Firefly: Illuminating Future Network-on-Chip with Nanophotonics

Yan Pan, Prabhat Kumar, John Kim[†], Gokhan Memik, Yu Zhang, Alok Choudhary

Northwestern University
2145 Sheridan Road, Evanston, IL
{panyan,prabhat-kumar,g-memik,
yu-zhang,a-choudhary}@northwestern.edu

[†]KAIST
Daejeon, Korea
jjk12@cs.kaist.ac.kr

ABSTRACT

Future many-core processors will require high-performance yet energy-efficient on-chip networks to provide a communication substrate for the increasing number of cores. Recent advances in silicon nanophotonics create new opportunities for on-chip networks. To efficiently exploit the benefits of nanophotonics, we propose Firefly – a hybrid, hierarchical network architecture. Firefly consists of *clusters* of nodes that are connected using conventional, electrical signaling while the inter-cluster communication is done using nanophotonics – exploiting the benefits of electrical signaling for short, local communication while nanophotonics is used only for global communication to realize an efficient on-chip network. Crossbar architecture is used for inter-cluster communication. However, to avoid global arbitration, the crossbar is partitioned into multiple, logical crossbars and their arbitration is localized. Our evaluations show that Firefly improves the performance by up to 57% compared to an all-electrical concentrated mesh (CMESH) topology on adversarial traffic patterns and up to 54% compared to an all-optical crossbar (OP_XBAR) on traffic patterns with locality. If the energy-delay-product is compared, Firefly improves the efficiency of the on-chip network by up to 51% and 38% compared to CMESH and OP_XBAR, respectively.

Categories and Subject Descriptors

C.1.2 [**Computer Systems Organization**]: Multiprocessors—*Interconnection architectures*; B.4.3 [**Hardware**]: Interconnections—*Topology*

General Terms

Design, Performance

Keywords

Interconnection Networks, Topology, Nanophotonics, Hierarchical Network

ISCA'09, June 20–24, 2009, Austin, Texas, USA.
Copyright 2009 ACM 978-1-60558-526-0/09/06 ...$5.00.

1. INTRODUCTION

With the prevalence of dual-core and quad-core processors in the market, researchers could not help projecting a many-core era with tens, hundreds, or even thousands of cores integrated on a single chip [14, 6, 37]. One of the most critical issues in the many-core era will be the communication among different on-chip components. The increasing number of components on a chip calls for efficient Network-on-chip (NoC) designs where data are routed in packets on shared channels instead of dedicated buses [12].

Because of the high latency of on-chip, global communication using conventional RC wires, designers have explored alternative technologies including electrical transmission lines [17], radio frequency (RF) signaling [9], and nanophotonics [38, 24]. While electrical transmission lines and RF signaling both provide low latency, they suffer from low bandwidth density, relatively large components, or electromagnetic interference. Nanophotonics, on the other hand, provides high bandwidth density, low latency, and distance-independent power consumption, which make it a promising candidate for future NoC designs. However, nanophotonics has its own constraints. For example, unlike conventional electrical signaling, static power consumption constitutes a major portion of the total nanophotonic communication power [5].Nanophotonics also come with the additional energy cost for electrical to optical (E/O) and optical to electrical (O/E) signal conversions.

In this paper, we propose the Firefly architecture – a hybrid, hierarchical on-chip network that employs conventional electrical signaling for short/local communication and nanophotonics for long/global traffic. The nanophotonic channels implement a crossbar, but to avoid global switch arbitration, the crossbar is partitioned into multiple, smaller crossbars and the arbitration is localized. By reducing the size of the crossbars, the nanophotonic hardware is significantly reduced while maintaining high-performance. Localized crossbar arbitration requires single-write-multi-read (SWMR) bus structure which can be power inefficient. To overcome this problem, we describe the *reservation-assisted* SWMR design that incurs additional latency but significant reduction in energy consumption. The Firefly architecture is compared against alternative architectures using synthetic traffic patterns and traces from SPLASH2 [40] benchmarks as well as data mining applications [30].

In summary, the contributions of this paper include:

- A hybrid on-chip network architecture that exploits the benefits of both electrical signaling and silicon nanophotonics to improve the efficiency of on-chip networks.

- A scalable topology, Firefly, which supports high throughput with multiple global crossbars efficiently implemented by leveraging nanophotonics and its broadcast capability.

- A thorough evaluation of the Firefly and alternative architectures using synthetic traffic patterns with varying degrees of locality and traces from SPLASH2 and MineBench benchmarks.

The remainder of this paper is organized as follows. In Section 2, we provide background information on silicon nanophotonic technology and state-of-art topologies for on-chip networks. Details of the proposed Firefly network architecture is described in Section 3 along with the routing algorithm and flow control. Performance and energy evaluation is presented in Section 4. Section 5 discusses related work and we conclude the paper in Section 6.

2. BACKGROUND

In this section, we review the relevant nanophotonic components that Firefly exploits to implement an efficient NoC. We also present information on state-of-art on-chip network topology design, with a focus on two alternative topologies that can exploit nanophotonics.

2.1 Nanophotonic devices

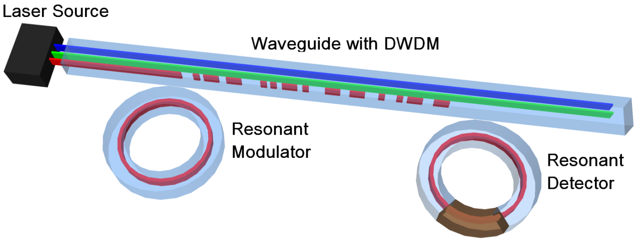


Figure 1: Schematic of nanophotonic devices

The nanophotonic devices needed to support the Firefly architecture include waveguides for routing optical signals, ring modulators for E/O signal conversion, resonant detectors for O/E signal conversion, and a laser source. A schematic representation of these components is shown in Figure 1. Lasers of multiple wavelengths are fed from the laser source into a shared waveguide. The resonant modulator modulates electrical signal onto a specific wavelength, which traverses the waveguide and is absorbed by the resonant detector of that specific wavelength. The resonant detector ring has a Ge-doped section which converts the optical energy into electrical signal. This modulation/detection process does not interfere with the lasers of other wavelengths.

- **Waveguides & Laser Source:** Planar optical waveguides can be fabricated using Si as core and SiO_2 as cladding with transmission loss as low as 3.6dB/cm [39] and good light confinement, allowing for sharp turns (radius of 2um) with minimal loss (0.013dB) [39]. With Dense Wavelength Division Multiplexing (DWDM) technique, lasers of different wavelengths can be transmitted within the same waveguide without interfering with each other. This allows for high bandwidth density and reduced layout complexity. We assume an off-chip laser source [15, 14, 25], which provides 64 wavelengths with low power.

- **Resonant Modulators & Detectors:** DWDM can be realized using ring resonators. The radius of the ring together with thermal tuning decides the specific wavelength it modulates and it can be brought in and out of resonance by charge injection. Such direct modulation can achieve a data rate as high as 12.5Gbps/link [41]. CMOS-compatible Germanium (Ge) can be introduced to dope resonant rings to build selective detectors [31, 42]. Efficient low capacitance detectors can be designed to absorb a fraction of the laser power of its resonant wavelength, enabling broadcast support for the reservation channels in Firefly. Optical splitters can also be used for such multi-cast structures.

2.2 On-chip Network Topologies

2D mesh topology has often been assumed for on-chip networks as it maps well to a 2D VLSI planar layout with low complexity. Different on-chip networks have been built using a 2D mesh topology [6, 37]. However, the 2D mesh topology has several disadvantages, including the need to traverse through large number of intermediate routers. This increases packet latency and results in an inefficient network in terms of power and area [3]. Recent work has shown that the use of concentration and high-radix topology is more efficient for on-chip networks [3, 20]. However, these evaluations were done assuming conventional, electrical signaling. The availability of silicon nanophotonics presents new opportunities in on-chip network architecture. Previous work that incorporate nanophotonics into on-chip networks assume conventional topologies such as a crossbar [38] or torus [33]. In this section, we briefly review two alternative nanophotonic on-chip networks: the Dragonfly topology [21, 22] and Corona [38], an on-chip optical crossbar.

