads due to packet drops and re-transmissions.

Our results are depicted in Figure 6 and 7. Figure 6 shows the average and tail (99th-percentile) packet latency

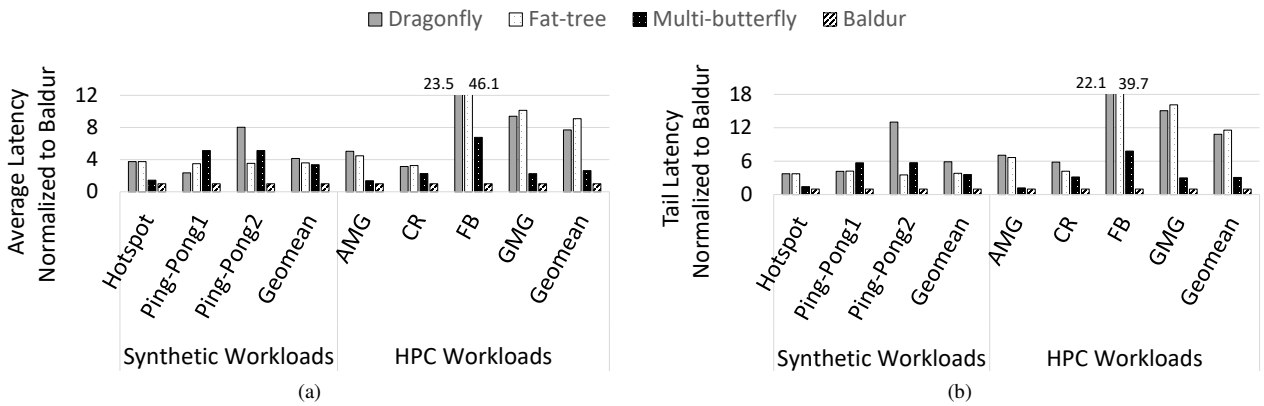


Figure 7: Normalized Average Packet Latency (a) and Normalized Tail Packet Latency (b) for Synthetic and HPC Workloads. Input load is 0.7 for hotspot (latency results are similar for other load values), and is not applicable for other workloads.

results for various networks running the random_permutation, transpose, bisection, and group_permutation synthetic traffic patterns. The advantage of the multi-butterfly topology is clear: both Baldur and electrical multi-butterfly achieve much better average and tail latency results, and they saturate at higher input loads compared to dragonfly and fat-tree. For input loads that are less than or equal to 0.7, Baldur achieves the lowest average (tail) latency among all networks – e.g., 1.9-6.3X (1.5-4.2X) vs. fat-tree, 1000-3000X (1000-2000X) vs. dragonfly, and 2.2-4.3X (1.3-2.1X) vs. electrical multi-butterfly when the load is 0.7. Although they have the same topology, Baldur outperforms electrical multi-butterfly because packet processing/switching in Baldur is performed in the ultra-fast optical domain. When the input load is high (>0.7), Baldur still achieves the best results in almost all cases. The exception is that the tail latency of electrical multi-butterfly is lower than Baldur in some of the traffic patterns, because in Baldur some packets are dropped and re-transmitted. Note that, although electrical multi-butterfly may achieve good performance results, it is highly expensive and thus impractical at large scales (see Sec. VI).

When comparing Baldur with the ideal network, Baldur's average packet latency is only 1.7X-3.4X higher. The difference is mostly due to packet drops. Moreover, Baldur's total switch latency (1.5 ns per stage) is much smaller than the total link latency (200 ns or 100 ns per input/output link, which is equal to the packet latency in the ideal network). Thus, we expect that Baldur's packet latency will approach the latency of the ideal network as path multiplicity increases, because packet drop rate will decrease as a result.

In Figure 7, we show the average and tail latency results for the rest of the workloads: hotspot, ping_pong1, ping_pong2, and the HPC workloads. Baldur achieves the best performance results for all synthetic traffic patterns – for example, the Geomean of Baldur's average (tail) latency is 3.4X-4.1X (3.6X-5.9X) lower than other networks. For hotspot, which results in high network congestion, path multiplicity in both Baldur and electrical multi-butterfly effectively alleviates the congestion, thereby achieving better latency results than dragonfly and fat-tree. For ping_pong1 and ping_pong2, where the impact of packet latency on the overall network performance is emphasized by serialization dependency between server nodes, we see that Baldur achieves significantly better results due to its ultra-low switch latency, while the performance of all other networks is limited due to slow electrical header processing and switching. Dragonfly's packet latency values are particularly high for ping_pong2, because ping_pong2 is constructed in a way that forces dragonfly to experience high levels of congestion between two groups, where the network bandwidth is limited.

