SLaC: Stage Laser Control for a Flattened Butterfly Network

Yigit Demir
Intel Corporation
Portland, OR, USA
yigit@u.northwestern.edu

Nikos Hardavellas
Department of Electrical Engineering and Computer Science
Northwestern University, Evanston, IL, USA
nikos@northwestern.edu

ABSTRACT

Photonic interconnects have emerged as a promising candidate technology for high-performance energy-efficient on-chip, on-board, and datacenter-scale interconnects. However, the high optical loss of many nanophotonic components coupled with the low efficiency of current laser sources result in exceedingly high total power requirements for the laser. As optical interconnects stay on even during periods of system inactivity, most of this power is wasted, which has prompted research on laser gating. Unfortunately, prior work on laser gating has only focused on low-scalability on-chip photonic interconnects (photonic crossbars), and disrupts the connectivity of the network which renders a high-performance implementation challenging. In this paper we propose SLaC, a laser gating technique that turns on and off redundant paths in a photonic flattened-butterfly network to save laser energy while maintaining high performance and full connectivity. Maintaining full connectivity removes the laser turn-on latency from the critical path and results in minimal performance degradation. SLaC is equally applicable to on-chip, on-board, and datacenter level interconnects. For on-chip and multi-chip applications, SLaC saves up to 67% of the laser energy (43–57% on average) when running real-world workloads. On a datacenter network, SLaC saves 79% of the laser energy on average when running traffic traces collected from university datacenter servers.

1. INTRODUCTION

Photonics have emerged as a promising solution to meet the growing demand for high-bandwidth, low-latency, and energy-efficient communication in manycore processors [4, 9,24,27,32,36,37,44], chip-to-chip interconnects [5,3,11,15, 29,31], and large-scale datacenters [1,19,41]. However, the laser is a major contributor to the total power consumption of the photonic interconnect, and the majority of the laser energy is wasted when the utilization is low. In real life, the interconnect often stays idle for long periods: compute-intensive workloads underutilize the interconnect (common in scientific applications), and servers in the cloud often stay idle or exhibit load imbalances (Google-scale datacenters are typically less than 30% utilized [2]). While the full laser power is required to support periods of high interconnect activity, the laser is wasted during idle times between message arrivals because photonic interconnects are always on.

Previous work has focused on making on-chip photonic interconnects energy proportional [13,14], but has focused mainly on photonic crossbars. Energy proportionality is desirable not only for on-chip photonic interconnects, but also for multi-chip systems and datacenters with photonic networks. To service high levels of traffic across a large number of nodes, such scaled-out systems often exploit scalable network topologies. For example, full-optical clos networks are widely deployed in datacenters in Facebook [41] and Microsoft [19], and flattened butterfly has been proposed as a cost-efficient alternative for datacenters by researchers at Google [1]. The flattened butterfly topology provides path diversity between source and destination pairs and has half the cost of iso-performance clos [25]. Thus, flattened butterfly offers high throughput while keeping the hardware cost modest.

Laser power-gating is a promising technique to mitigate the high laser power consumption of photonic interconnects, but it reduces the system's performance when messages have to wait for the laser to turn on. On a flattened butterfly, power-gating photonic links naively may result in significant performance degradation, because a packet may end up waiting for the laser turn-on multiple times as it crosses multiple routers. In this paper we propose Staged Laser Control (***SLaC***), a laser control technique for flattened butterfly networks which turns off the majority of the network to save laser energy, while maintaining full connectivity. Maintaining full connectivity removes the laser turn-on latency from the critical path and results in minimal performance degradation. SLaC turns off the majority of the network when the utilization is low to save energy, and activates additional stages when the utilization is high to increase performance.

From an on-chip interconnect to a multi-chip system to a datacenter, any network with a topology that provides path diversity, such as flattened butterfly, clos, dragonfly, or fat trees, can implement SLaC with very little changes. In this paper we choose to implement SLaC on a flattened butterfly network. More specifically, our contributions are:

- We present SLaC, a laser gating technique adapted specifically to flattened butterfly topologies and show that naively turning on/off redundant paths is impractical.
- We adapt SLaC to on-chip, chip-to-chip (board- or system-level), and datacenter-scale interconnects and show its efficacy in all these domains using real-world applications and traces collected from live datacenter servers.
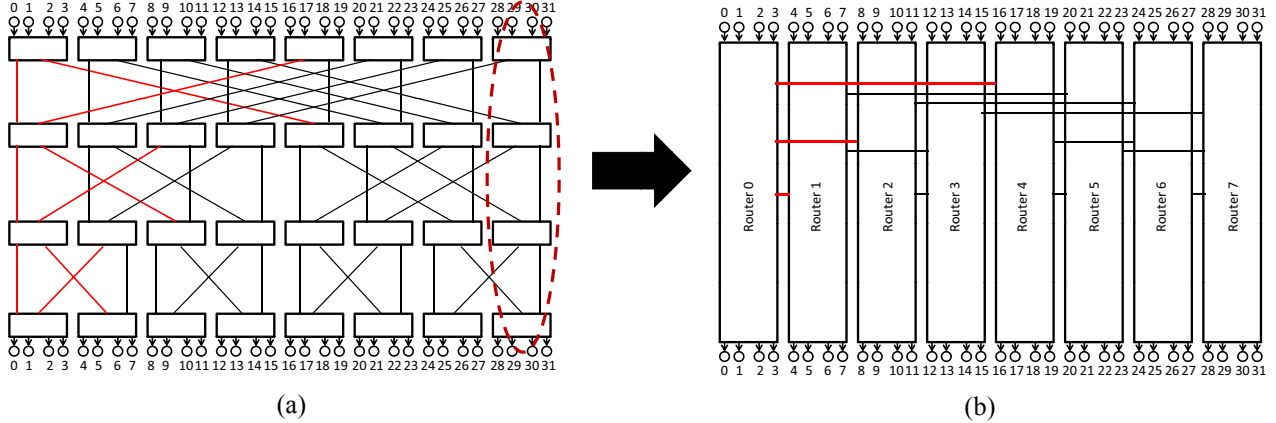
FIGURE 1. 2-ary 4-fly butterfly (a) and 2-ary 4-flat flattened butterfly (b) networks.

- In the case of datacenter flattened butterfly networks, we show how the socket interface can be intercepted to turn the entire network off when it is beneficial and fully hide the performance penalty for re-activating it.
- We evaluate SLaC on on-chip and multi-chip networks and show that it saves up to 67% of the laser energy (43–57% average) while reducing performance by only 2% on real-world workloads. On a datacenter network, SLaC saves 79% of the laser energy on average when running traces collected from a university datacenter.

2. BACKGROUND

2.1 Flattened Butterfly Topology

Flattened butterfly is derived by combining the routers in each column (in different dimensions) of a conventional butterfly network into a single router. In Figure 1a, we present a 2-ary 4-fly butterfly network with 32 routers, which can be flattened into a 2-ary 4-flat butterfly network with 8 routers shown in Figure 1b. The flattening of the routers help flattened butterfly provide a higher path diversity compared to a butterfly network because the inter-hops between the dimensions can be taken in any order. For example messages from node 0 to node 22 can either follow the route from Router 0 to Router 4 to Router 5 or from Router 0 to Router 1 to Router 5. The flattened butterfly exploits high-radix routers and an adaptive load balancing routing algorithm to realize a low cost network that provides high performance.

A traditional on-chip flattened butterfly network [25] uses long electrical links to connect the row and the column neighboring routers (Figure 2a). These links provide efficient message transfer between the routers because they connect the source and the destination pairs in the shortest way. However, this layout is impractical for a photonic implementation of flattened butterfly, because it requires waveguide crossings. When waveguides cross, every wavelength in a waveguide imposes crosstalk over every other wavelength in the crossing waveguide, which reduces the signal quality. In order to maintain the quality of the communication, high laser power is needed, which reduces the energy efficiency and makes this design impractical.