2.2.1 *Dragonfly Topology*

The Dragonfly topology [21, 22] has been recently proposed for large-scale, off-chip networks to exploit the availability of economical, optical signaling technology and high-

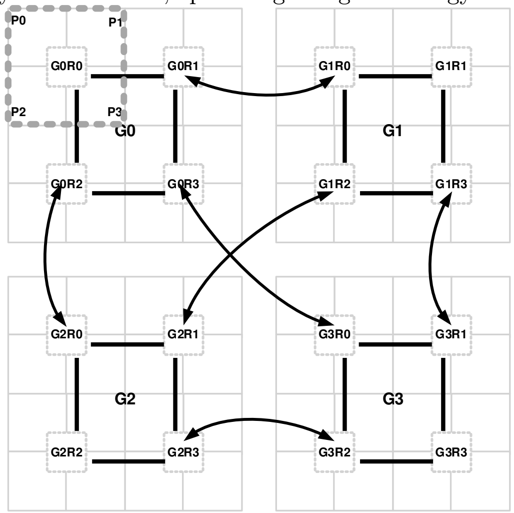
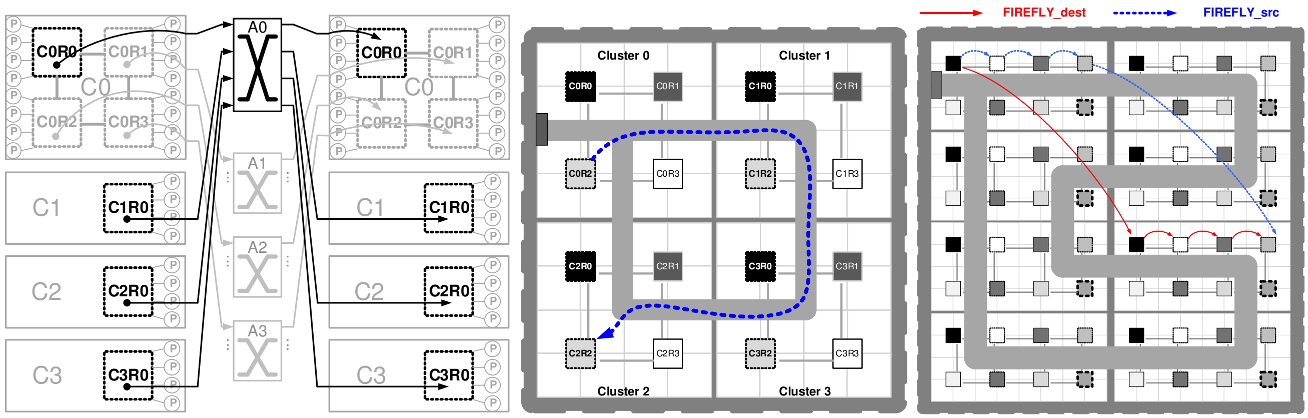


Figure 2: Dragonfly topology mapped to on-chip networks.



(a) (b) (c)

Figure 3: Firefly topology: (a) logical inter-cluster crossbar for 64-core CMP, (b) shared waveguide supporting the inter-cluster crossbars, and (c) waveguide for a 256-core CMP with the routing schemes.

radix routers to create a cost-efficient topology. A schematic view of the topology mapped to on-chip networks is shown in Figure 2, which depicts a 64-core chip assuming a concentration of 4. The 16 routers are divided into 4 groups. Within each group, the routers are electrically connected using mesh topology and each router in the group is connected to a different group through optical signaling.

By creating groups, the effective radix of each router is increased to minimize the cost of the network. However, it relies on *indirect* adaptive routing [16] and multiple global channel traversals for load balancing – resulting in additional complexity to support an extra E/O, O/E conversion. In addition, packets are routed within both the source and destination groups, which increases hop count.

2.2.2 *Corona Architecture*

Corona [38] exploits nanophotonics by using an all-optical crossbar topology. A 64×64 crossbar is implemented with multi-write-single-read optical buses. Each of the 64 buses or channels consists of 4 waveguides, each with 64 wavelengths and each channel is assigned to a different node in the network.

Scaling switch arbitration in a high-radix crossbar presents many challenges [23] but Corona also exploits nanophotonics for their global, switch arbitration by using an optical token-ring arbitration. A token for each node, which represents the right to modulate on each node's wavelength, is passed around all the nodes continuously on a dedicated arbitration waveguide. If a node can grab a token, it absorbs the token, transmits the packet, and then releases the token to allow other nodes to obtain the token. In this paper, the Firefly architecture we propose partitions a large crossbar into multiple, smaller crossbars – avoiding global arbitration by using localized, electrical arbitration done among smaller number of ports. Instead of using multi-write optical buses, the Firefly topology uses multi-read optical buses assisted with reservation broadcasting, which results in a trade-off between additional energy for laser and less hardware.

3. FIREFLY ARCHITECTURE

Firefly is a hierarchical network topology that consists of *clusters* of nodes connected through local, electrical networks, while nanophotonic links are overlaid for global, inter-cluster communication, connecting routers in different clusters, as shown in Figure 3(a). Routers from different clusters that are optically connected to each other form an *assembly* and a crossbar topology is used. Each router is labeled with $CxRy$ where x is the cluster ID and y is the assembly ID – routers with the same x value share the same cluster and communicate through the conventional electrical network while routers with the same y value communicate through the global nanophotonic links. For example, routers C0R0, C1R0, C2R0, and C3R0 in Figure 3(a) form a logical crossbar and are part of Assembly 0 (A0), while C0R0, C0R1, C0R2, and C0R3 are part of Cluster 0 (C0).

3.1 Cluster and Assembly

Since conventional electrical signaling are efficient for short-range communication, electrical signaling is used to create a *cluster* of local nodes. We use a concentrated mesh (CMESH) [3] topology for intra-cluster communication with 4-way concentration – i.e., 4 processors share a single router. We implement *external* concentration [28] instead of increasing router radix to reduce router complexity.

Such a hierarchical network can result in inefficiency for local traffic that crosses the boundaries of clusters. Additional electrical channels can be provided as "stitching" channels to connect all physically neighboring routers – i.e., add a channel between C0R1 and C1R0 in Figure 3(b). However, our analysis shows that stitching increases the number of electrical channels by approximately 40% for a 256-core chip multiprocessors (CMP) while the performance gain was negligible for uniform random traffic pattern. The use of stitching channels also complicates routing. Thus, because of the added complexity with minimal benefits, we do not adopt stitching channels for the Firefly architecture.

3.2 Nanophotonic Crossbar Implementation

The nanophotonic crossbars can be implemented in various ways. One such implementation is single-write-multiple-read (SWMR) nanophotonic buses [24] as shown in Figure 4(a). Each node has a dedicated sending channel (CH_0, CH_1, ..., $CH_{(N-1)}$), which is used to transmit data to other nodes. Each channel consist of multiple waveguides with multiple wavelengths on each waveguide through DWDM – resulting in w bits of data transferred in each cycle. All the nodes on a crossbar are equipped to "listen" on all the sending channels and if the destination of the data packet is the

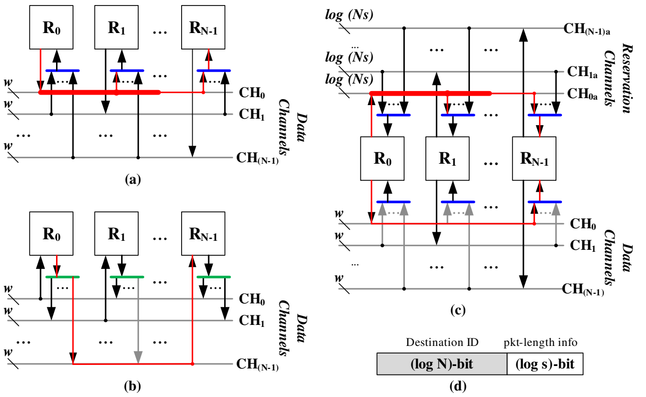


Figure 4: Implementations of a nanophotonic crossbar (a) Single-write-multi-read bus (SWMR), (b) Multi-write-single-read bus (MWSR), (c) Reservation-assisted SWMR (R-SWMR), (d) Reservation flit

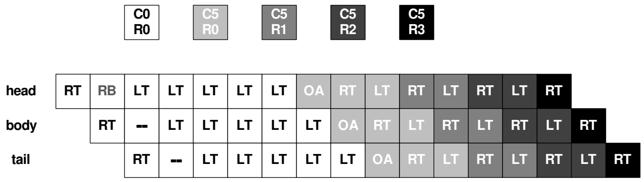
current node, the packet is received. Thus, each node uses one channel to transmit data while having $N-1$ channels to receive from the other $N-1$ nodes.

Another implementation of nanophotonic crossbars is multiple-write-single-read (MWSR) nanophotonic buses as shown in Figure 4(b). Each router "listens" on a dedicated channel and sends on the listening channels of all the other routers. Contention is created when two routers (e.g., R_0 and R_1 in Figure 4(b)) attempt to transmit to the same destination (R_{N-1}) using the same channel ($\mathrm{CH}_{(N-1)}$). Thus, arbitration is required to guarantee that only a single router transmits on a given channel at any moment. The all-optical crossbar (OP_XBAR) we evaluate in Section 4 adopts MWSR and uses token-based arbitration to resolve write contention.

The SWMR and MWSR implementations have their respective pros and cons. SWMR avoids the need for global arbitration by preventing write contention; however, SWMR has higher power consumption. As shown in Figure 4(a), when a router (R_0) sends a packet, it essentially broadcasts to all the other $N-1$ routers, which are continuously coupling energy from the laser of the sending channel (CH_0) to check if they are the destination. Thus, the sender (R_0) has a fan-out of $(N-1)$ and the laser power has to be, in general, $(N-1)\times$ stronger than that of a unicast laser, to activate all the receivers. Extra demodulation power is also consumed during this broadcast process.