For the HPC workloads, Baldur also achieves the best results: the Geomean of its average (tail) latency is 2.6X-9.1X (3.1X-11.6X) better compared to other networks. Moreover, both dragonfly and fat-tree suffer from very high average latency in certain workloads. For example, in *FB*, the average (tail) latency of dragonfly/fat-tree is 23.5X/46.1X (22.1X/39.7X) higher than Baldur. These results suggest that, unlike Baldur, dragonfly and fat-tree are not immune to worst-case traffic patterns.

VI. Power, Cost, and Scalability Analysis

A. Power Analysis and Results

In this section, we report the total power of Baldur, and compare it with the power of dragonfly, fat-tree, and electrical multi-butterfly.

1) Methodology: We estimate the power of a network by summing up the power of the following major components:

1. Optical transceivers. The power consumption of Cisco's SFP28 modules [58] (1.5 W) is used in our calculations for each optical transceiver.

2. SerDes units. We use the power number reported in [59], which is 0.693 W, for each SerDes unit.

3. Re-transmission buffers, which consume 0.741 W (for 1 MB) per node [60] and is only required in Baldur (assuming that packet re-transmission is handled in hardware).

4. Network switches. For Baldur, the power of each TL switch is the product of the total number of TL gates per switch and the power of a TL gate (0.406 mW as reported in Sec. III). For other networks, we use ORION 3.0 [61] and Cacti 6.5 [62] to obtain the power associated with virtual

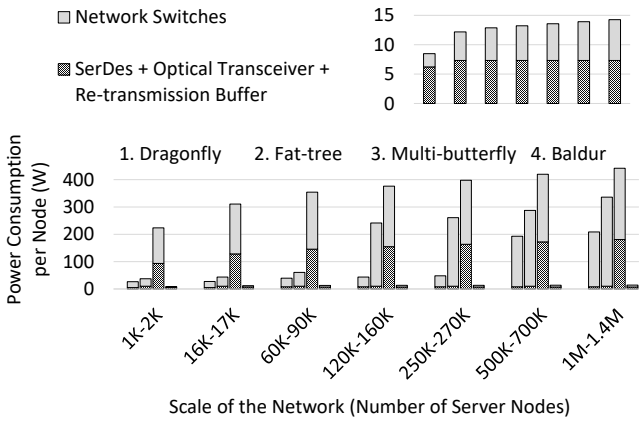


Figure 8: Power Consumption per Server Node in Various Network Scales. The small figure (top right) shows a zoom-in view of Baldur's power results for all scales.

channel allocation, switch allocation, crossbar, clocking, and buffering. Note that, the results in dragonfly and fat-tree are optimistic since the power cost of the adaptive routing logic is not included.

To see how total network power changes as network size increases, we scale the number of servers from 1,024 to over 1M (where 1M is the expected scale for exascale computing). Our methodology ensures that all networks are optimized for a given scale. For example, in dragonfly, starting from ~83K server nodes it is not efficient to use electrical links for intra-group connections any more (because of the long distance between the switches within a group). Therefore, we report power numbers assuming that optical links are also used for intra-group connections for network scales greater than or equal to 83K. Scaling dragonfly this way results in far less power (e.g., by 23.45X) than keeping the group size small but increasing the number of groups for a network with ~1.3M nodes.

Note that, the number of server nodes in each scale is not exactly the same in different networks due to the particular construction of the different topologies (for example, the number of nodes is always a power of two in Baldur and multi-butterfly, but this is not the case in dragonfly and fat-tree). Thus, when we present the power results, at each scale we show a range that includes the number of servers in all the networks.