The layout shown in Figure 2b implements all links within the serpentine waveguide similar to [24] and avoids waveguide crossings. However, it results in increased message latency and additional laser power consumption due to long waveguides. On this serpentine waveguide layout, a message from Router 1 to Router 2 has to traverse one dimension of the chip twice (over Router 12 and Router 14), whereas in the electrical implementation (Figure 2a) it takes only one short direct link. Similarly, the link connecting Router 0 to Router 4 is approximately 2.5x longer than its electrical counterpart. Message latency and optical loss in a waveguide are proportional to the waveguide length, so these longer links are

FIGURE 2: Electrical link (a) and serpentine waveguide (b) layouts for flattened butterfly topology.
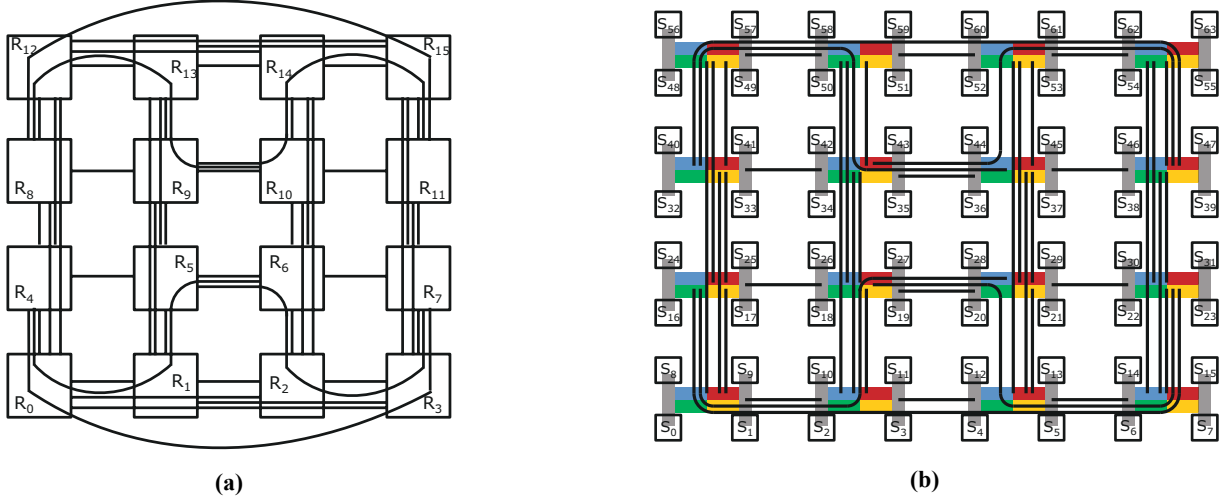
FIGURE 3: On-chip (a) and multi-chip (b) photonic flattened butterfly layout.

slower and require more powerful lasers which reduces the energy efficiency.

We map a flattened butterfly in the photonic domain following the layout shown in Figure 3a, which aims to connect the routers using the shortest links possible while avoiding crossings. This layout allows waveguides to run across the chip in one dimension (i.e., y-dimension) and routes the waveguides in the other dimension (i.e,. x-dimension) around and in between them to connect the routers with the shortest possible links without intersecting. Figure 3b presents the same layout for a board-level (i.e., chip-to-chip) flattened butterfly interconnect. It is important to note that in Figure 3's photonic flattened butterfly layout all routers are connected to their immediate neighbors via a short straight waveguide. Therefore, a message that needs to travel across one side of the chip twice in Figure 2b's serpentine waveguide layout (from Router 1 to Router 2), now only takes a small and direct hop. Moreover, the longest link in Figure 3's photonic flattened butterfly layout is 2.5x shorter than the longest link in the serpentine layout (from Router 0 to Router 4), which lowers the average message latency and laser energy.

Due to all these advantages, in this paper we evaluate a photonic flattened butterfly with the layout shown in Figure 3.

2.2 Laser Primer

Before diving into laser technology details, it is important to emphasize that SLaC does not depend on a singular laser technology. SLaC on datacenter-scale interconnects can be successfully implemented using traditional gaussian comb lasers with long turn-on delay times, as our evaluation in Section 5.5 shows. In fact, this is exactly the type of lasers that this paper assumes for SLaC on datacenter interconnects. For SLaC on on-chip and on-board (i.e., chip-to-chip) interconnects, any fast WDM-compatible continuous-wave laser that can be integrated on chip is suitable, including the InP and Ge lasers [7,30,38,18] we assume in this work. This

section focuses mostly on the advantages of newer laser technologies for on-chip and chip-to-chip applications.

Previous works [3,4,27,32,37] typically use off-chip lasers because of their temperature stability, easy replacement, and energy efficiency (30% for gaussian comb lasers [16]). However, recent research [21] shows that output spectrum power variations and laser-to-fiber and fiber-to-chip coupling losses add 7–8 *dB* optical loss, thus off-chip lasers are in reality only 6% efficient. In comparison, on-chip laser sources [28] attain wall-plug efficiencies up to 15%, while enabling wavelength-division multiplexing (WDM). WDM can be implemented by feeding a set of wavelengths generated by an array of single-wavelength lasers into an optical bus. On-chip lasers offer energy efficiency and easy packaging, but their wall-plug power consumption counts against the processor's overall power budget. In either case, the laser power consumption remains a considerable overhead, especially when accounting for realistic optical loss parameters and laser efficiencies, emphasizing the need for power-gating the laser source. Power-gating on-chip lasers can increase the energy efficiency of a photonic interconnect by up to 4x [21].

Laser power-gating has been overlooked due to the high turn-on latency (up to 1 *μs* [21]) of the traditional distributed feedback comb lasers that are widely assumed in photonic interconnects [3,4,27,32,37]. Comb lasers use diffraction grating to form the optical cavity. Temperature affects the diffraction grating pitch and the active region's refractive index, which alter the diffraction grating's wavelength selection, and hence the laser's emission wavelength. Thus, when comb lasers turn on they need time to reach a set temperature and lock at the designated wavelength. This high delay hampers power gating. In contrast, Fabry-Perot (FB) lasers use two discrete mirrors to form the optical cavity, and their emission wavelength depends not on temperature but on the n-type doping level and the strain applied during the cavity development. Thus, when they are turned on (pumped to the
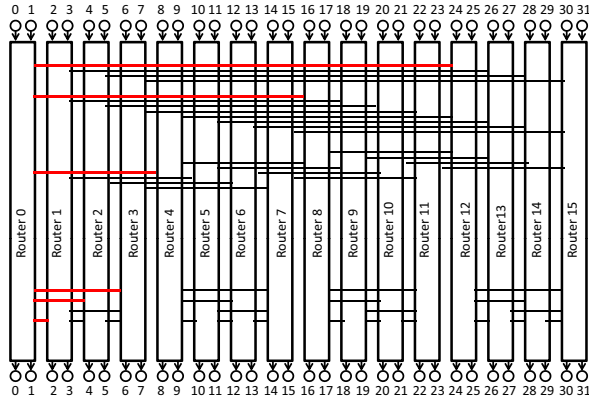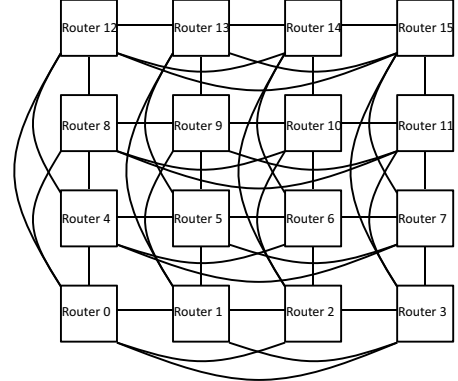
FIGURE 4. Flattened butterfly configurations.

(a) (b)

lasing threshold), they lase at the designated wavelength without requiring time for temperature stabilization/locking, and, hence, are suitable for power gating.

In general, laser power-gating depends strongly on fast lasers. While not a mature technology yet, it is important to note that fast lasers with *ns*-scale turn-on times have been manufactured and their turn-on delay has been characterized on real hardware prototypes [30,38,18,7,34], and is in agreement with theoretically-derived results. To turn the laser on, a supply current is applied to the laser. When the carrier density exceeds the threshold density, laser oscillation starts and light output increases drastically (laser turn-on). The time it takes from the current injection to lasing at stable power is the "laser turn-on delay" which is governed by the carrier life time and is in the order of *ns* ([39], pp. 80–82). The turn-on delay of Fabry-Perot lasers is highly tunable by design parameters, and nanosecond or sub-nanosecond laser turn-on delays are both theoretically predicted [23,22,39 pp. 83] and achievable in real implementations [7,34,30,39].

For example, InP-based diode FB lasers [30] have been manufactured and shown to emit light with a 2 *ns* long electrical pulse excitation (so the laser turn-on latency is at most 2 *ns*). InP-lasers have high peak power, and their emission wavelength is tunable in a wide range and highly stable with temperature, which makes them DWDM-compatible. Moreover, InP-lasers can be integrated on Si [38,18] so they can be used as an on-chip laser source. Similarly, Ge-based FB on-chip lasers have been manufactured [34] and the turn-on delay of hardware prototypes was measured at 1.5 *ns* at most for both optically- and electrically-pumped implementations [34, 7, 26]. Besides their fast turn-on time, Ge-lasers [7] are suitable for on-chip photonic interconnects because they can be built within a standard-width (1 *μm*) waveguide at only 7.68×10⁻³ *mm²*, operate in room temperature, and are DWDM-compatible as they exhibit a gain spectrum of over 200 *nm* [7].