3.3 Reservation-assisted SWMR

One possible improvement over the baseline SWMR implementation is to turn off the receiver ring detectors as soon as possible by broadcasting the head flit [1]. Once the head flit is broadcast, all the receivers can compare their ID with the destination ID within the head flit. Non-destination receivers can turn off their detectors and the SWMR bus essentially becomes a unicast channel for the remaining flits in the packet. Thus, theoretically, unicast laser power can be used for the transmission of the remaining flits. However, this method has significant limitations. First, with wide datapath, packets consist of only few flits – thus, broadcasting the head flit is still inefficient. Second, since off-chip laser source is employed and the routers are physically far from the laser sources, it is difficult to regulate the laser power on-the-fly at a flit granularity.



head	RT	RB	LT	LT	LT	LT	LT	OA	RT	LT	RT	LT	RT	LT	RT		
body		RT	--	LT	LT	LT	LT	LT	OA	RT	LT	RT	LT	RT	LT	RT	
tail			RT	--	LT	LT	LT	LT	LT	OA	RT	LT	RT	LT	RT	LT	RT

Figure 5: Pipeline stages for a 3-flit packet from C0R0 to C5R3. Single cycle routers (RT), Reservation Broadcast (RB), link traversal (LT), and optical input arbitration (OA)

To overcome these problems and achieve both localized arbitration and power efficiency, we propose the implementation of *reservation-assisted* SWMR (R-SWMR) buses. Dedicated reservation channels (CH_{0a},CH_{1a},…,$\mathrm{CH}_{(N-1)a}$) are used to *reserve* or establish communication within the assemblies as shown in Figure 4(c). All the receivers are turned off by default. When a router attempts to send a packet, it first broadcasts a reservation flit, which contains the destination and packet length information, to all the other routers within the assembly. Then, only the destination router will tune in on the corresponding data channel to receive the packet in the following cycles, while all the other routers in the assembly will not be coupling laser energy – resulting in point-to-point or unicast communication instead of expensive broadcast on the wider data channels. R-SWMR results in an extra pipeline stage – Reservation Broadcast (RB), as shown in Figure 5. Virtual cut-through flow control [19] is adopted to guarantee that packets are not interleaved once a reservation is established.

Thus, with R-SWMR, we avoid power hungry broadcasting on the wide data channels, but still eliminate the need for global arbitration by broadcasting on the much narrower dedicated reservation channels.

An example is shown in Figure 4(c). For an assembly of size N, with w-bit datapath and supporting s different packet sizes, the reservation flit is $\log N + \log s = \log(Ns)$ bits wide, with $\log N$ bits used for destination identification and $\log s$ bits for packet size information. When R_0 tries to send a packet to R_{N-1}, it first broadcasts on the reservation channel CH_{0a}, to inform R_{N-1} to listen on CH_0 in the following cycles, then the w-bit flits are sent, with unicast power, from R_0 to R_{N-1}. The reservation channels in the R-SWMR architecture introduces overhead in terms of area and energy. The area overhead in terms of additional waveguide is $\log(Ns)/w$ and the static laser power overhead is approximately $(N-1)\log(Ns)/w$. The dynamic E/O and O/E power overhead for reservation flits depends on the packet size t and can be estimated as $\log(Ns)/(wt)$. Based on the parameters that we used in our evaluation in Section 4 ($N=8, s=2, w=256, t=2$), this results in only 1.5% area overhead, 11% static power overhead, as well as 5.5% dynamic power overhead.

3.4 Router Microarchitecture

To support the Firefly architecture, one extra port is required for inter-cluster communication as shown in Figure 6, which highlights the added logic compared to a conventional virtual-channel router. With R-SWMR implementation, each router sends data on a dedicated channel and thus, packets going to any other cluster are switched to the

[1] A packet is partitioned into one or more flits and a packet consists of a head flit, followed by zero or more body flits [11].

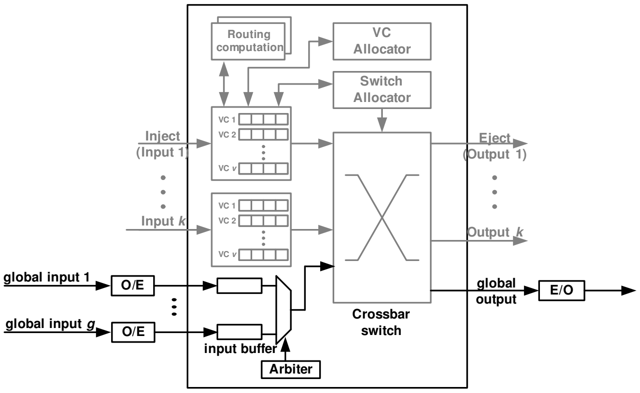


Figure 6: On-chip network router micro-architecture for Firefly.

same router output port for E/O conversion. On the receiver end, each router has separate receivers and buffers for every other router in the same assembly. The detectors of the reservation channels compare the destination ID in the received reservation flit (from all the senders in different clusters), and controls which receivers on the data channels to turn on for O/E conversion and the duration of the reception. Buffered packets, from different clusters are then multiplexed into a single global input port of the router. Round robin arbitration is used for the local arbitration and we conservatively allocate one extra cycle for the arbitration (OA stage in Figure 5).

With this architecture, the inter-cluster crossbar arbitration is localized to the receiver side. While avoiding global bus arbitration, this architecture requires extra buffers. For a cluster size of 8, it requires 1.4× more buffers than a radix-5 virtual-channel router, if the per-VC buffer depth and the number of VCs are held constant.

3.5 Routing and Flow Control

Routing in Firefly consists of two steps: intra-cluster routing and traversing the nanophotonic link. The intra-cluster routing can be done either within the source cluster (FIREFLY_src) or the destination cluster (FIREFLY_dest), as shown in Figure 3(c). For FIREFLY_src, the packet first traverses the electrical links within the source cluster (C0) towards the "take-off" router (C0R3). Then, it traverses the nanophotonic link to reach its final destination (C5R3). The routing steps are reversed with FIREFLY_dest – first traversing the nanophotonic link to reach the destination cluster and then, routing within the destination cluster to reach its destination. A third option (FIREFLY_rand) is to randomize between these two schemes. For both FIREFLY_src and FIREFLY_dest, no additional virtual channel (VC) is needed to avoid routing deadlock but for FIREFLY_rand, 2 VCs are required. Our analysis shows that all the 3 routing schemes show very similar performance and for the rest of this paper, we use FIREFLY_src for evaluation.

Credit-based flow control is used for both local, electrical channels and the global, optical channels to ensure no packets are dropped in the network. Credits are decremented once a flit is ready to be transmitted and sent back upstream through piggybacking. Note that there are multiple buffers at each optical input port before the multiplexer (Figure 6) and their credits are maintained separately and sent upstream to different routers.

3.6 Summary of the Firefly Architecture

Nanophotonic communication provides many benefits such as low latency, high bandwidth density, and repeater-less long range transmission. However, it also presents some new challenges. Various aspects of Firefly are designed to address these challenges to improve the efficiency of the design.

- **Hierarchical Architecture:** Even though nanophotonics can transmit data at the speed of light, it also consumes considerable amount of energy in the form of static and dynamic power dissipation ($0.5pJ/bit$ [5]). Thus, Firefly uses nanophotonics only for long, inter-cluster links, while utilizing economical electrical signaling for local, intra-cluster links. Hence the total hardware and power consumption is reduced.

- **Efficiently Partitioned Optical Crossbars:** Crossbar as a topology has many benefits, including uniform bandwidth and unit network diameter. However, the conventional, electrical crossbar scales poorly, while the all-optical crossbar requires global arbitration. Firefly uses multiple smaller crossbars – eliminating the need for global arbitration and also reducing the hardware complexity. We localize the arbitration for each small crossbar and exploit R-SWMR optical buses to reduce power consumption.

- **Simplifying Routing with Extra Bandwidth:** Instead of relying on adaptive, non-minimal routing which requires multiple E/O, O/E conversions, we leverage the high-bandwidth density provided by nanophotonics and devised a topology that provides scalable inter-cluster bandwidth, which scales up with the number of routers in a cluster.

4. EVALUATION

In this section, we evaluate the performance of Firefly and compare it against alternative architectures using synthetic traffic patterns and traces from SPLASH2 [40] and MineBench [30] benchmarks. We compare energy efficiency of alternative architectures, provide discussion on the impact of datapath width, and discuss how different cost models impact the optimal architecture.