2) Results: Figure 8 shows the total power consumption per server node in different network scales. Baldur achieves the best scalability, since its power consumption (per server) at the *1M-1.4M* scale is only increased by 1.7X compared to the *1K-2K* scale. On the other hand, between these two network scales, the power per server node is increased by 7.8X, 9.0X, and 2.0X, respectively, for dragonfly, fat-tree, and electrical multi-butterfly. The large increase in power for dragonfly and fat-tree limits the scalability of these two networks, and is a result of the significantly larger switch radix, which changes from 16 to 96 and 16 to 160, respectively. Electrical multi-butterfly shows the smallest increase in power among the three electrical networks because

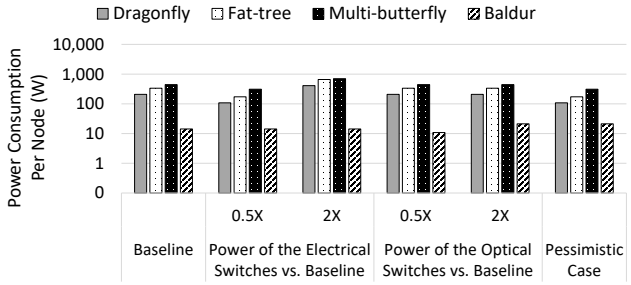


Figure 9: Sensitivity Analysis on the Power of Electrical and Optical Switches for the *1M-1.4M* Network Scale. "Baseline": same results as Figure 8 for this scale. "Pessimistic Case": the power of electrical switches is reduced by 50%, and the power of optical switches is increased by 2X.

its scalability is not limited by switch radix; however, it consumes more power than the other two networks in all scales.

Compared with other networks, Baldur consumes less power than all other networks at all scales. In particular, although the number of switches is the same for both electrical multi-butterfly, each switch in Baldur consumes significantly less power (e.g., 96.6X less at the *1K-2K* scale), which is due to various optimizations including simpler control logic, and the elimination of optical transceivers, SerDes units, packet buffering, clocking, and so on. Moreover, as we scale the network, in general the power benefit of Baldur over other networks increases: Baldur reduces network power by 3.2X-26.4X at the *1K-2K* scale, and 14.6X-31.0X at the *1M-1.4M* scale (note that, the power benefit of Baldur decreases slightly as we scale the network from *1K-2K* to *16K-17K* because the multiplicity is increased from 4 to 5).

3) Sensitivity Analysis: We perform sensitivity analysis by scaling the power of network switches by 0.5X and 2X (for both electrical and optical switches) to account for possible sources of inaccuracy in technology parameters and network power modeling, and the results for a *1M-1.4M* scale network are shown in Figure 9. The observation is that, even considering the pessimistic case, Baldur is still the most power efficient, consuming 5.1X, 8.2X and 14.7X less power than dragonfly, fat-tree and electrical multi-butterfly, respectively.

In summary, thanks to its high scalability and low power consumption, Baldur is a promising network for exascale computing.

B. Cost Analysis and Results in Various Network Scales

To estimate the cost of deploying Baldur in practice, we adopt a cost model which takes into account the cost of fibers, FAUs, RFECs, optical interposers, and optical transceivers, similar to previous work [2], [63]. Furthermore, we pessimistically assume that the cost of optical interposers (including the TL chips and other passive optical devices) is 5x more than the cost of current CMOS chips for the same area.

As shown in Figure 10, Baldur achieves low cost (in terms of USD per server node) in various network scales.

For example, at the *1K-2K* scale, Baldur's cost per node is 523 USD, compared to 1,992 USD for a fat-tree network with 2,560 nodes [17], [63]. The cost of Baldur increases only slightly as the number of server nodes increases, which suggests that Baldur is highly scalable (this is consistent with our power analysis). Moreover, the cost of optical interposers dominates the total cost. Thus, the cost of Baldur may further decrease since the number of optical interposers may become smaller as a result of reduced fiber pitch in the future (as discussed in Sec. IV-G).

VII. Related Work

In this section, we discuss various optical network and optical logic proposals in the literature, and compare these proposals with Baldur.

Optical packet switching (OPS): OPS performs optical switching at the packet level [4], [5]. A popular example of OPS is based on AWGRs [3], [5], and an overview of AWGR networks can be found in Sec. II. In this section, we focus on a quantitative comparison between Baldur and an AWGR network for the scale of 32 nodes (and we expect the general conclusion to hold for other network scales and OPS schemes as well). For this scale, multiplicity of 3 is sufficient for Baldur to achieve a packet drop rate of <1%. The AWGR network is built using a 32-radix AWGR [3], which is capable of sending up to 3 packets to each output port in parallel by using 3 different wavelengths. Note that, although in theory 32 wavelengths are available in a 32-radix AWGR, a much smaller number of wavelengths should be used instead due to cost and practical considerations [3].