It is important to offer the interested reader an additional perspective on laser turn-on times: VCSELs can turn-on with sub-100 *ps* delay [39], and thus can be directly modulated at

over 35 GHz [46]. However, VCSELs are unsuitable for on-chip applications with WDM because they emit significant heat, and their operating wavelengths are defined by the epitaxial growth [21] which challenges the implementation of a multi-wavelength VCSEL array on chip. Moreover, it is hard to protect the integrity of messages with direct laser modulation due to chirping and the pattern effect [39].

3. LASER CONTROL SCHEMES

The laser control schemes aim to save laser energy by turning the lasers off whenever the photonic links are not utilized. Energy savings come at the cost of increased message latency, because messages ready for transmission may find the laser off and wait for it to turn on before they are sent.

A naive approach to laser power-gating is to turn off the photonic links whenever they are idle (*Naive*). In this case, all paths between router pairs can be turned off simultaneously, forcing messages to wait for the lasers to turn on before transmission. Furthermore, a packet routed over multiple hops can experience the laser turn-on delay multiple times, as multiple photonic links along the path could be turned off. This cumulative laser turn-on delay effect can have a significant impact on the performance and should be avoided.

3.1 Stage Laser Control Scheme

Flattened butterfly provides high path diversity which increases the probability of packets avoiding the laser turn-on latency. Figure 4a presents a 4-ary 3-flat flattened butterfly configuration, where Router 0 can send a message to Router 15 using either Router 3 or Router 12. So, if the photonic link between Router 3 and Router 15 is turned off, the messages can still be directed through Router 12 without waiting for that link to be turned on. Another important point to note is that by steering the traffic through Router 12, the opportunity to turn the laser off for the photonic link between Router 3 and Router 15 is maximized.

Removing the laser turn-on delay from the critical path of the messages reduces the performance penalty of laser gating. A *k*-ary *n*-flat flattened butterfly consists of *n* *k*-ary *(n-1)*-flat

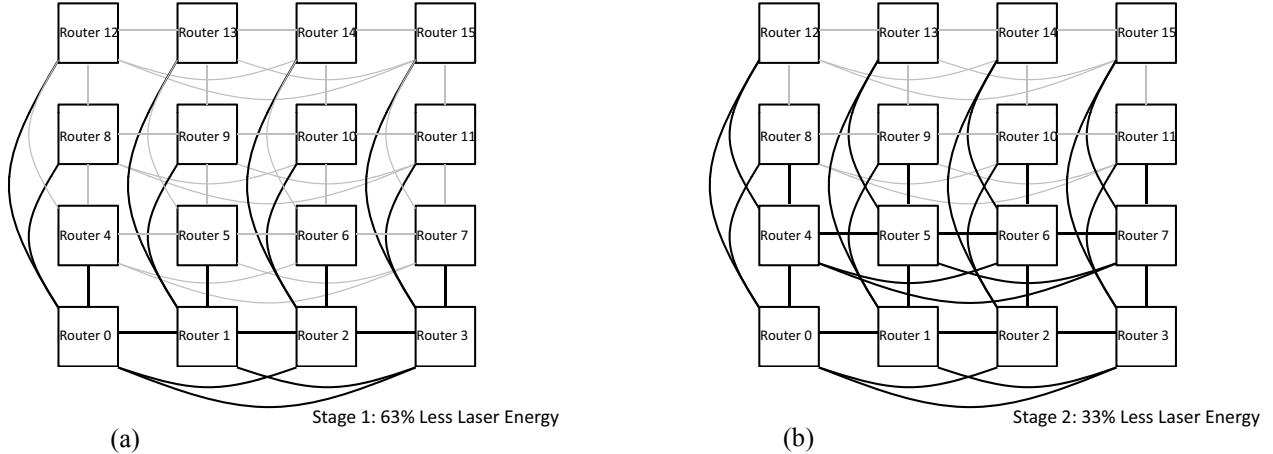(a)   (b)

FIGURE 5. Laser-gating stages for the flattened butterfly network.

flattened butterfly networks connected together (Figure 4b). We refer to each one of these networks as a *Stage*. Stage *k* comprises the *x*-dimension links of the *k*-th row of routers, and the *y*-dimension links connecting the *k*-th row routers to routers at row *n>k*. We can provide full connectivity even if we turn off the *n-1* of these *k*-ary *(n-1)*-flat flattened butterfly networks (stages) given that all other stages have active connections to the active stage. Turning off stages saves laser energy, while turning on additional stages increases path diversity and performance. SLaC (Stage Laser Control) monitors the interconnect traffic and turns off stages when the utilization is low to save energy, and activates additional stages when the utilization is high to increase performance.

SLaC monitors the message traffic by inspecting the buffer utilization levels, an accurate and lightweight method [10], to decide on the number of stages to activate on-the-fly. When a router in an active stage has an input buffer with utilization over a certain high threshold, that router broadcasts a message to all other routers to activate a new stage. This message traverses the part of the network that is already active. SLaC activates additional stages in ascending order (and deactivates them in descending order), thus the next available stage is activated after its routers receive the turn-on signal. The newly activated stage broadcasts a message after its activation, so all routers update their routing table and can use the newly activated stage.

The newly activated stage can be turned off when the router that activated it becomes underutilized, i.e., its input buffer utilization falls below a low threshold. In that case, that router broadcasts a stage turn-off message which deactivates the last activated stage. The deactivating stage first broadcasts a message to update all other routers' routing tables, so it stops receiving messages. The stage then turns off after it serves all the packets in its buffers.

SLaC adaptively routes the traffic through the active stages (a stage that received a turn-off signal is not considered active, so it does not receive new messages) which balances

the traffic, avoids messages waiting for the laser to turn-on (thus it completely hides the turn-on delay) and avoids unnecessary stage activation which maximizes the laser energy savings. In SLaC, when stage k is active, all stages from stage 1 to stage k are active. Thus, for simplicity from now on by "***Stage k***" we will denote that all stages 1 through k (inclusive) are active.

Figure 5 shows how SLaC works on a 4-ary 3-flat butterfly network. This network consist of 4 stages which SLaC activates adaptively. Each stage includes the *x*-dimension links of one row of routers, and the *y*-dimension links than connect that row to higher rows. The set of black links in Figure 5a show Stage 1 activated, and Figure 5b shows Stages 1 and 2 activated. When in Stage 1 (Figure 5a), if an input buffer in Router 0 has over 75% utilization, it broadcasts a turn-on message. As a result, Routers 4, 5, 6, and 7 (i.e., row 2) turn on their links and Stage 2 is activated. Once Stage 2 is active (Figure 5b), it broadcasts a message so all routers update the list of stages they can send messages to. If the same input buffer in Router 0 goes below 25% utilization, it broadcasts a turn-off message, so Stage 2 turns off and all other routers update their routing tables. In our experimental evaluation we faithfully model all the additional message broadcasts and latencies in both turn-on and turn-off sequences.

On Stage 1 the flattened butterfly network consumes 63% less laser energy compared to a conventional flattened butterfly network with photonic links that are always on (*No-Ctrl*). Stage 1 can cause high contention, because there is only one path between each router pair, which may reduce performance. SLaC always keeps the Stage 1 on. Stage 2 saves 33% of the total laser energy while providing multiple paths between source and destination pairs which can provide higher throughput (via a dynamic routing algorithm).

3.2 Routing with SLaC
SLaC employs an adaptive routing algorithm which increases the opportunities to save laser energy when the traffic is low by using active stages only, and activates addi-
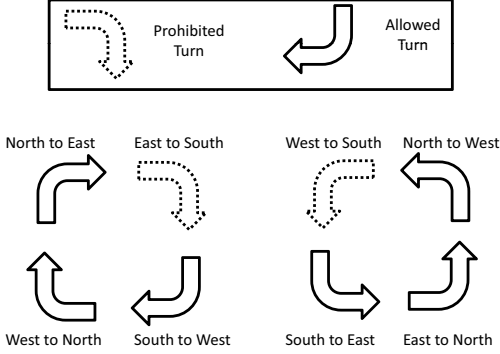
FIGURE 6. Turns in SLaC's routing algorithm.

TABLE 1. Architectural Parameters.