4.1 Simulation Methodology

A cycle accurate network simulator is developed based on the booksim simulator [11, 3] and modified to represent the topologies and routing algorithms that are evaluated. The simulator models both a 4-stage pipelined router [11, 37] and an aggressive, single-cycle router [27]. The total latency of E/O and O/E conversion is reported to be around 75ps [18] and is modeled as part of the nanophotonic link traversal time. Assuming a die size of $400mm^2$, the nanophotonic link traversal time amounts to be 1 to 8 cycles based on the distance between the sender and receiver. Electrical link

Table 1: Simulation configuration

Concentration (# cores per router)	4
Total Buffer per link	1.5KB
Router Pipeline Stages	4-cycle / 1-cycle
Electrical Link Latency	1 cycle
Optical Link Latency (func of dist)	1 – 8 cycles
Data bus width / Flit Size	256-bit
CPU Frequency	5 GHz

Table 2: Evaluated topologies & routing

Code Name	Topology	Global Routing	Min #VC
CMESH	Concentrated mesh	dimension-ordered routing	1
DFLY_MIN	Dragonfly topology mapped to on-chip network	Minimal routing, traversing nanophotonics at most once.	2
DFLY_VAL		Nonminimal routing, traversing nanophotonics up to twice.	3
OP_XBAR	All-optical crossbar using token-based global arbitration	destination-based routing	1
FIREFLY	Proposed hybrid architecture with multiple logical optical inter-cluster crossbar.	Intra-cluster routing in the source cluster before traversing nanophotonics	1

traversal time is modeled as 1 cycle between neighboring routers, as the time to cover the distance is predicted to be 50 ps for 45 nm technology [10]. The clock frequency is targeted at 5GHz. Table 1 summarizes the architectural configuration.

The topologies and routing algorithms evaluated are listed in Table 2. We evaluate a CMP with 256 cores. All topologies implement a concentration factor of 4 – i.e., four processor nodes sharing a single router such that the topologies result in a 64-node network. To reduce the complexity, we assume an *external* implementation of concentration [28]. Because of the complexity of indirect adaptive routing [16] in on-chip networks for a Dragonfly topology, we use both minimal (MIN) routing and non-minimal routing with Valiant's algorithm (VAL) to evaluate the performance of Dragonfly. OP_XBAR uses token-based global arbitration similar to Corona [38]. However, OP_XBAR is not identical to the Corona architecture. For example, in the token arbitration of Corona, multiple requests can be submitted for arbitration in a single cycle to increase the chance of obtaining a token [7]. This arbitration will increase the throughput on traffic patterns such as uniform random compared to our OP_XBAR. However, we assume a single request from four nodes can be submitted for arbitration to simplify the architecture and provide a fair comparison against alternative architectures.

The network traffic loads used for evaluation are listed in Table 3. In addition to load/latency comparisons, we evaluate synthetic workloads to model the memory coherence traffic of a shared memory with each processor generating 100K remote memory operations requests. Once requests are received, responses are generated. We allow 4 outstanding requests per router to mimic the effect of MSHRs – thus, when 4 outstanding requests are injected into the network, new requests are blocked from entering the network until response packets are received. The synthetic traffic patterns used are described in Table 3 and include two traffic patterns (Mix_Lx and Taper_LxDy) that incorporate traffic locality [13].

4.2 Load-Latency Comparison

To compare the throughput of the various topologies, the simulator is warmed up under the specified loads without taking measurements until steady-state is reached. Then a sample of injected packets are labeled during a measurement interval. The simulation is run until all labeled packets exit the system. The performance of the system is measured utilizing the time it takes to process these labeled packets. The total amount of buffer for each port is fixed at 48 flits and is divided into the minimum number of virtual channels needed by each topology/routing algorithm, as listed in Table 2. The cumulative injection rates for the four concentrated processors are used as the load metrics.

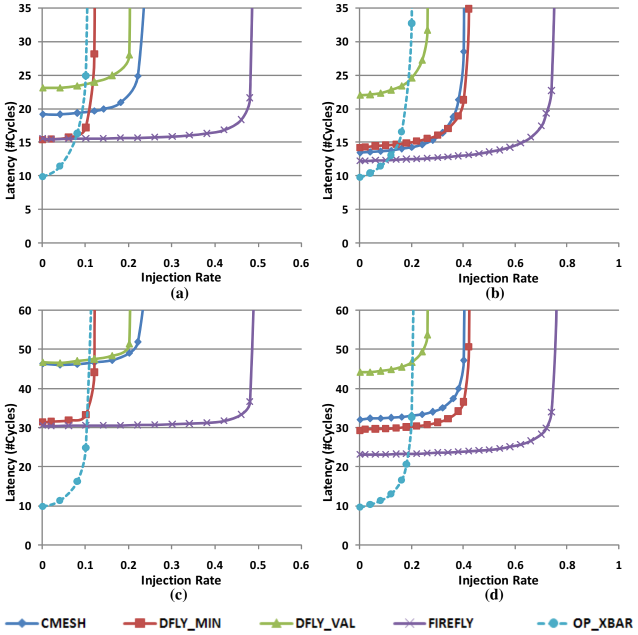


Figure 7: Load latency curve (single-flit pkts) for (a,c) bitcomp and (b,d) uniform traffic using (a,b) single-cycle router and (c,d) 4-cycle router.

Figure 7 shows the results of two synthetic traffic patterns, bitcomp and uniform random. In the comparison, the bisection bandwidth for the topologies using nanophotonics is held constant – i.e., the number of waveguides are identical. However, the amount of optical hardware (e.g., ring modulators) to support the topologies are different: Dragonfly requires approximately $\frac{1}{4}$ the number of rings of the Firefly and $\frac{1}{32}$ that of the OP_XBAR. Firefly exceeds the throughput of Dragonfly (both DFLY_MIN and DFLY_VAL) by at least 70% and OP_XBAR by up to 4.8× because of better utilization of the optical channels. For Dragonfly, the number of global channels in each router needs to be increased to provide sufficient global bandwidth [21]; however, this would require increasing the router radix and complexity and we do not assume this implementation of Dragonfly. The throughput of OP_XBAR is limited by the token based channel arbitration scheme. For example, under uniform random traffic, with single flit packets, each packet has to wait for 4 cycles on average before being sent, hence the throughput is less than 0.25. Alternative arbitration schemes such as generating multiple requests for the token [7] can improve the throughput but would also require additional complexity.

Compared to CMESH, Firefly reduces zero-load latency by 24% and 16% for bitcomp and uniform random traffic, respectively. With use of low-latency nanophotonics, the reduction increases to over 30% if 4-cycle routers are assumed. Despite higher hop count in the Firefly topology compared to OP_XBAR, with single-cycle routers and uniform random traffic, the zero-load latency of Firefly is within 24% of OP_XBAR which does not require any intermediate routers because OP_XBAR has to wait, on average, 4 cycles for the token before traversing the waveguide. However, if the

Table 3: Network loads

Synthetic Traffic Patterns	
Traffic Name	**Details**
Bitcomp	dest id = bit-wise not (src id)
Neighbor	Randomly send to one of the source's neighbors
Transpose	(i,j) => (j,i)
Uniform	Uniform Random traffic
Mix_L*x*	Mixture of intra-cluster and inter-cluster U.R. traffic. *x* is the ratio of intra-cluster traffic.
Taper_L*x* Dy	Mixture of short-range and long-range U.R. traffic. x is the ratio of short-range (manhatton distance < y) traffic.

Traces	
MineBench	**kmeans, scalparc**
SPLASH2	**barnes,cholesky,lu,water_spatial**
Synthetic Load Type	**Details**
Synthetic Workload	100K reqs/node, request & reply inter-dependence. Read_Req & Write_Reply: 8 Bytes Write_Req & Read_Reply: 64 Bytes

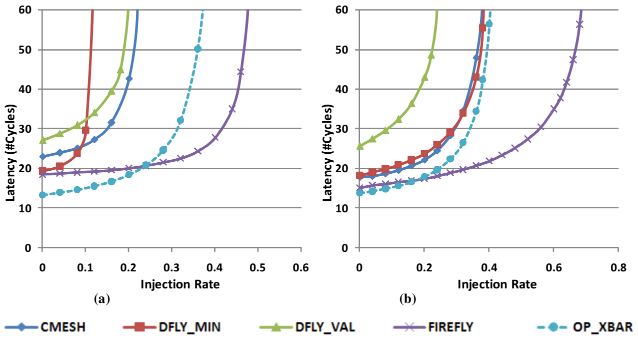


Figure 8: Load latency curves (5-flit Pkts, single-cycle router) (a) bitcomp, (b) uniform

router latency is increased to 4 cycles, Firefly results in 2.4× increase in zero-load latency compared to OP_XBAR. The need for intra-group routing at both the source and destination groups increases the latency of Dragonfly and results in 16% and 26% higher zero-load latency compared to the Firefly for single-cycle and 4-cycle routers, respectively.