In terms of performance, both Baldur and the AWGR network provide the same bandwidth, but the AWGR network needs to pay a large latency overhead due to electrical header processing (e.g., 90ns [54], much higher than the switching latency in Baldur). In terms of power, excluding the power cost of optical transceivers and SerDes units at the server nodes (which is the same for both networks), Baldur consumes 0.7 W per node (which accounts for the power of TL chips), while the AWGR network consumes 4.2 W per node (which accounts for optical receivers, SerDes units, buffers to support electrical header processing, and tunable wavelength converters). Therefore, Baldur is superior to the AWGR network in terms of both latency and power. In terms of scalability, unlike Baldur which is highly scalable, the scalability of AWGR networks is limited by the switch radix as discussed in Sec. II. Moreover, as the network size increases, the number of wavelengths required in AWGR network increases, which may significantly increase the power/cost/complexity of the optical transceivers [24].

Optical circuit switching (OCS) and hybrid OCS/Electrical switching (ES): MEMS-based OCS relies on mirrors to perform switching [64], and a separate processing unit is responsible for establishing the correct paths between input and output ports by repositioning the mirrors. The main disadvantage of OCS is that the switching time is long – in the order of milliseconds. In the hybrid OCS/ES schemes [2], [63], ES is used to deliver packets with low latency, and OCS is used to provide high bandwidth transmission. Compared to these OCS and hybrid OCS/ES schemes, Baldur has the following advantages: (1) when ES is used, Baldur has clear packet latency and power benefits as shown in Sec. V and VI; (2) when OCS is used, Baldur is expected to achieve similar performance and bandwidth due to its ultra-low packet latency; (3) in hybrid OCS/ES schemes, complex control logic is required to decide when to use OCS and ES. In contrast, the routing and control complexity in Baldur is low; (4) Baldur achieves lower cost than a OCS-based scheme (e.g., 523 USD per node for Baldur (see Sec. VI) vs. 1,719 USD per node for OCS [63] for the scale of a few thousand nodes).

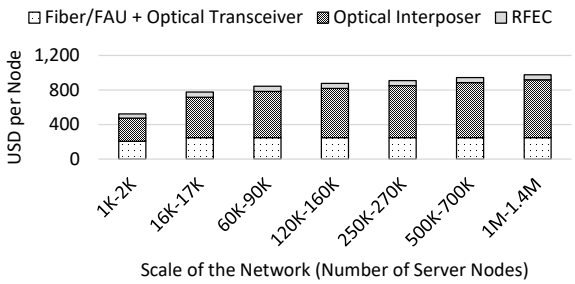


Figure 10: Cost of Baldur in Various Network Scales.

Optical logic: The TL is fundamentally different from other optical logic technologies (e.g., [65], [66]). First, it is a unique technology that integrates the functionalities of a transistor, a laser, and a photodetector in a single compact device. Second, it is possible to directly interface TL devices and logic gates with circuits in the electrical domain. Last but not least, the TL technology has been demonstrated to be efficient, fast, and reliable through actual fabricated prototypes, unlike several previously proposed optical technologies (e.g., [66]) that are impractical. Therefore, the TL serves as a solid and promising technology basis to develop all-optical networks and computing systems.

.

VIII. Conclusions and Future Work

In this paper, an all-optical network called Baldur is introduced. Baldur is able to achieve significant improvements in network latency, power, and cost compared to state-of-the-art networks. Baldur is also highly scalable to support exascale computing. These benefits are obtained by fundamentally understanding the TL technology and its architectural implications, which leads us to non-conventional design decisions and new design techniques that are optimized for the Baldur network.

In the future, we envision that the TL will drive further architectural innovations and enable new network capabilities. Some examples include in-flight routing for >100 G links, and host-based and in-network accelerations (such as network filtering for security purposes, and traffic combining to improve performance). We also plan to apply TL-based optical computing in other application domains (e.g., machine learning) that demand ultra-high-speed and efficient processing.

IX. ACKNOWLEDGEMENTS

We thank Prof. Andrew A. Chien and Chris Jones of the University of Chicago, Misbah Mubarak of Argonne National Laboratory, and anonymous reviewers for their help and comments. This work is sponsored in part by NSF grants 1640192 and 1405959, and also E2CDA-NRI, a funded center of NRI, a Semiconductor Research Corporation (SRC) program sponsored by NERC and NIST.