CMP Size	64 cores, 480 mm^2
Processing Cores	ULTRASPARC III ISA, up to 5 Ghz, OoO, 4-wide dispatch/retirement, 96-entry ROB
L1 Cache	Split I/D, 64KB 2-way, 2-cycle load-to-use, 2 ports, 64-byte blocks, 32 MSHRs, 16-entry victim cache
L2 Cache	Shared, 512 KB per core, 16 way, 64-byte blocks, 14 cycle-hit, 32 MSHRs, 16-entry victim cache
Memory Controllers	One per 4 cores, 1 channel per Memory Controller Round-robin page interleaving
Main Memory	Optically connected memory [3], 10 ns access

tional links only when it detects heavier traffic. The routing algorithm randomly selects the active stage to use, so it balances the traffic. SLaC's routing algorithm is deadlock free because it uses a dimension-ordered routing scheme. The routing algorithm routes packets in the north-south dimension first, and then in the east-west dimension. When a packet is generated, SLaC's routing algorithm first checks if the destination stage is active (i.e., Stage k if the destination router is at row k). If so, it routes a packet to that active stage first and then routes it to the destination router within that active stage. For example, in Figure 5b, when Router 4 sends a message to Router 1, packets will be routed through Router 0, rather than Router 5 (avoiding East-to-South turn). Similarly, when Router 6 sends a message to Router 1, packets will be routed through Router 2, rather than Router 5 (avoiding West-to-South turn). In conclusion, in this dimension-ordered routing, turns from East to South and from West to South are prohibited (Figure 6), therefore the routing algorithm avoids forming cycles and stays deadlock free.

If the destination router (e.g., at row k) has not activated its stage yet (Stage k), the routing algorithm randomly selects an active stage and makes three hops on it. For example, in Figure 5b, when Router 13 sends a message to Router 10, packets could be routed either through Router 5 and Router 6, or Router 1 and Router 2. The routing algorithm randomly picks which active stage to use, which balances the traffic. It is important to note that the number of maximum hops does not change, and it is limited to 3 in this example (Figure 5b).

4. EXPERIMENTAL METHODOLOGY

4.1 Interconnect Performance and Energy Analysis

To evaluate the performance and energy consumption of SLaC for a flattened butterfly (FBFLY) on-chip network in isolation from the interference of other system components or application characteristics, we employ a cycle-accurate network simulator based on Booksim 2.0 [12], which models a 4-ary 3-flat FBFLY network servicing random uniform traffic (with concentration of 4). The simulator models a 3-cycle router, with 1-cycle E/O and O/E conversions. We assume a 480 mm^2 chip, where the link latency (1–3 cycles)

is calculated based on the traversed waveguide length. The buffers are 20-flits deep, with a flit size of 300 bits. The maximum core frequency is 5 GHz, and the optical interconnect runs at 10 GHz. Latency is measured as the time required for the network to process a sample of injected packets. The on-chip FBFLY has 6144 single-wavelength lasers (one laser per wavelength per link) occupying a total of 30 mm^2. To facilitate laser turn-on/off, a laser is supplied a virtual V_{dd} through a transistor controlled by a sleep signal, as in electronics. State retention is unnecessary, eliminating most overhead. SLaC requires a slightly modified logic on the router's control plane, for which we estimate at most a 2.5% hardware overhead as in [10] which discusses a similarly-modified routing logic and control. The overhead for an optical network is likely much smaller, as electronic components are much smaller than optical ones (nm vs. μm). There is one controller per stage per router.

We evaluate the load-latency characteristics of SLaC and compare it against a photonic FBFLY interconnect that always keeps the lasers on (***No-Ctrl***), a Naive control scheme that turns off the photonic links whenever they are idle (***Naive***) and an electrical flattened butterfly network (***Electric-FBFLY***). To make a fair comparison we equalize the average power consumption of Electric-FBFLY to the power consumption of No-Ctrl by adjusting the Electric-FBFLY's datapath width (flit size). Thus, for the Electric-FBFLY we model 6-port routers with 3-cycle delay and 100-bit bi-directional links with 1-cycle, 2-cycle and 3-cycle latency (local, mid-range, and global, respectively).

For a multi-chip (wafer- or board-) scale FBFLY we model a 8-ary 3-flat flattened butterfly network where the link latency (2–15 cycles) is calculated based on the length of the traversed waveguide. The flit size is 50 bits. The datapath is narrower than the datapath of the on-chip flattened butterfly in order to keep the power consumption of the laser at reasonable levels. As Table 2 indicates, a multi-chip flattened butterfly with 50-bit flits requires 200 W of laser power, so an implementation with 300-bit flits (i.e., as wide as the on-chip FBFLY) would require 1.2 KW, which is impractical.

TABLE 2. Nanophotonic Parameters and Laser Power.

On-Chip		FBFLY	Multi-Chip		FBFLY	PtoP
	per Unit	Total		per Unit	Total	Total
DWDM		64	DWDM		16	16
Splitter	0.2 *dB*	0.6 *dB*	WG Loss	0.3 *dB/cm*	4.5 *dB*	10.5 *dB*
WG Loss	0.3 *dB/cm*	0.75 *dB*	WG Loss*	0.05 *dB/cm*	0.75 *dB*	1.75 *dB*
Nonlinearity	1 *dB*	1 *dB*	Bridge WG Loss	1 *dB*	1 *dB*	1 *dB*
Modulator Ins.	0.5 *dB*	0.5 *dB*	Modulator Ins.	4 *dB*	4 *dB*	4 *dB*
Ring Through	0.01 *dB*	0.63 *dB*	Ring Through	0.05 *dB*	0.8 *dB*	0.8 *dB*
Filter Drop	1.2 *dB*	1.2 *dB*	Filter Drop	1 *dB*	1 *dB*	1 *dB*
Receiver Margin	4 *dB*	4 *dB*	Receiver Margin	4 *dB*	4 *dB*	4 *dB*
Coupler	2 *dB*	2 *dB*	Coupler	2 *dB*	6 *dB*	6 *dB*
Total Loss		**8.68 *dB***	**Total Loss**		**21.3 *dB***	**27.3 *dB***
Detector		**-20 *dBm***	**Detector**		**-20 *dBm***	**-20 *dBm***
Laser Power Per Wavelength		**0.073 *mW***	**Laser Power Per Wavelength**		**1.34896 *mW***	**4.7863 *mW***
On-Chip Laser Power	*10% Eff.*	**21.25 W**	**Total Laser Power**	*30% Eff.*	**199.43W**	**124.73W**
Off-Chip Laser Power	*30% Eff.*	**11.11 W**	**Total Laser Power***	*30% Eff.*	**84.1W**	**43.96W**

The datacenter network we model is an 8-ary 3-flat flattened butterfly network with concentration of 8, so it supports up to 512 nodes. The router delay is 200 *ns*, and the link latency (100–200 *ns*) is calculated based on the traversed optical fiber length. The flit size is 300 bits. The multi-chip and the datacenter FBFLY have a total of 896 lasers (one per link).

4.2 Performance and Energy Modeling
To evaluate the impact of SLaC on a realistic multicore system, we model a 64-core processor on a full-system cycle-accurate simulator based on Flexus 4.0 [20,45] integrated with Booksim 2.0 [12] and DRAMSim 2.0 [40]. Table 1 details the architectural modeling parameters for the on-chip analysis. We assume a shared and physically distributed L2 cache and directories. The memory controllers are uniformly distributed on the chip, and they use the same physical interconnect with VCs to avoid deadlock. All messages below the L1 cache traverse the interconnect. The power consumption of the electrical interconnect is calculated using DSENT [42]. We target a 16 *nm* technology, and have updated our tool chain accordingly based on ITRS projections [17]. The simulated system executes a selection of benchmarks from SPLASH-2, PARSEC and other scientific workloads. To analyze the multi-chip configurations we conduct a similar-size simulation where each thread is placed at a different site.

To evaluate the performance of SLaC on the datacenter-scale FBFLY we use snippets of traces collected from routers in a university datacenter [6]. The EDU1 and EDU2 traces in [6] consist of packets passing through a single router in a university datacenter, so we scale the workload to reflect all-to-all traffic on the FBFLY network. We inject a different copy of the packet trace at each FBFLY router starting from a random location within the trace, and we estimate network performance by measuring the average message delivery latency.

4.3 Laser Power Consumption Calculation
We calculate the laser power savings of SLaC and compare it against equal power networks for both on-chip and multi-chip implementations. Table 2 shows the optical loss parameters for the modulators, demodulators, drop filters, and detectors introduced in [3] and assumed for the modeling of the on-chip FBFLY, and the optical loss parameters introduced in [29] which are assumed for the multi-chip integration. The modulation and demodulation energy is 150 *fJ/bit* at 10 *GHz* [3] for both cases. The laser power per wavelength and the total laser power are calculated in Table 2 using the analytical models in [24]. We model on-chip laser efficiency of 10% and off-chip laser efficiency of 30%. We calculate the laser power consumption for both traditional (0.3 *dB/cm* [8]) and aggressive (0.05 *dB/cm* [29]) waveguide loss (the aggressive assumption is noted with a ***** in Table 2).