4.2.1 *Packet Size Impact*

Figure 8 shows the load-latency curve for the various architectures under traffic with 5-flit packets. Comparing Figure 7 and Figure 8, one interesting observation is that larger packet size results in improved throughput for OP_XBAR. This is because OP_XBAR holds on to the token when sending the multiple flits in a packet and improves the average utilization of the token and channel. However, the localized arbitration of Firefly still allows more than 25% throughput increase compared to OP_XBAR.

4.3 Synthetic Workload Evaluation

Workloads using synthetic traffic patterns are used to compare the different topologies, with completion time of execution used as a metric for comparison (Figure 9). With 256-bit flit size, read requests and write replies are single-flit packets and 64B cache lines in read replies and write requests will require 2 flit packets. Assuming single-cycle routers and with the exception of neighbor traffic, Firefly provides the highest performance across all traffic patterns thanks to its low latency and high throughput. Compared to CMESH, Firefly reduces the execution time by 29% on average, apart from the neighbor traffic, where Firefly suffers from the "boundary" effect as described earlier in Section 3.1.

Compared to OP_XBAR, an average of 40% execution time reduction is achieved with low per-hop latency. However, if 4-cycle routers are assumed, the comparison changes. For traffic with little locality such as bitcomp and transpose permutation traffic, OP_XBAR with a network diameter of one outperforms Firefly by 9% and 17%, respectively. However, because of the hierarchical network of the Firefly topology, Firefly outperforms OP_XBAR by 14% and 22% on highly localized traffic patterns mix_L0.7 and taper_L0.7D7, respectively. The higher hop count of CMESH results in higher performance degradation with 4-cycle router as the Firefly provides up to 51% speedup over CMESH. Performance of Dragonfly heavily relies on the choice of routing scheme for different traffic. Even with a proper routing scheme adopted for Dragonfly, Firefly still achieves around 22% execution time reduction on average (compared with the better of DFLY_MIN and DFLY_VAL). This is because of the 8× inter-cluster bandwidth provided by the Firefly topology.

4.4 Trace-Based Evaluation

Traces from SPLASH2 and MineBench benchmarks are used to compare the performance of the various architectures. We use the average latency of packets injected into the network as a metric in our comparison as shown in Figure 10. Assuming 4-stage routers, Firefly reduces average packet latency by 30% on average compared to CMESH and is within 50% compared to OP_XBAR. With a single-cycle router, the latency is reduced by 32% on average compared to OP_XBAR. For benchmarks such as Scalparc where a hot-spot traffic is created with a single node as the bottleneck, Firefly provides 62% reduction in latency compared to OP_XBAR (27% with 4-cycle router).

4.5 Energy Comparison

4.5.1 *Energy Model*

In this section, we estimate the energy consumption of the various architectures under the same network loads. We model the energy components in Table 4. The functioning of ring modulators and resonators are sensitive to temperature and thus requires external heating [5]. OP_XBAR consists of 8× more micro-rings compared to Firefly, but considering the heat flow, we assume a 4× ring heating power. Similarly, we assume Dragonfly consumes $\frac{1}{3}$ heating power compared to Firefly as it has $\frac{1}{4}$ amount of rings. To establish the communication within an assembly, Firefly needs to broadcast on the reservation channels, as described in Section 3.2, and the laser power for the reservation channels is estimated to be 7× that of a unicast laser. We conservatively ignore the laser power for the token waveguides in OP_XBAR. A 1-to-64 demux is used for OP_XBAR to route flits to the ap-

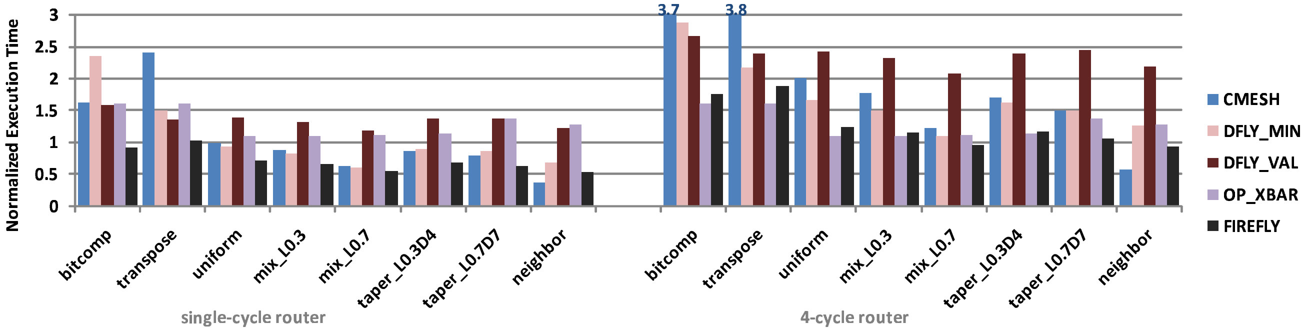


Figure 9: Normalized (with respect to CMESH, uniform traffic, single-cycle router) execution time for synthetic workloads

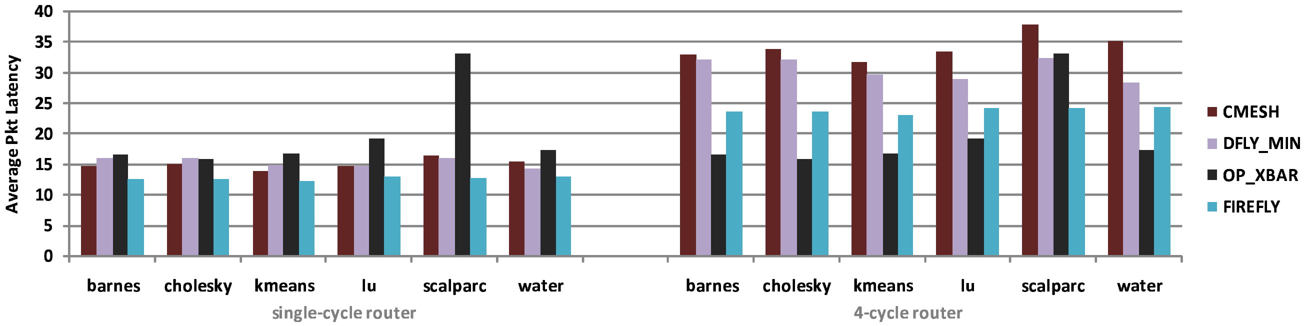


Figure 10: Average Latency for SPLASH2 and MineBench Applications

Table 4: Energy components

Dynamic Power	**Static Power**
Router Buffer	Laser
Router Switch / Demux	Ring Heating
Electrical Link Traversal	
Photonic Modulation & Demodulation	

Table 5: Firefly energy parameters [5, 38]

Flit Size	256 bits
Electrical Router (pJ/flit/hop)	60
Electrical Link (pJ/flit/hop)	38
Optical Transmission (pJ/flit)	40
Chip Laser Power (W)	9.3
Chip Ring Heating Power (W)	3.6

propriate modulator circuit. Dynamic energy consumption in the routers and demux are estimated according to their respective sizes (number of inputs × number of outputs). Electrical leakage power is ignored across all the topologies.

To represent the state-of-art energy consumption of electrical links, we assume the per-hop electrical energy consumption to transmit a 256-bit flit to be $98pJ/hop$ [38]. For the optical components, values reported by Batten et. al. [5] are followed. With these parameters, the resulting energy parameters for Firefly with 256-bit flits are listed in Table 5. The total static optical power (laser and ring heating) is 12.9W for Firefly. The ring heating power of OP_XBAR is estimated to be 4× of Firefly at 14.3W, while its laser power is lower at 8.4W, giving a total of 22.7W. Dragonfly requires even fewer rings than Firefly and its total static optical power comes to 9.6W.

4.5.2 *Synthetic Workload Energy Comparison*

A detailed energy breakdown is shown in Figure 11 for taper_L0.7D7 and bitcomp traffic assuming single-cycle routers. With partitioned crossbars, Firefly reduces the amount of micro-rings on the datapath by a factor of 8 compared to OP_XBAR and thus saves 75% ring heating energy. The reservation broadcasting imposes a 11% static laser power overhead for Firefly as compared to OP_XBAR. However, due to the high throughput and low execution time (i.e., 57% under bitcomp) of Firefly, the average laser energy per packet is 37% lower in Firefly under bitcomp traffic than OP_XBAR. For a similar reason, the per-packet ring heating energy for Firefly is 14% of that in OP_XBAR for bitcomp traffic. Firefly utilizes local electrical meshes, and the associated router and electrical link traversal energy constitutes 65% of the total per-packet energy for bitcomp traffic. But overall, Firefly achieves only a 4% per-packet energy reduction over OP_XBAR under bitcomp traffic. When locality is introduced in the traffic, the reduced average hop-count and reduced optical link traversal in Firefly results in much lower per-packet energy consumption. Thus, for taper_L0.7D7, Firefly reduces per-packet energy consumption by 34% over OP_XBAR.