REFERENCES

[1] G. Pautsch, D. Roweth, and S. Schroeder, "The cray® XC™ supercomputer series: Energy-efficient computing," tech. rep., Technical Report, 2013.

[2] N. Farrington *et al.*, "Helios: A hybrid electrical/optical switch architecture for modular data centers," in *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, (New York, NY, USA), pp. 339–350, ACM, 2010.

[3] X. Ye *et al.*, "Dos: A scalable optical switch for datacenters," in *Proceedings of the 6th ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, p. 24, ACM, 2010.

[4] O. Liboiron-Ladouceur *et al.*, "The data vortex optical packet switched interconnection network," *Journal of Lightwave Technology*, vol. 26, no. 13, pp. 1777–1789, 2008.

[5] K. Xi, Y.-H. Kao, and H. J. Chao, "A petabit bufferless optical switch for data center networks," in *Optical interconnects for future data center networks*, pp. 135–154, Springer, 2013.

[6] G. Walter, N. H. Jr., M. Feng, and R. Chan, "Laser operation of a heterojunction bipolar light-emitting transistor," *Applied Physics Letters*, vol. 85, no. 20, pp. 4768–4770, 2004.

[7] M. R. Jokar, L. Zhang, J. M. Dallesasse, F. T. Chong, and Y. Li, "Direct-modulated optical networks for interposer systems," in *Proceedings of the 13th IEEE/ACM International Symposium on Networks-on-Chip*, p. 10, ACM, 2019.

[8] M. R. Jokar *et al.*, "A high-performance and energy-efficient optical network using transistor laser," in *TECHCON*, 2019.

[9] M. R. Jokar, L. Zhang, Y. Li, and F. T. Chong, "Investigating energy-efficient technologies for next-generation optical interconnection networks," in *TECHCON*, 2017.

[10] M. Feng, H. W. Then, and N. H. Jr., "Transistor laser optical nor gate for high speed optical logic processors," *GOMACTech*, 2017.

[11] A. Winoto, J. Qiu, D. Wu, and M. Feng, "Transistor laser integrated photonics for optical logics," *IEEE Nanotechnology*, 2018.

[12] S. Aleksic, "Optical burst-switched wdm networks with channel bonding," in *Communication Systems, Networks and Digital Signal Processing, 2008. CNSDSP 2008. 6th International Symposium on*, pp. 405–409, IEEE, 2008.

[13] V. Eramo and M. Listanti, "Power consumption in bufferless optical packet switches in soa technology," *Journal of Optical Communications and Networking*, vol. 1, no. 3, pp. B15–B29, 2009.

[14] F. T. Chong, E. A. Brewer, F. T. Leighton, and T. F. Knight, "Building a better butterfly: the multiplexed metabutterfly," in *Parallel Architectures, Algorithms and Networks, 1994.(ISPAN), International Symposium on*, pp. 65–72, IEEE, 1994.

[15] A. V. Goldberg, B. M. Maggs, and S. A. Plotkin, "A parallel algorithm for reconfiguring a multibutterfly network with faulty switches," *IEEE Transactions on Computers*, vol. 43, no. 3, pp. 321–326, 1994.

[16] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable dragonfly topology," in *Computer Architecture, 2008. ISCA'08. 35th International Symposium on*, pp. 77–88, IEEE, 2008.

[17] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *ACM SIGCOMM Computer Communication Review*, vol. 38, pp. 63–74, ACM, 2008.

[18] E. Upfal, "An O (log N) deterministic packet-routing scheme," *Journal of the ACM (JACM)*, vol. 39, no. 1, pp. 55–70, 1992.

[19] F. T. Leighton and B. M. Maggs, "Fast algorithms for routing around faults in multibutterflies and randomly-wired splitter networks," *IEEE Transactions on Computers*, vol. 41, no. 5, pp. 578–587, 1992.

[20] H. Simon, "Exascale challenges for the computational science community," *Lawrence Berkeley National Laboratory and UC Berkeley, Tech. Rep*, 2010.

[21] N. Binkert *et al.*, "The role of optics in future high radix switch design," in *ACM SIGARCH Computer Architecture News*, vol. 39, pp. 437–448, ACM, 2011.

[22] D. Alistarh *et al.*, "A high-radix, low-latency optical switch for data centers," in *ACM SIGCOMM Computer Communication Review*, vol. 45, pp. 367–368, ACM, 2015.