We model a laser turn-on delay of 1.5 *ns* for the on-chip laser source [30,34,7,26]. Gaussian "comb" lasers are a popular choice for external lasers, and they can be turned on and off within 1 *μs* [21]. Different than their on-chip counterparts, turning on a comb laser is followed by clock and data recovery (CDR) which can synchronize within 200 *ns* [1]. In our modeling we add this latency to the comb laser turn-on delay.

It is important to note that any uncertainty in the nanophotonic parameters at Table 2 will not impact the performance of SLaC. SLaC will still save the same fraction of laser power as it depends only on the shape of traffic, not on the components' *dB* rating, thus it can tolerate high variability.
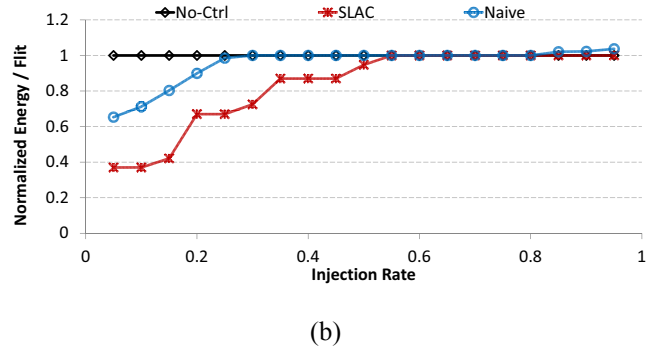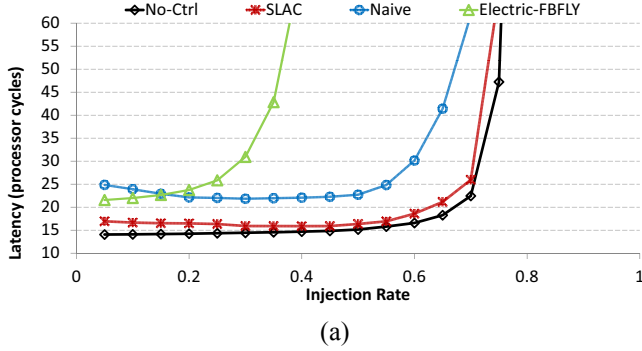
FIGURE 7. Load-latency (a) and laser energy per flit (b) for FBFLY topology with No-Ctrl, SLaC, Naive, and Electric-FBFLY.

5. EXPERIMENTAL RESULTS

5.1 Network Performance and Energy Impact

SLaC increases the message latency due to non-minimal routing, but provides high throughput. Figure 7a compares the load-latency of SLaC against No-Ctrl, Naive, and Electric-FBFLY. Under random uniform traffic SLaC achieves on average 16.9 cycles zero-load message latency, which is 2.8 cycles higher than No-Ctrl. However, the throughput provided by SLaC under higher injection rates is almost equal to No-Ctrl's, and 1.15x and 2.14x higher than Naive's and Electric-FBFLY's respectively. Naive control incurs an additional 10.8-cycle zero-load message latency over No-Ctrl due to the cumulative laser turn-on delay (packets wait for the laser to turn on at almost every hop most of the time).

We find that broadcasting a control message when stages activate or deactivate does not increase network congestion much. Upon a broadcast, either a new stage just activated and immediately relieved pressure (dropping utilization much below 75%) or a stage deactivated (i.e., utilization is already below 25%). Broadcast messages are single-flit and increase traffic by less than 3.6%, and thus cause no congestion.

Figure 7b presents the Laser Energy per Flit (EPF) with SLaC compared against No-Ctrl and Naive. SLaC trades off a small latency increase for laser energy savings up to 63%. The steps observed in the EPF graph correspond to new stages turning on. Naive control achieves lower energy savings because it does not reuse activated links, wasting additional laser turn-on time and energy unnecessarily.

5.2 Performance and Energy Impact on a Multicore

SLaC achieves high laser energy savings but increases the average message latency slightly, because it uses non-minimal routing which prefers to use active links. In this section, we investigate the performance impact of SLaC on a multicore processor with a 4-ary 3-flat FBFLY. Figure 8a shows the speedup of SLaC compared against No-Ctrl, Naive and Electric-FBFLY (power-equivalent to No-Ctrl). The numbers below each application denote the injection rate that this application imposes on the interconnect. The performance of SLaC is only 2% away from No-Ctrl. SLaC outperforms the Naive control and Electric-FBFLY by 1.1x and 1.31x respectively, because it provides higher throughput under heavier traffic by turning on additional stages. Figure 8b presents the laser energy consumption per flit, where SLaC saves 43% laser energy on average (59% maximum). Naive control manages to save some laser energy on workloads with lighter traffic, however it slows down the execution significantly when the traffic demand is high and ends up consuming higher laser energy. This shows the importance of providing high performance (by maintaining full connectivity and additional stage activation) when targeting laser energy savings. The laser energy per flit for Electric-FBFLY is not shown in Figure 8b as it is power-equivalent to No-Ctrl.

While the energy savings of SLaC depend on the traffic rate, it still saves a significant amount of laser energy across all workloads. Figure 9 presents the fraction of time spent in each stage for SLaC when running appbt, fmm and bodytrack. In workloads with low message traffic (fmm) SLaC
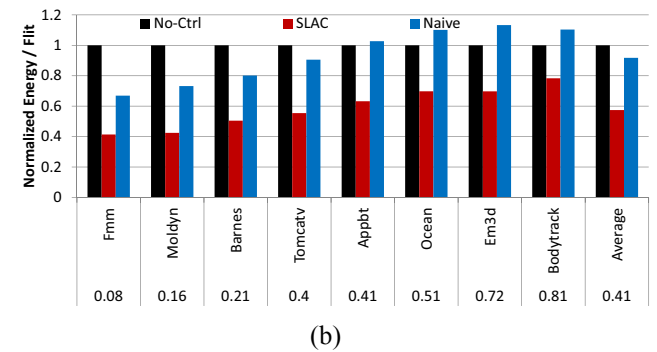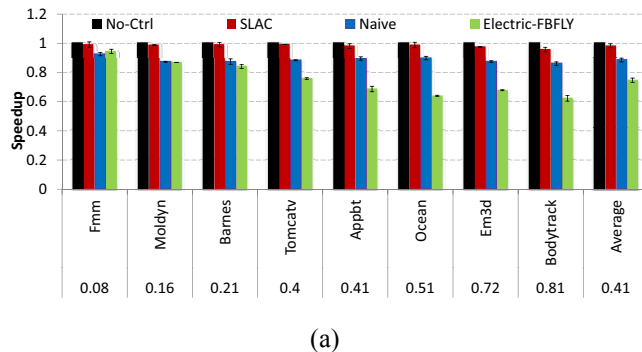

FIGURE 8. Speedup (a) and laser energy per flit (b) for a multicore with No-Ctrl, SLaC, Naive, and Electric-FBFLY.

8

stays in Stage 1, maximizing energy savings, while for the ones with higher traffic demand (bodytrack) SLaC tends to turn on higher stages to provide increased performance. For the appbt workload, the fraction of time spent in Stage 3 is higher than the fraction of time spent in Stage 2, which shows the bursty message traffic behavior of the workload. Recall that "Stage 3" denotes that stages 1–3 are active.

5.3 Thermal Effects on a Multicore

The on-chip lasers we model consume 21.25W under the No-Ctrl scheme. This power consumption is counted against the power budget of the chip, and the corresponding dissipation increases temperature by 5.8°C. SLaC reduces the laser power by 43%, which reduces dissipation to 12W, i.e., only 3°C heating. Our evaluation accounts for all these effects.

5.4 Performance and Energy Impact on a Multi-Chip

FBFLY networks are highly scalable and can connect thousands of nodes. For that reason they are preferred for multi-chip integration systems (e.g., wafer-scale integration similar to the Oracle Macrochip [29,31]). When SLaC is employed on a wafer-scale photonic FBFLY network, its impact is higher due to the increased laser power consumption of the wafer-scale network. In this section we present the performance increase and laser energy savings of SLaC when it is employed on a wafer-scale 8-ary 2-flat FBFLY network. We compare a wafer-scale SLaC against wafer-scale No-Ctrl, Naive, and a point to point (PtoP) network which has been proposed for wafer-scale integration [29,31]. To make a fair comparison, we compare SLaC against an equal power PtoP network by adjusting PtoP's datapath width. However, there is little consensus on the waveguide loss parameter which has a direct impact on the laser power consumption, so we consider both aggressive waveguides (0.05 *dB/cm* loss [29]) and traditional waveguides (0.3 *dB/cm* loss [8]). With the aggressive waveguides, a power-equivalent PtoP network can support 25-bit links, while with traditional waveguides it can only support 4-bit wide links.