CMESH is also sensitive to traffic locality. For global traffic like bitcomp, the large hop counts results in low energy efficiency. For such traffic patterns, CMESH performs worse than all the optical alternatives. When locality is available, the efficiency of CMESH significantly improves. However, with taper_L0.7D7, Firefly still achieves 14% lower per-packet energy consumption than CMESH, while 18% reduction is achieved for bitcomp traffic. Dragonfly topology requires less optical hardware than Firefly and thus consumes lower heating power. However, VAL routing requires traversing nanophotonic links twice while minimal routing does not exploit path diversity and results in poor performance – which translates into increased static energy consumption.

Figure 12 shows the energy consumption for a wide range of synthetic workloads. In general, OP_XBAR is more effi-

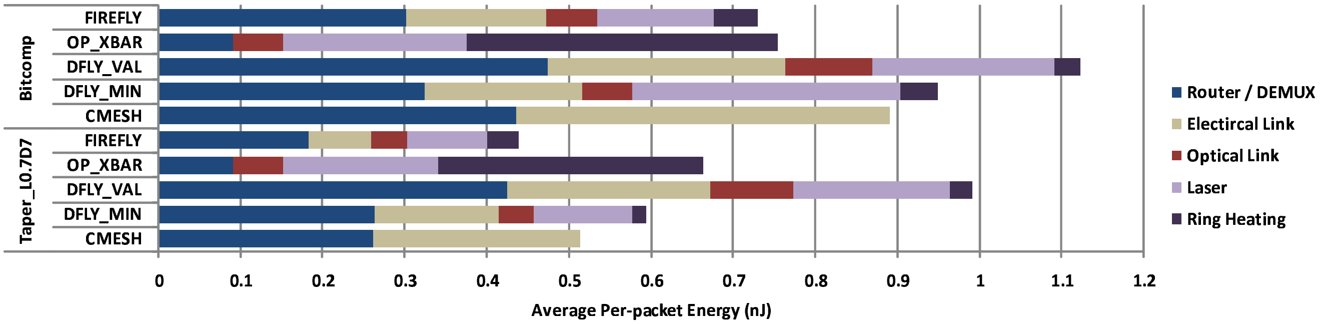


Figure 11: Energy breakdown for bitcomp and taper_L0.7D7 traffic (1-cycle router)

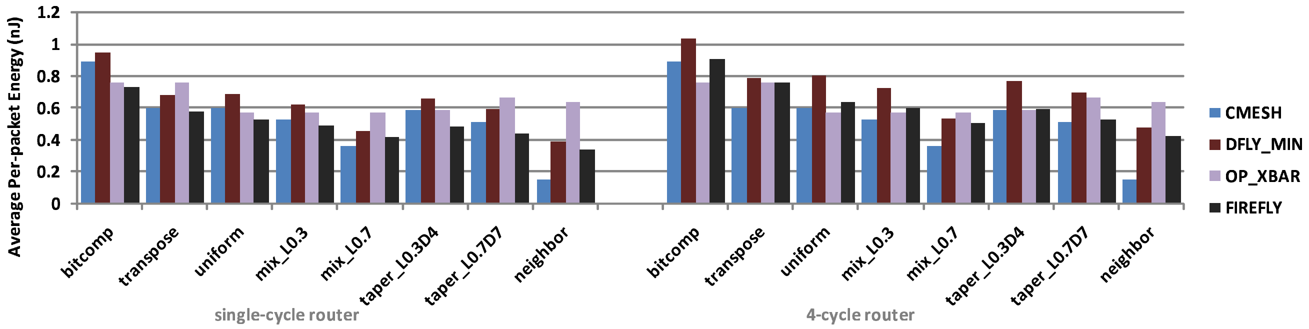


Figure 12: Average per-packet energy consumption for synthetic workloads

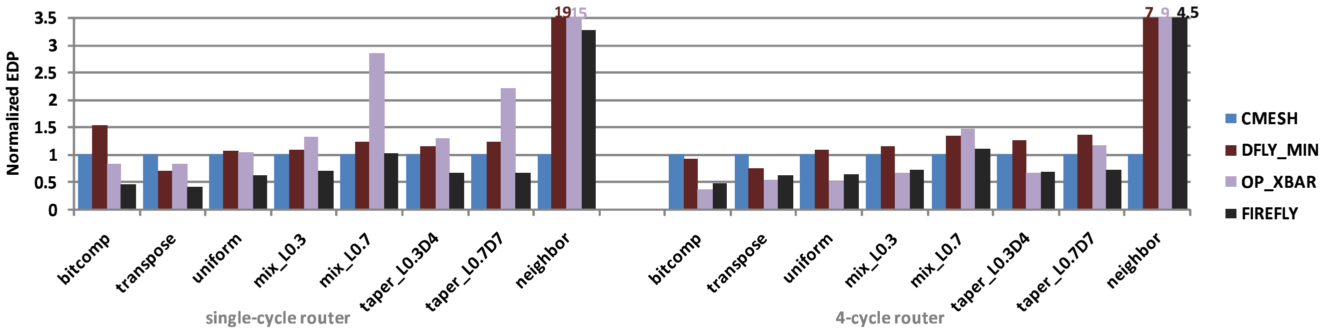


Figure 13: Energy delay product for synthetic workloads normalized to CMESH

cient for global traffic patterns (with 4-cycle routers, OP_XBAR is most efficient for bitcomp and uniform traffic), while CMESH and Firefly are more efficient for traffic with locality. On average, Firefly reduces per-packet energy consumption by 4% over OP_XBAR with 4-cycle routers (21% if single-cycle routers are used). Compared with CMESH, with single-cycle routers, Firefly consumes 8% less energy per packet on average except for neighbor traffic.

4.5.3 *Synthetic Workload Energy Delay Product Comparison*

In addition to performance and energy consumption, we compare the efficiency of the alternative topologies by using the Energy-Delay-Product (EDP) (= Total Energy × Total Execution Time) metric as shown in Figure 13. With 4-cycle routers, OP_XBAR is most efficient for global traffic patterns (bitcomp, transpose, and uniform), and has on average 25% lower EDP than Firefly. However, with locality in the traffic, Firefly reduces EDP by up to 38% compared to OP_XBAR on mix and taper traffic patterns. Firefly also reduces EDP by up to 51% compared to CMESH on all traffic patterns except neighbor traffic. By reducing the per-hop latency to a single-cycle router, Firefly is the most efficient across all non-neighbor traffic patterns – achieving EDP reduction by up to 64% compared to OP_XBAR, and up to 59% compared to CMESH.

4.6 Impact of Datapath Width

The wide datapath of on-chip networks are used to exploit abundant on-chip bandwidth; however, a wider datapath can also increase the cost of the network despite its higher performance. In Figure 14, we compare alternative architectures as we vary the width of the datapath for bitcomp and uniform random traffic patterns. The impact of datapath width across the architectures and traffic patterns is similar for both performance and energy cost. Reducing datapath width increases the serialization latency and thus, reduces performance. However, the energy per packet is also lowered with narrower datapath because of the reduced static and dynamic power consumption. For exam-

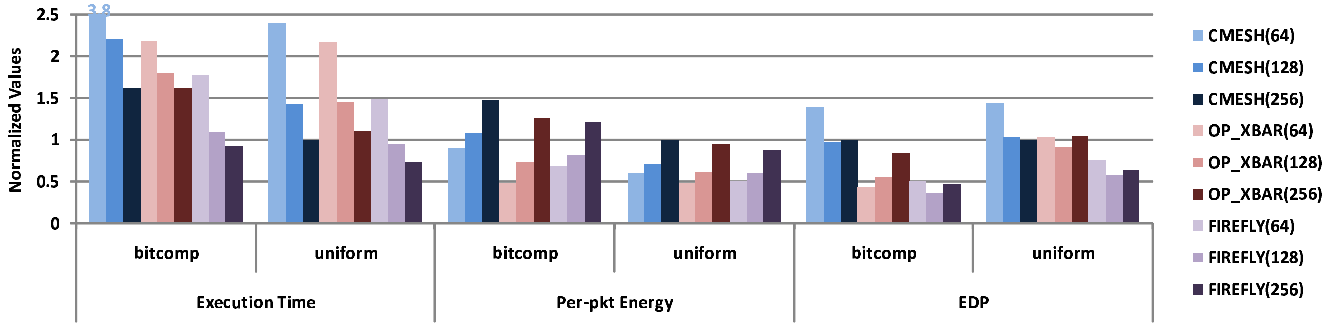


Figure 14: Datapath width trade-off. Execution time and per-pkt energy are normalized to 256-bit CMESH under uniform random traffic. EDP is normalized to 256-bit CMESH under each traffic pattern.