[23] J. Gripp, J. Simsarian, J. LeGrange, P. Bernasconi, and D. Neilson, "Photonic terabit routers: The iris project," in *Optical Fiber Communication Conference*, p. OThP3, Optical Society of America, 2010.

[24] R. Proietti, Z. Cao, C. J. Nitta, Y. Li, and S. B. Yoo, "A scalable, low-latency, high-throughput, optical interconnect architecture based on arrayed waveguide grating routers," *Journal of Lightwave Technology*, vol. 33, no. 4, pp. 911–920, 2015.

[25] N. E. Jerger, T. Krishna, and L.-S. Peh, "On-chip networks," *Synthesis Lectures on Computer Architecture*, vol. 12, no. 3, pp. 1–210, 2017.

[26] T. Moscibroda and O. Mutlu, "A case for bufferless routing in on-chip networks," *ACM SIGARCH Computer Architecture News*, vol. 37, no. 3, pp. 196–207, 2009.

[27] P. Lotfi-Kamran, M. Modarressi, and H. Sarbazi-Azad, "An efficient hybrid-switched network-on-chip for chip multiprocessors," *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1656–1662, 2015.

[28] C. Wang, M. Liu, M. Feng, and N. Holonyak Jr, "Microwave extraction method of radiative recombination and photon lifetimes up to 85° c on 50 gb/s oxide-vertical cavity surface emitting laser," *Journal of Applied Physics*, vol. 120, no. 22, p. 223103, 2016.

[29] M. Feng, H. Then, N. Holonyak Jr, G. Walter, and A. James, "Resonance-free frequency response of a semiconductor laser," *Applied Physics Letters*, vol. 95, no. 3, p. 033509, 2009.

[30] C. Wu, F. Tan, M. Wu, M. Feng, and N. Holonyak, "The effect of microcavity laser recombination lifetime on microwave bandwidth and eye-diagram signal integrity," *Journal of Applied Physics*, vol. 109, no. 5, p. 053112, 2011.

[31] H. W. Then, M. Feng, and N. Holonyak, "The transistor laser: Theory and experiment," *Proceedings of the IEEE*, vol. 101, no. 10, pp. 2271–2298, 2013.

[32] M. Feng, J. Qiu, and N. Holonyak, "Tunneling modulation of transistor lasers: Theory and experiment," *IEEE Journal of Quantum Electronics*, 2018.

[33] Y. Sakamaki, T. Saida, M. Tamura, T. Hashimoto, and H. Takahashi, "Low-loss y-branch waveguides designed by wavefront matching method and their application to a compact 1 x 32 splitter," *Electronics Letters*, vol. 43, no. 4, pp. 217–219, 2007.

[34] L. B. Soldano and E. C. Pennings, "Optical multi-mode interference devices based on self-imaging: principles and applications," *Journal of lightwave technology*, vol. 13, no. 4, pp. 615–627, 1995.

[35] H. Lee, T. Chen, J. Li, O. Painter, and K. J. Vahala, "Ultra-low-loss optical delay line on a silicon chip," *Nature communications*, vol. 3, p. 867, 2012.

[36] M. Povinelli, S. G. Johnson, and J. Joannopoulos, "Slow-light, band-edge waveguides for tunable time delays," *Optics Express*, vol. 13, no. 18, pp. 7145–7159, 2005.

[37] J. M. Dallesasse and N. Holonyak Jr, "Oxidation of al-bearing III-V materials: A review of key progress," *Journal of Applied Physics*, vol. 113, no. 5, p. 5, 2013.

[38] J. M. Dallesasse and D. G. Deppe, "III–V oxidation: discoveries and applications in vertical-cavity surface-emitting lasers," *Proceedings of the IEEE*, vol. 101, no. 10, pp. 2234–2242, 2013.

[39] J. A. Carlson, C. G. Williams, M. Ganjoo, and J. M. Dallesasse, "Epitaxial bonding and transfer processes for large-scale heterogeneously integrated electronic-photonic circuitry," *Journal of The Electrochemical Society*, vol. 166, no. 1, p. D3158, 2018.

[40] Y. Arakawa, T. Nakamura, Y. Urino, and T. Fujita, "Silicon photonics for next generation system integration platform," *IEEE Communications Magazine*, vol. 51, no. 3, pp. 72–77, 2013.