Figure 10a shows the speedup of SLaC for the wafer-scale network. On average SLaC is only 3% slower than No-Ctrl, and 1.44x and 1.86x faster than Naive and PtoP respectively. Even with the aggressive waveguides, SLaC is 1.14x faster than the PtoP network proposed in [19]. Figure 10b shows
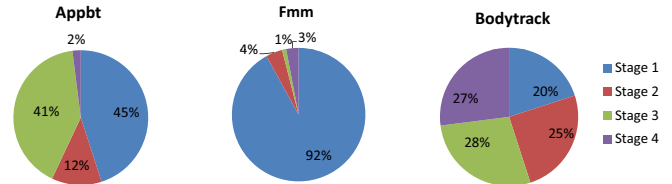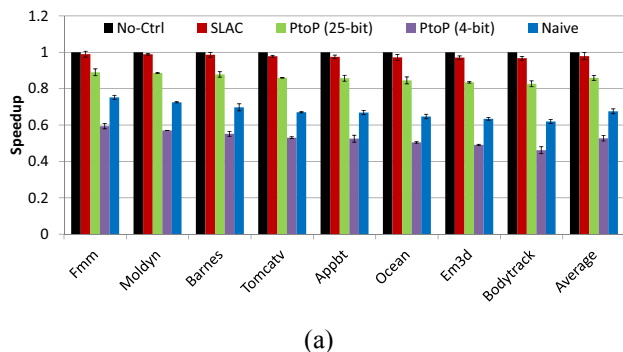
FIGURE 9. Time spent at each stage level (multicore).

the laser energy per flit comparison. SLaC saves 57% of the laser energy on average (and up to 67%), whereas PtoP saves between 4–17% on average, and Naive causes an increase in the energy consumption by 10%.

5.5 Performance and Energy Impact on a Datacenter

FBFLY networks have been proposed for deployment in the datacenter because they provide low latency, high throughput, and scalability at reasonable cost [1]. SLaC can be exploited to improve the energy efficiency of a photonic datacenter network with FBFLY topology. We model typical large-scale datacenter networks that employ optical fibers powered by external lasers attached to the network switches.

Figure 11a shows the increase in message latency caused by SLaC and Naive control as a function of the laser turn-on delay under the EDU1 and EDU2 traces. A 1.2 *μs* laser turn-on delay results in 0.29–0.35 *μs* increase for SLaC and 1.14–1.27 *μs* increase for Naive control. As the turn-on delay increases, the message latency increases slowly for SLaC and much faster for Naive because messages have to wait for the laser turn-on more frequently and they fill up the buffers. A hypothetical laser with a high 10 *μs* turn-on delay results in 0.75–1.2 *μs* increase for SLaC and 35.4–41.3 *μs* increase for Naive control. Overall, even though comb lasers are slow to turn on, SLaC avoids stalling messages by transmitting them through active stages. As a result, for our datacenter traces SLaC can tolerate up to 10 *μs* laser turn-on delay, which is 8x higher than a typical external laser [21]. Under an exceedingly high 100 *μs* laser turn-on delay, Naive control causes more than 1,000 *μs* additional delay to the packets, while SLaC keeps it under 20 *μs*.

The datacenter traces EDU1 and EDU2 exhibit sparse and bursty packet injection trends. Therefore SLaC can turn-off
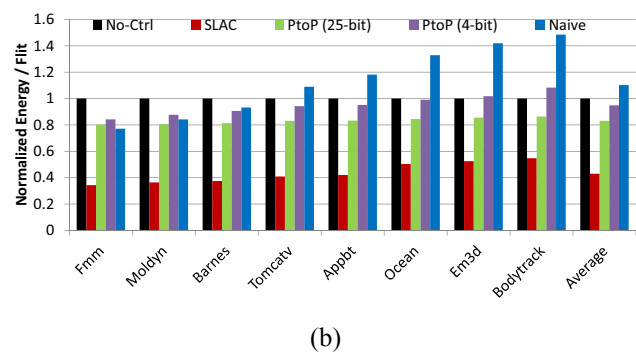
(a)   (b)

FIGURE 10. Speedup (a) and laser energy per flit (b) for a multi-chip system with No-Ctrl, SLaC, PtoP, and Naive control.
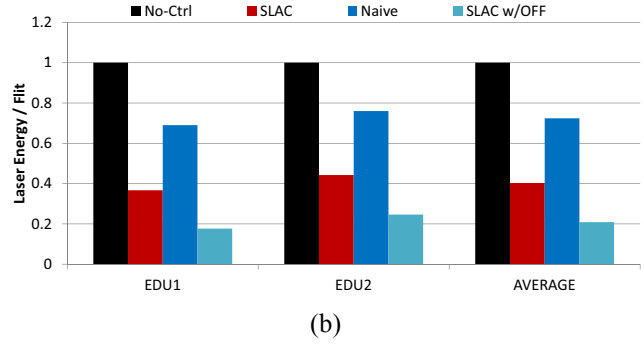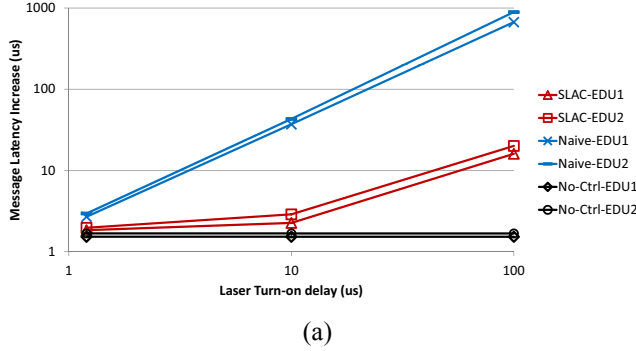
(a)

(b)

FIGURE 11. Message latency (a) and laser energy per flit (b) for a datacenter flattened butterfly with No-Ctrl, SLaC, Naive, and SLaC w/OFF configurations.

most of its stages during periods of low traffic and achieve high laser energy savings. Figure 11b shows the laser energy per flit for SLaC, No-Ctrl and Naive control when running database workloads. SLaC saves 60% of the laser energy while Naive only saves 28%.

5.6 Cooperation of SLaC with the OS

Figure 12a shows the fraction of time spent in each stage during the execution. Due to the sparse arrival of the messages, most of the execution time is spent in Stage 1. SLaC aims to remove the laser turn-on latency from the critical path, so it keeps Stage 1 always turned on, which means that laser energy is still wasted when there are no messages in the network. To minimize this energy waste and still hide the laser turn-on delay, SLaC can turn off all stages, and predict an upcoming message ahead of time with the help of the OS. We term this optimization "***SLaC w/OFF***".

The main idea behind SLaC w/OFF is that the OS can take advantage of the packet preparation latency of the TCP/IP stack to turn on the lasers ahead of time to hide the laser turn-on latency completely. Performance measurements on modern Intel Xeon 2.13 *GHz* 5138 based servers with a fiber optic NIC running Linux 2.6.18 RC3 kernel [33] show that it takes 950 *ns* for a process to send a message to the socket interface on a connection that has already been established. Evoking a socket write begins the TCP layer to initiate transmission, copy the application buffer into the transmit queue in kernel space and prepare a datagram for the IP layer (260 *ns*). Then the IP layer does routing, segmentation, processes the IP header, and eventually calls the network device driver (550 *ns*). The network device driver constructs the output packet queue entry and calls the precise hardware implementation of the NIC card to transmit the frame by passing a pointer to the packet descriptor (430 *ns*). This causes a control register write within the NIC to set up a DMA transfer to fetch the pointer, and when it completes control is handed to the NIC card (400 *ns*). Another 760 *ns* are consumed by the NIC to process the core register write, interpret the descriptor, and based on the descriptor initiate a DMA to fetch from main memory the data of the packet to transmit. Each 64-byte cache line access to memory takes an

estimated 400 *ns* to propagate from the PCIe signal pins to memory and back. Thus, it takes a total of 3.75 *μs* for an application to launch a packet onto the fiber interface.

This means that the SLaC w/OFF laser control has plenty of time to turn the lasers on and completely hide the 1.2 *μs* latency of turning on a comb laser. Thus, SLaC w/OFF can turn off the whole network without incurring any additional message delay. Even OS-level optimizations to minimize the software overhead and memory copies are unlikely to bring the socket interface's transmission delay below 1.2 *μs*, as the hardware latency of the NIC alone is 1.16 *μs*. Thus, there is plenty of opportunity for the OS to intercept the *socket_write( )* call, send a laser turn-on signal, and proceed with TPC/IP processing and device driver execution. By the time the first bits are ready to transmit through the fiber interface, the laser is on and ready to send.