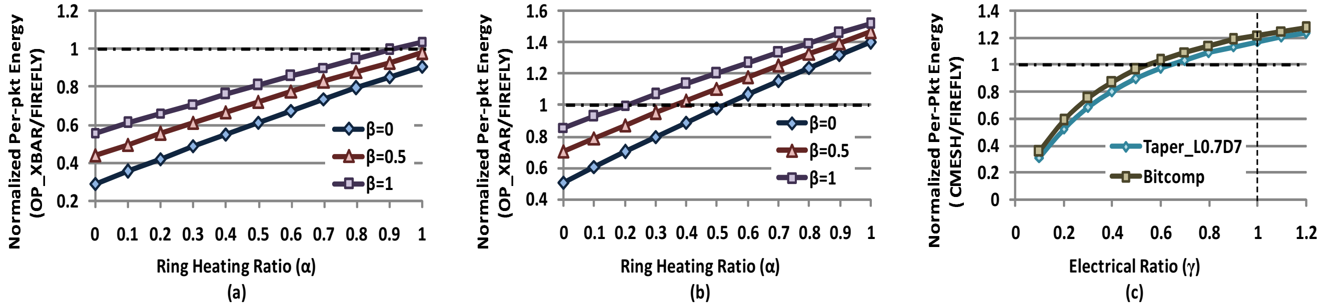


Figure 15: Technology sensitivity comparing OP_XBAR vs. Firefly under (a) bitcomp and (b) taper_L0.7D7 traffic and (c) CMESH vs. Firefly under taper_L0.7D7 and bitcomp

ple, when reducing the datapath width by 2× (from 256 to 128 bits), the performance of Firefly is reduced by up to 23% while also reducing the energy per-packet by up to 33%. When the EDP of the different datapath width are compared, the optimal datapath width of each architecture vary. For example, for all-electrical CMESH, wider datapath is more efficient for both bitcomp and UR because of the significant increase in performance. On the other hand, for all-optical OP_XBAR, narrower datapath are more efficient for bitcomp as the performance benefit of increased datapath width is much smaller. For the Firefly topology, a more efficient architecture can be achieved by halving the datapath from 256 to 128 bits and reducing EDP by up to 20%

4.7 Technology Sensitivity Study

The comparison of alternative architectures was based on the technology parameters described in Section 4.1. However, as technology continues to evolve, technology parameters will change and impact the energy cost of the various topologies in different ways. In this section, we vary the energy cost of critical nanophotonic and electrical components and evaluate their impact on the efficiency of Firefly.

To evaluate the impact of nanophotonic technology, we use the following two parameters :

$$\text{Ring Heating Ratio } (\alpha) = \frac{\text{Future per-ring heating power}}{\text{Current per-ring heating power}}$$

$$\text{Laser Ratio } (\beta) = \frac{\text{Future unicast laser power}}{\text{Current unicast laser power}}$$

as they represent a significant component of nanophotonic energy consumption. Although optical modulation and demodulation energy will change as technology evolves, for simplicity, we assume that they do not scale. The energy consumed by the electrical network is also assumed to be constant for this comparison. Figure 15(a, b) shows the comparison of OP_XBAR and Firefly where the y-axis is the per-packet energy of the OP_XBAR normalized to that of Firefly. Any value greater than 1 on the y-axis represents technology parameters which results in Firefly consuming lower energy. With the parameters used in Section 4.1 ($\alpha = \beta = 1$), OP_XBAR consumes 3.8% more energy than Firefly on bitcomp. However, our analysis shows that for $\alpha \leq 0.9$ or $\beta \leq 0.7$, OP_XBAR will consume lower energy. As the cost of nanophotonics is reduced with lower α and β values, an all-optical architecture will be more efficient. However, for traffic with locality, such as taper_L0.7D7 (Figure 15(b)), the power budget of ring heating needs to be reduced by 80% (i.e., α reduced to 0.2) for OP_XBAR to consume less energy than Firefly.

In order to study the effect of scaling of electrical technology, we keep the cost of nanophotonics constant and vary the relative cost of electrical technology – electrical ratio, γ, as follows:

$$\text{Electrical Ratio } (\gamma) = \frac{\text{Future per-hop electrical energy}}{\text{Current per-hop electrical energy}}$$

As shown in Figure 15(c), the traffic locality does not have a significant impact in comparing CMESH and Firefly as γ is changed. However, as γ is decreased, the per-hop energy cost of electrical network is reduced. With the energy cost of nanophotonic assumed to remain constant, if the cost of electrical per-hop energy cost is reduced by more than 40%, CMESH will consume lower energy per packet compared to Firefly.

5. RELATED WORK

Optical signaling has been widely used in long-haul networks because of its low-latency and high-bandwidth [1]. Optical signaling has also been proposed in multicomputers [8, 29, 34], but not widely used due to its high cost. However, recent advances in economical optical signaling have enabled off-chip networks with longer channels and topologies such as the Dragonfly topology [21, 22].

Recent advances in optical signaling [2, 4, 32, 36] have made the use of on-chip optical signaling a possibility. Different on-chip network architectures have been proposed to exploit silicon nanophotonics including Corona [38] architecture described earlier in Section 2.2. A crossbar structure has also been proposed by Batten et al. [5] to connect a many-core processor to the DRAM memory using monolithic silicon. Their work focuses on core-to-memory communication whereas the Firefly exploit nanophotonics for intra-chip communication. Kirman et al. [24] proposed a 64-node CMP architecture which takes of advantage of nanophotonics to create an on-chip bus and results in a hierarchical, multi-bus interconnect. However, since the optical signaling is used as a bus, the on-chip network is not scalable as the network size increases. Shacham et al. [35] proposed using an electrical layer for control signals while the channels and the switches used to transmit data are done in optical signaling. The resulting network uses conventional on-chip network topology such as a 2D mesh/torus topology and creates a circuit switched network. However, since the size of the packets are relatively small compared to the size of the channel width, using circuit switching is not efficient for on-chip networks. In addition, zero-load latency is increased on all packets with the need to setup a circuit.

Chang et al. [9] use radio frequency (RF) signaling interconnect to reduce the latency of global communication in on-chip networks. The proposed topology used a 2D mesh network overlaid with a RF interconnect and frequency division multiplexing to increase the effective bandwidth. However, this overlay approach creates an asymmetric topology and requires complicated deadlock avoidance scheme to recover from deadlock. Krishna et al. [26] also proposed a hybrid approach to interconnect design by using multi-drop wires with low-latency in addition to conventional electrical signaling. They share a similar objective as our proposed architecture in exploiting characteristics of different interconnect but use the low-latency interconnect for control signals only. In addition, they use a 2D mesh topology with advanced flow control mechanism (express virtual channel) to improve efficiency while this work describes an alternative topology to exploit nanophotonics. Partitioning a high-radix crossbar into multiple, smaller crossbar was proposed in the micro-architecture of a high-radix router to create a hierarchical crossbar [23]. The nanophotonic crossbar in the Firefly is similar to the hierarchical crossbar but we exploit the benefits of nanophotonic to provide uniform global bandwidth between all clusters.

6. CONCLUSION

In this work, we proposed a hybrid, hierarchical on-chip network architecture that utilizes both optical signaling and conventional, electrical signaling to achieve an energy-efficient on-chip network. The multiple, locally arbitrated optical crossbars are used for global communication and an electrical concentrated mesh is used for local, intra-cluster communication. This hierarchical topology results in a scalable on-chip network that provides higher performance while minimizing energy consumption. Compared to an all-electrical concentrated mesh topology, Firefly improves performance by up to 57% while improving the efficiency (in terms of EDP) by 51% on synthetic workload with adversarial traffic patterns. Compared to an all-optical crossbar, Firefly improves performance by 54% and efficiency by 38% on traffic patterns with locality.

Acknowledgements

This work is supported by NSF grants CNS-0551639, IIS-0536994, CCF-0747201, NSF HECURA CCF-0621443, NSF SDCI OCI-0724599 and CCF-0541337; DoE CAREER Award DEFG02-05ER25691; and by Wissner-Slivka Chair funds. We would like to thank Nathan Binkert, Jungho Ahn, Norm Jouppi, and Robert Schreiber for their help in understanding the Corona architecture and their feedback on the paper, and Hooman Mohseni for his comments on our work. We would also like to thank all the anonymous referees for their detailed comments. This work was done while John Kim was affiliated with Northwestern University.

References

[1] A. Al-Azzawi. *Photonics: Principles and Practices*. CRC Press, 2007.

[2] V. Almeida, C. Barrios, R. Panepucci, M. Lipson, M. Foster, D. Ouzounov, and A. Gaeta. All-optical switching on a silicon chip. *Optics Letters*, 29:2867–2869, 2004.

[3] J. Balfour and W. J. Dally. Design tradeoffs for tiled CMP on-chip networks. In *Proc. of the International Conference on Supercomputing (ICS)*, pages 187–198, Carns, Queensland, Australia, 2006.

[4] C. A. Barrios, V. R. Almeida, and M. Lipson. Low-power-consumption short-length and high-modulation-depth silicon electro-optic modulator. *Journal of Lightwave Technology*, 21(4):1089–1098, 2003.