[41] V. E. Beneš, *Mathematical theory of connecting networks and telephone traffic*, vol. 17. Academic press, 1965.

[42] D. H. Lawrie, "Access and alignment of data in an array processor," *IEEE Transactions on Computers*, vol. 100, no. 12, pp. 1145–1155, 1975.

[43] S.-Y. Li and X. J. Tan, "On rearrangeability of tandem connection of banyan-type networks," *IEEE Transactions on Communications*, vol. 57, no. 1, pp. 164–170, 2009.

[44] J. F. Adam, D. S. Engelkemier, and E. E. Sprague, "Method and system for encoding data for transmission over a serial link," Sept. 30 2003. US Patent 6,628,725.

[45] M. Sato, M. Murata, and T. Namekawa, "A new optical communication system using the pulse interval and width modulated code," *IEEE Transactions on Cable Television*, no. 1, pp. 1–9, 1979.

[46] Z. Ghassemlooy, R. Reyher, E. Kaluarachchi, and A. Simmonds, "Digital pulse interval and width modulation," *Microwave and Optical Technology Letters*, vol. 11, no. 4, pp. 231–236, 1996.

[47] S. Patil, "Arbiters and synchronizers," *Project MAC Progress Report X, Massachusetts Institute of Technology, Cambridge, Massachusetts*, pp. 24–28, 1972.

[48] B.-J. Kwak, N.-O. Song, and L. E. Miller, "Performance analysis of exponential backoff," *IEEE/ACM Transactions on Networking (TON)*, vol. 13, no. 2, pp. 343–355, 2005.

[49] M. A. Kossel and M. L. Schmatz, "Jitter measurements of high-speed serial links," *IEEE Design & Test of Computers*, vol. 21, no. 6, pp. 536–543, 2004.

[50] Corning, *Fiber Array Units datasheet*, 2017.

[51] Molex, *Rack-Mount Fiber Enclosures and Cassettes Datasheet*, 2018.

[52] DARPA, "Photonics in the package for extreme scalability (pipes)," 2018.

[53] M. Mubarak, C. D. Carothers, R. B. Ross, and P. H. Carns, "Enabling parallel simulation of large-scale hpc network systems.," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 1, pp. 87–100, 2017.

[54] Mellanox, *Mellanox 1U EDR 100Gb/s InfiniBand Switch System and IB Router Hardware User Manual, Model: SB7700*, 2017.

[55] N. Wolfe *et al.*, "Preliminary performance analysis of multi-rail fat-tree networks," in *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pp. 258–261, IEEE, 2017.

[56] DOE, "Characterization of the doe mini-apps." Accessed: Nov. 2018.

[57] S. N. Labs, "SST DUMPI trace library." Accessed: Nov. 2018.

[58] Cisco, *25GBASE SFP28 modules datasheet*, 2018.

[59] J. F. Bulzacchelli *et al.*, "A 28-gb/s 4-tap ffe/15-tap dfe serial link transceiver in 32-nm soi cmos technology," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 12, pp. 3232–3248, 2012.

[60] N. S. Kim, D. Blaauw, and T. Mudge, "Quantitative analysis and optimization techniques for on-chip cache leakage power," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 10, pp. 1147–1156, 2005.

[61] A. B. Kahng, B. Lin, and S. Nath, "Orion3. 0: a comprehensive noc router estimation tool," *IEEE Embedded Systems Letters*, vol. 7, no. 2, pp. 41–45, 2015.

[62] N. Muralimanohar, R. Balasubramonian, and N. P. Jouppi, "Cacti 6.0: A tool to model large caches," *HP laboratories*, pp. 22–31, 2009.

[63] K. Chen *et al.*, "Osa: An optical switching architecture for data center networks with unprecedented flexibility," *IEEE/ACM Transactions on Networking*, vol. 22, no. 2, pp. 498–511, 2014.

[64] P. Beebe, J. M. Ballantyne, and M. F. Tung, "An introduction to MEMS optical switches," 2001.

[65] P. Singh, D. K. Tripathi, S. Jaiswal, and H. Dixit, "All-optical logic gates: designs, classification, and comparison," *Advances in Optical Technologies*, vol. 2014, 2014.

[66] A. Huang, "Optical digital computers," in *Proceedings of the 1989 Acm/ieee Conference on Supercomputing*, pp. 446–449, ACM, 1989.