Our results show that SLaC w/OFF turns off all of the stages completely for 54–62% of the whole execution (Figure 12b), and saves 79% of the laser energy compared the No-Ctrl (Figure 11).

6. RELATED WORK

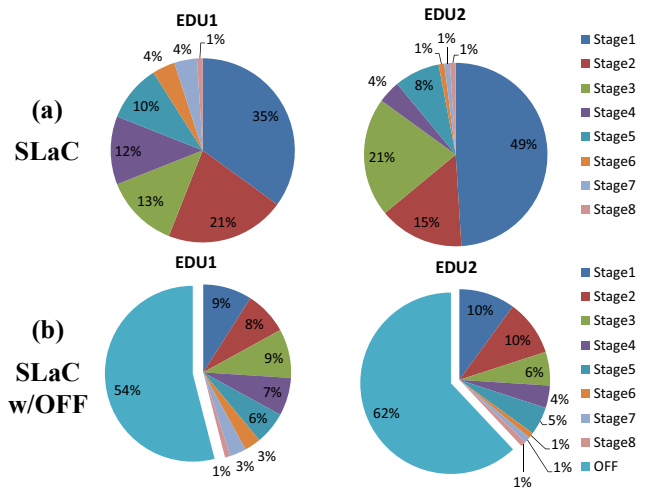Previous studies show that computers rarely operate at full

FIGURE 12. Time spent at each stage level (datacenter).

utilization in both scientific and server computing which leads to an under-utilization of interconnection networks [1,2]. To address this problem, Thonnart *et al.* [43] propose power regulation techniques to reduce the static power consumption in electrical interconnects. They show that powering down the unused asynchronous units results in substantial energy savings. Chen *et al.* [10] propose to power down portions of an electrical on-chip clos network to reduce the static power consumption while providing high performance due to the clos network's high path diversity. Abts *et al.* [1] propose to design an energy-proportional electrical datacenter network that chooses the optimal data rate by monitoring the amount of network traffic.

Silicon photonics have emerged as a promising solution to meet the growing demand for high-bandwidth, low-latency, and energy-efficient communication in manycore and multi-chip processors, as well as large-scale datacenter networks. Recent research proposes laser power-gating to turn off portions of an on-chip interconnect to increase energy efficiency while providing high performance [13,14]. EcoLaser [13] proposes an adaptive laser control scheme for SWMR and MWSR optical crossbars that leaves the laser on for some time after the end of transmission to allow potential senders to transmit opportunistically, without waiting for the turn on delay. As reads of optical packets are destructive, EcoLaser relies on a complicated token design that spans two cycles to communicate the current state of the laser to the other nodes, encode whether another node upstream has opportunistically grabbed the laser, and to allow the same token to transmit a laser turn-on signal if needed. In contrast to EcoLaser, SLaC does not require any special token design. EcoLaser+ [14] employs a different scheme in which it predicts future uses of an on-chip interconnect by co-designing the laser control mechanism with the cache coherence protocol. In contrast to EcoLaser+, SLaC does not require a cache coherence protocol, it is not limited to on-chip applications, and can be employed on a network of any scale, from on-chip, to board-level, to datacenters.

Zhou *et al.* [47] identify as inefficiency the constant laser power consumption when channel utilization is low, and propose a predictive mechanism to increase the average channel utilization. The mechanism controls active splitters to tune channel bandwidth on a binary tree network. Kurian *et al.* [32] propose an optical SWMR crossbar and electrical hybrid interconnection network, and improve performance by utilizing the coherence protocol. [32] mentions that a Ge-based laser can be controlled to improve the laser energy efficiency, but does not present nor evaluate a detailed laser-control scheme. Nitta *et al.* [35] show the energy inefficiency of photonic interconnects under low utilization, and propose to improve efficiency by recapturing the energy of photons which are not used for communication.

Energy-proportionality of scaled-out photonic interconnects

has remained a largely unexplored topic. In this work, we propose SLaC which is a laser control technique for flattened butterfly networks that turns off the majority of the network to save laser energy, while providing high performance. Different that previous work, SLaC works with a highly scalable flattened butterfly network, which means it can be applied to on-chip and multi-chip interconnects, as well as datacenter networks. On top of that, SLaC's performance does not strictly depend on the laser technology, because SLaC always provides full connectivity which removes the laser turn-on latency from the critical path. Using an adaptive routing algorithm, SLaC maximizes the opportunities to save laser energy, while balancing the network traffic and providing high throughput.

7. CONCLUSION

SLaC turns off the majority of a flattened butterfly network when the utilization is low to save energy, and activates additional stages when the utilization is high to provide higher performance. From an on-chip interconnect to a chip-to-chip system to a datacenter network, any network with path diversity can utilize SLaC. Our results show that, for on-chip and multi-chip flattened butterfly, SLaC can save 43–57% of the laser energy on average (up to 67%) while reducing the performance by only 2% on real-world workloads. On a flattened butterfly datacenter network, SLaC saves 79% of the laser energy on average when running traces collected from university datacenter servers.

REFERENCES

[1] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu. Energy proportional datacenter networks. In *Proceedings of the 37th Annual International Symposium on Computer Architecture*, ISCA 10, pages 338- 347, New York, NY, USA, 2010. ACM.

[2] L. A. Barroso and U. Holzle. The case for energy-proportional computing. *IEEE Computer*, 40(12):33- 37, 2007.

[3] C. Batten, A. Joshi, J. Orcutt, A. Khilo, B. Moss, C. W. Holzwarth, M. A. Popovic, H. Li, H. I. Smith, J. L. Hoyt, F. X. Kartner, R. J. Ram, V. Stojanovic, and K. Asanovic. Building many-core processor-to-dram networks with monolithic cmos silicon photonics. *IEEE Micro*, 29(4):8- 21, 2009.

[4] C. Batten, A. Joshi, V. Stojanovic, and K. Asanovic. Designing chip-level nanophotonic interconnection networks. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2(2), 2012.

[5] S. Beamer, K. Asanovic, C. Batten, A. Joshi, and V. Stojanovic. Designing multi-socket systems using silicon photonics. In *Proceedings of the International Conference on Supercomputing (ICS)*, 2009.

[6] T. Benson, A. Akella, and D. A. Maltz. Network traffic characteristics of data centers in the wild. In *Proceedings of the 10th ACM SIG-COMM Conference on Internet Measurement*, IMC 10, 2010.

[7] R. E. Camacho-Aguilera, Y. Cai, N. Patel, J. T. Bessette, M. Romagnoli, L. C. Kimerling, and J. Michel. An electrically pumped germanium laser. *Optics Express*, 20(10):11316- 11320, 2012.

[8] J. Cardenas, C. Poitras, J. Robinson, K. Preston, L. Chen, and M. Lipson. Low loss etchless silicon photonic waveguides. *Optics Express*, 17(6):4752- 4757, 2009.

[9] G. Chen, H. Chen, M. Haurylau, N. Nelson, P. M. Fauchet, E. G. Friedman, and D. H. Albonesi. Electrical and optical on-chip interconnects in scaled microprocessors. In *IEEE International Symposium on Circuits and Systems*, pages 2514- 2517, 2005.

[10] L. Chen, L. Zhao, R. Wang, and T. Pinkston. Mp3: Minimizing performance penalty for power-gating of clos network-on-chip. In *High*

Performance Computer Architecture (HPCA), 2014 IEEE 20th International Symposium on, pages 296- 307, Feb 2014.

[11] M. Cianchetti, N. Sherwood-Droz, and C. Batten. Implementing System-in-Package with Nanophotonic Interconnect. *Workshop on the Interaction between Nanophotonic Devices and Systems*, 2010.

[12] W. J. Dally and T. B. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishing Inc., 2004.

[13] Y. Demir and N. Hardavellas. Ecolaser: An adaptive laser control for energy efficient on-chip photonic interconnects. In *Proceedings of the International Symposium on Low-Power Electronics and Design*, ISLPED 14, August 2014.

[14] Y. Demir and N. Hardavellas. Towards energy-efficient photonic interconnects. In *Proceedings of Optical Interconnects XV, SPIE Photonics West*, February 2015.

[15] Y. Demir, Y. Pan, S. Song, N. Hardavellas, J. Kim, and G. Memik. Galaxy: A high-performance energy-efficient multi-chip architecture using photonic interconnects. In *Proceedings of the 28th ACM International Conference on Supercomputing*, ICS 14, June 2014.