[5] C. Batten, A. Joshi, J. Orcutt, A. Khilo, B. Moss, C. Holzwarth, M. Popovic, H. Li, H. Smith, J. Hoyt, F. Kartner, R. Ram, V. Stojanovic, and K. Asanovic. Building manycore processor-to-dram networks with monolithic silicon photonics. In *Proc. of Hot Interconnects*, pages 21–30, Stanford, CA, 2008.

[6] S. Bell, B. Edwards, J. Amann, R. Conlin, K. Joyce, V. Leung, J. Mackay, M. Reif, L. Bao, J. Brown, M. Mattina, C.-C. Miao, C. Ramey, D. Wentzlaff, W. Anderson, E. Berger, N. Fairbanks, D. Khan, F. Montenegro, J. Stickney, and J. Zook. Tile64 processor: A 64-core soc with mesh interconnect. In *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*, pages 88–598, 2008.

[7] N. Binkert. Personal communication, Aug. 2008.

[8] R. D. Chamberlain, M. A. Franklin, and C. S. Baw. Gemini: An optical interconnection network for parallel processing. *IEEE Trans. on Parallel and Distributed Systems*, 13:1038–1055, 2002.

[9] M. Chang, J. Cong, A. Kaplan, M. Naik, G. Reinman, E. Socher, and S.-W. Tam. CMP network-on-chip overlaid with multi-band rf-interconnect. In *International Symposium on High-Performance Computer Architecture (HPCA)*, pages 191–202, Feb. 2008.

[10] G. Chen, H. Chen, M. Haurylau, N. Nelson, P. M. Fauchet, E. Friedman, and D. Albonesi. Predictions of cmos

compatible on-chip optical interconnect. In *7th International Workshop on System-Level Interconnect Prediction (SLIP)*, pages 13–20, San Francisco, CA, 2005.

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

[12] W. J. Dally and B. Towles. Route packets, not wires: on-chip inteconnection networks. In *Proc. of Design Automation Conference (DAC)*, pages 684–689, Las Vegas, NV, Jun 2001.

[13] R. Das, S. Eachempati, A. Mishra, V. Narayanan, and C. Das. Design and evaluation of a hierarchical on-chip interconnect for next-generation CMPs. In *International Symposium on High-Performance Computer Architecture (HPCA)*, pages 175–186, Raleigh, NC, USA, Feb. 2009.

[14] P. Gratz, C. Kim, R. McDonald, S. Keckler, and D. Burger. Implementation and evaluation of on-chip network architectures. In *International Conference on Computer Design (ICCD)*, pages 477–484, San Jose, CA, 2006.

[15] A. Gubenko, I. Krestnikov, D. Livshtis, S. Mikhrin, A. Kovsh, L. West, C. Bornholdt, N. Grote, and A. Zhukov. Error-free 10 gbit/s transmission using individual fabry-perot modes of low-noise quantum-dot laser. *Electronic Letters*, 43(25):1430–1431, 2007.

[16] N. Jiang, J. Kim, and W. J. Dally. Indirect adaptive routing on large scale interconnection networks. In *Proc. of the International Symposium on Computer Architecture (ISCA)*, Austin, TX, 2009.

[17] A. Jose and K. Shepard. Distributed loss-compensation techniques for energy-efficient low-latency on-chip communication. *Solid-State Circuits, IEEE Journal of*, 42(6):1415–1424, June 2007.

[18] P. Kapur and K. C. Saraswat. Comparisons between electrical and optical interconnects for on-chip signaling. In *International Interconnect Technology Conference*, pages 89–91, Burlingame, CA, Jun. 2002.

[19] P. Kermani and L. Kleinrock. Virtual cut-through: A new computer communication switching technique. *Computer Networks*, 3:267–286, 1979.

[20] J. Kim, J. Balfour, and W. J. Dally. Flattened butterfly topology for on-chip networks. In *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Chicago, Illinois, Dec. 2007.

[21] J. Kim, W. J. Dally, S. Scott, and D. Abts. Technology-driven, highly-scalable dragonfly network. In *Proc. of the International Symposium on Computer Architecture (ISCA)*, Beijing, China, 2008.

[22] J. Kim, W. J. Dally, S. Scott, and D. Abts. Cost-efficient dragonfly topology for large-scale system. In *Micro's Top Picks in Computer Architecture Conferences*, volume 29, pages 33–40, 2009.

[23] J. Kim, W. J. Dally, B. Towles, and A. Gupta. Microarchitecture of a high-radix router. In *Proc. of the International Symposium on Computer Architecture (ISCA)*, Madison, WI, Jun. 2005.

[24] 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 *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 492–503, Orlando, FL, 2006.

[25] A. Kovsh, I. Krestnikov, D. Livshits, S. Mikhrin, J. Weimert, and A. Zhukov. Quantum dot laser with 75nm broad spectrum of emission. *Optics Letters*, 32(7):793–795, 2007.

[26] T. Krishna, A. Kumar, P. Chiang, M. Erez, and L.-S. Peh. Noc with near-ideal express virtual channels using global-line communication. In *Proc. of Hot Interconnects*, pages 11–20, Stanford, CA, 2008.

[27] A. Kumar, P. Kundu, A. P. Singh, L.-S. Peh, and N. K. Jha. A 4.6tbits/s 3.6ghz single-cycle NoC router with a novel switch allocator in 65nm CMOS. In *International Conference on Computer Design (ICCD)*, pages 63–70, Lake Tahoe, CA, 2007.

[28] P. Kumar, Y. Pan, J. Kim, G. Memik, and A. Choudhary. Exploring concentration and channel slicing in on-chip network router. In *IEEE International Symposium on Network-on-Chip (NOCS)*, San Diego, CA, 2009.

[29] A. Louri and A. K. Kodi. An optical interconnection network and a modified snooping protocol for the design of large-scale symmetric multiprocessors (smps). *IEEE Trans. on Parallel and Distributed Systems*, 15(12):1093–1104, Dec. 2004.

[30] R. Narayanan, B. Ozisikyilmaz, J. Zambreno, G. Memik, and A. Choudhary. Minebench: A benchmark suite for data mining workloads. In *IEEE International Symposium on Workload Characterization (IISCW)*, pages 182–188, San Jose, CA, 2006.

[31] O.I.Dosunmu, M. K. Emsley, M. S. Unlu, D. DCannon, and L. C. Kimerling. High speed resonant cavity enhanced ge photodetectors on si reflecting substrates for 1550 nm operation. In *IEEE International Topical Meeting on Microwave Photonics, 2004.*, pages 266–268, Ogunquit, ME, 2004.

[32] L. Pavesi and D. J. Lockwood. Silicon photonics, 2004.

[33] M. Petracca, K. Bergman, and L. Carloni. Photonic networks-on-chip: Opportunities and challenges. pages 2789–2792, Seattle, WA, May 2008.

[34] T. Pinkston. Design considerations for optical interconnects in parallel computers. In *Proc. of the First International Workshop on Massively Parallel Processing Using Optical Interconnections*, pages 306–322, Cancun, Mexico, Apr. 1994.

[35] A. Shacham, K. Bergman, and L. P. Carloni. The case for low-power photonic networks-on-chip. In *Proc. of Design Automation Conference (DAC)*, pages 132–135, San Diego, CA, 2007.

[36] J. Tatum. Vcsels for 10 gb/s optical interconnects. In *IEEE Emerging Technologies Symposium on BroadBand Communications for the Internet Era*, pages 58–61, Richardson, TX, 2001.

[37] S. R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar. An 80-Tile sub-100-w teraflops processor in 65-nm cmos. *Solid-State Circuits, IEEE Journal of*, 43(1):29–41, 2008.

[38] D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. P. Jouppi, M. Fiorentino, A. Davis, N. L. Binkert, R. G. Beausoleil, and J. H. Ahn. Corona: System implications of emerging nanophotonic technology. In *Proc. of the International Symposium on Computer Architecture (ISCA)*, pages 153–164, Beijing, China, 2008.

[39] Y. Vlasov and S. McNab. Losses in single-mode silicon-on-insulator strip waveguides and bends. *Optics Express*, 12(8):1622–1631, 2004.

[40] S. Woo, M. Ohara, E. Torrie, J. Singh, and A. Gupta. The SPLASH-2 programs: Characterization and methodological considerations. In *Proc. of the International Symposium on Computer Architecture (ISCA)*, pages 24–36, Santa Margherita Ligure, Italy, Jun. 1995.

[41] Q. Xu, S. Manipatruni, B. Schmidt, J. Shakya, and M. Lipson. 12.5 gbit/s carrier-injection-based silicon micro-ring silicon modulators. *Opt. Express*, 15(2):430–436, Jan. 2007.

[42] T. Yin, R. Cohen, M. M. Morse, G. Sarid, Y. Chetrit, D. Rubin, and M. J. Paniccia. 40gb/s ge-on-soi waveguide photodetectors by selective ge growth. In *Conference on Optical Fiber communication/National Fiber Optic Engineers Conference (OFC/NFOEC)*, pages 24–28, San Diego, CA, 2008.