[16] G.-H. Duan, A. Shen, A. Akrout, F. V. Dijk, F. Lelarge, F. Pommereau, O. LeGouezigou, J.-G. Provost, H. Gariah, F. Blache, F. Mallecot, K. Merghem, A. Martinez, and A. Ramdane. High performance inp-based quantum dash semiconductor mode-locked lasers for optical communications. *Bell Labs Technical Journal*, 14(3):63- 84, 2009.

[17] European Semiconductor Industry Association (ESIA), Japan Electronics and Information Technology Industries Association (JEITA), Korean Semiconductor Industry Association (KSIA), Taiwan Semiconductor Industry Association (TSIA), and United States Semiconductor Industry Association (SIA). The international technology roadmap for semiconductors (itrs). http://www.itrs.net/, 2012 Edition.

[18] A. W. Fang, H. Park, O. Cohen, R. Jones, M. J. Paniccia, and J. E. Bowers. Electrically pumped hybrid AlGaInAs-silicon evanescent laser. *Optics Express*, 14(20):9203- 9210, Oct 2006.

[19] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta. Vl2: A scalable and flexible data center network. In *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, SIGCOMM 09, 2009.

[20] N. Hardavellas, S. Somogyi, T. F. Wenisch, R. E. Wunderlich, S. Chen, J. Kim, B. Falsafi, J. C. Hoe, and A. G. Nowatzyk. SimFlex: a fast, accurate, flexible full-system simulation framework for performance evaluation of server architecture. *SIGMETRICS Performance Evaluation Review, Special Issue on Tools for Computer Architecture Research*, 31(4):31- 35, April 2004.

[21] M. Heck and J. Bowers. Energy efficient and energy proportional optical interconnects for multi-core processors: Driving the need for on-chip sources. *Selected Topics in Quantum Electronics, IEEE Journal of*, 20(4):1- 12, July 2014.

[22] H. Hisham, G. Mahdiraji, A. Abas, M. Mahdi, and F. Adikan. Characterization of transient response in fiber grating fabry-perot lasers. *IEEE Photonics Journal*, 4(6):2353- 2371, Dec 2012.

[23] H. Hisham, G. Mahdiraji, A. Abas, M. Mahdi, and F. Adikan. Characterization of turn-on time delay in a fiber grating fabry-perot lasers. *IEEE Photonics Journal*, 4(5):1662- 1678, Oct 2012.

[24] A. Joshi, C. Batten, Y.-J. Kwon, S. Beamer, I. Shamim, K. Asanovic, and V. Stojanovic. Silicon-photonic clos networks for global on-chip communication. In *Proceedings of the IEEE International Symposium on Networks-on-Chip (NOCS)*, pages 124- 133, 2009.

[25] J. Kim, W. J. Dally, and D. Abts. Flattened butterfly: A cost-efficient topology for high-radix networks. In *Proceedings of the 34th Annual International Symposium on Computer Architecture*, ISCA 07, pages 126- 137, June 2007.

[26] L. C. Kimerling. Scaling functionality with silicon photonics: Achievement and potential. http://www.orc.soton.ac.uk/fileadmin/seminar_pdf/UKSP_Showcase_-_Lionel_Kimerling.pdf, November 2013.

[27] N. Kirman, M. Kirman, R. K. Dokania, J. F. Martinez, A. B. Apsel, M. A. Watkins, and D. H. Albonesi. Leveraging optical technology in future bus-based chip multiprocessors. In *Proceedings of the 39th IEEE/ACM Annual International Symposium on Microarchitecture*, MICRO 39, pages 492- 503, 2006.

[28] B. R. Koch, E. J. Norberg, B. Kim, J. Hutchinson, J.-H. Shin, G. Fish, and A. Fang. Integrated silicon photonic laser sources for telecom and datacom. In *Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2013*, page PDP5C.8. Optical Soci-

ety of America, 2013.

[29] P. Koka, M. McCracken, H. Schwetman, X. Zheng, R. Ho, and A. Krishnamoorthy. Silicon-photonic network architectures for scalable, power-efficient multi-chip systems. In *Proceedings of the 37th Annual International Symposium on Computer Architecture*, ISCA 10, pages 117- 128, Saint-Malo, France, 2010. ACM.

[30] E. Kotelnikov, A. Katsnelson, K. Patel, and I. Kudryashov. High-power single-mode ingaasp/inp laser diodes for pulsed operation. *Proceedings of SPIE*, 8277:827715- 827715- 6, 2012.

[31] A. Krishnamoorthy, R. Ho, X. Zheng, H. Schwetman, J. Lexau, P. Koka, G. Li, I. Shubin, and J. Cunningham. Computer systems based on silicon photonic interconnects. *Proceedings of the IEEE*, 97(7):1337 - 1361, july 2009.

[32] G. Kurian, C. Sun, C.-H. Chen, J. Miller, J. Michel, L. Wei, D. Antoniadis, L.-S. Peh, L. Kimerling, V. Stojanovic, and A. Agarwal. Cross-layer energy and performance evaluation of a nanophotonic manycore processor system using real application workloads. In *26th IEEE International Parallel Distributed Processing Symposium (IPDPS)*, pages 1117- 1130, 2012.

[33] S. Larsen, P. Sarangam, and R. Huggahalli. Architectural breakdown of end-to-end latency in a tcp/ip network. In *Computer Architecture and High Performance Computing, 2007. SBAC-PAD 2007. 19th International Symposium on*, pages 195- 202, Oct 2007.

[34] J. Liu, X. Sun, R. Camacho-Aguilera, L. C. Kimerling, and J. Michel. Ge-on-si laser operating at room temperature. *Opt. Lett.*, 35(5):679-681, Mar 2010.

[35] C. Nitta, M. Farrens, and V. Akella. Dcof: An arbitration free directly connected optical fabric. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2(2):169- 182, June 2012.

[36] C. J. Nitta, M. K. Farrens, and V. Akella. *On-Chip Photonic Interconnects: A Computer Architect s Perspective*. Morgan & Claypool Publishers, 2013.

[37] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, and A. Choudhary. Firefly: Illuminating future network-on-chip with nanophotonics. In *Proceedings of the 36th Annual International Symposium on Computer Architecture*, ISCA 09, Austin, TX, 2009.

[38] M. Paniccia and J. Bowers. First electrically pumped hybrid pumped hybrid silicon laser silicon laser. http://www.intel.com/content/dam/www/public/us/en/documents/technology-briefs/intel-labs-hybrid-silicon-laser-announcement.pdf, September 2006.

[39] K. Petermann. *Laser Diode Modulation and Noise*, volume 3 of *Advances in Optoelectronics (ADOP)*. Springer, 1988.

[40] P. Rosenfeld, E. Cooper-Balis, and B. Jacob. Dramsim2: A cycle accurate memory system simulator. *Computer Architecture Letters*, 10(1):16- 19, 2011.

[41] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren. Inside the social network s (datacenter) network. In *Proceedings of the ACM SIGCOMM 2015 Conference on Data Communication*, SIGCOMM 15, pages 123- 137, 2015.

[42] C. Sun, C.-H. O. Chen, G. Kurian, L. Wei, J. Miller, A. Agarwal, L.-S. Peh, and V. Stojanovic. Dsent - a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling. In *6th IEEE/ACM International Symposium on Networks-on-Chip*, pages 201- 210, 2012.

[43] Y. Thonnart, E. Beigne, A. Valentian, and P. Vivet. Automatic power regulation based on an asynchronous activity detection and its application to anoc node leakage reduction. In *14th IEEE International Symposium on Asynchronous Circuits and Systems*, 2008.

[44] D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. P. Jouppi, M. Fiorentino, A. Davis, N. Binkert, R. G. Beausoleil, and J. H. Ahn. Corona: System implications of emerging nanophotonic technology. In *Proceedings of the 35th Annual International Symposium on Computer Architecture*, ISCA 08, pages 153- 164, 2008.

[45] T. F. Wenisch, R. E. Wunderlich, M. Ferdman, A. Ailamaki, B. Falsafi, and J. C. Hoe. SimFlex: statistical sampling of computer system simulation. *IEEE Micro*, 26(4):18- 31, Jul-Aug 2006.

[46] P. Wolf, P. Moser, G. Larisch, W. Hofmann, H. Li, J. Lott, C.-Y. Lu, S. Chuang, and D. Bimberg. Energy-efficient and temperature-stable high-speed VCSELs for optical interconnects. In *15th International Conference on Transparent Optical Networks (ICTON)*, June 2013.

[47] L. Zhou and A. Kodi. Probe: Prediction-based optical bandwidth scaling for energy-efficient nocs. In *Seventh IEEE/ACM International Symposium on Networks on Chip (NoCS)*, pages 1- 8, 2013.