2012 IEEE/ACM 45th Annual International Symposium on Microarchitecture

Dynamic Reconfiguration of 3D Photonic Networks-on-Chip for Maximizing Performance and Improving Fault Tolerance

Randy Morris †, Avinash Karanth Kodi †, and Ahmed Louri ‡
†Electrical Engineering and Computer Science, Ohio University, Athens, OH 45701
‡Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721
rm700603@ohio.edu, kodi@ohio.edu, louri@email.arizona.edu

Abstract

As power dissipation in future Networks-on-Chips (NoCs) is projected to be a major bottleneck, researchers are actively engaged in developing alternate power-efficient technology solutions. Photonic interconnects is a disruptive technology solution that is capable of delivering the communication bandwidth at low power dissipation when the number of cores is scaled to large numbers. Similarly, 3D stacking is another interconnect technology solution that can lead to low energy/bit for communication. In this paper, we propose to combine photonic interconnects with 3D stacking to develop a scalable, reconfigurable, power-efficient and high-performance interconnect for future many-core systems, called R-3PO (Reconfigurable 3D-Photonic Networks-on-Chip). We propose to develop a multi-layer photonic interconnect that can dynamically reconfigure without system intervention and allocate channel bandwidth from less utilized links to more utilized communication links. In addition to improving performance, reconfiguration can re-allocate bandwidth around faulty channels, thereby increasing the resiliency of the architecture and gracefully degrading performance. For 64-core reconfigured network, our simulation results indicate that the performance can be further improved by 10%-25% for Splash-2, PARSEC and SPEC CPU2006 benchmarks, where as simulation results for 256-core chip indicate a performance improvement of more than 25% while saving 6%-36% energy when compared to state-of-the-art on-chip electrical and optical networks.

1. Introduction

Future projections based on ITRS roadmap indicates that complementary metal oxide semiconductor (CMOS) feature sizes will shrink to sub-nanometer within a few years, and we could possibly have as many as 256 cores on-chip by the next decade. While Networks-on-Chip (NoC) design paradigm offers modular and scalable performance, increasing core counts leads to increase in serialization latency and power dissipation as packets are processed at many intermdediate routers. Many electronic NoC designs such as Flattened butterfly [12], concentrated mesh and MECS topologies [10] provide express channels to avoid excess hops between distant nodes. While metallic interconnects can provide the required bandwidth due to abundance of wires in on-chip networks, ensuring high-speed inter-core communication within the allocated power budget in the face of technology scaling (and increased leakage currents) will become a major bottleneck for future multicore designs [4].

Emerging technologies such as photonic interconnects and 3D stacking are under serious consideration for meeting the communication challenges posed by the multicores. Photonic interconnects provides several advantages such as: (1) bit rates independent of distance, (2) higher bandwidth due to multiplexing of wavelengths, (3) larger bandwidth density by multiplexing wavelengths on the same waveguide/fiber, (4) lower power by dissipating only at the endpoints of the communication channel and many more [29, 3, 23]. Similarly, 3D stacking of multiple layers have shown to be advantageous due to (1) shorter inter-layer channel, (2) reduced number of hops and (3) increased bandwidth density. A prevalent way to connect 3D interconnects is to use TSVs (through-silicon vias), micro-bump or flip-chip bonding. The pitch of these vertical vias is very small (4μm$\sim10\mu$m), and delays on the order of 20 ps for a 20-layer stack. Most prior photonic interconnect are 2D designs that are plagued with high optical losses due to waveguide crossings or long snake-like waveguides that coil around the chip to prevent waveguide crossings altogether. For example, a photonic channel with 100 waveguide crossings will have a -5 dB loss if we assume a -0.05 dB loss per waveguide crossing [3]. [1] 3D stacking can avoid waveguide crossings and enable efficient stacking of multiple optical layers to design power-efficient topologies. Jalali's group at UCLA has fabricated a SIMOX (Separation by IMplantation of Oxygen) 3D sculpting to stack optical devices in multiple layers [16]. Lipson group at Cornell has successfully buried active optical ring modulator in polyscrystalline silicon [24]. Moreover, recent work on using silicon nitride has shown the possibility of designing multi-layer 3D integration of photonic layers with layer-to-layer optical losses as low as 0.1 dB [5].

With an emerging technology such as photonic interconnects, it is essential to realize that the hardware cost of designing large scale fully photonic networks requires a substantial investment.[2] Therefore, energy, hardware and architecture limitations could force future designs to limit the number of photonic components at the on-chip level. Moreover, the static channel allocation (wavelengths, waveguides) proposed for most photonic interconnects can provide good performance for uniform traffic, however, for non-uniform and temporal and spatial varying traffic as seen in real traffic, the static allocation could limit the network throughput. Moreover, in case of faults in the channel either due to photonic device or electronic backend circuitry failure, communication can breakdown isolating otherwise healthy cores. However, if the network itself could determine the current load on a channel and re-allocate bandwidth by reconfiguring the network at run-time, then we could improve the throughput, reduce the overall latency, provide alternate routes in case of channel failure and ensure that the network delivers the best performance-per-Watt per application.

To address the requirements of energy-efficient and high-throughput NoCs, we leverage the advantages of two emerging technologies, photonic interconnects and 3D stacking with architectural innovations to design high-bandwidth, low-latency, multi-layer, re-

[1] It should be noted that if the electro-optic integration is not monolithic, then the E/O layers are built separately and integrated in 3D via flip-chip bonding. However, here we refer to 2D designs only in the optic layer.

[2] For example, even after more than a decade of research in optics, Jagaur machine from CRAY which employs a dragonfly topology will account for 20-40% photonics and the rest being metal interconnects due to cost constraints.

1072-4451/12 $26.00 © 2012 IEEE
DOI 10.1109/MICRO.2012.34

configurable network, called **R-3PO (Reconfigurable 3D-Photonic On-chip Interconnect)**. R-3PO consists of 16 decomposed photonic interconnect based crossbars placed on four optical communication layers, thereby eliminating waveguide crossing and reducing the optical power losses. The proposed architecture divides a single large monolithic crossbar into several smaller and manageable crossbars which reduces the optical hardware complexity and provides additional disconnected waveguides which provide opportunities for reconfiguration. As the cost of integrating photonics with electronics will be high, statically designed network topologies will find it challenging to meet the dynamically varying communication demands of applications. Therefore, in order to improve network performance, we propose a reconfiguration algorithm whose purpose is to improve performance (throughput, latency) and bypass channel faults by adapting available network bandwidth to application demand by multiplexing signals on crossbar channels that are either idle or healthy. This is accomplished by monitoring the traffic load and applying a reconfiguration algorithm that works in the background without disrupting the on-going communication. Our simulation results on 64-cores and 256-cores using synthetic traffic, SPEC CPU2006, Splash-2 [30] and PARSEC [6] benchmarks provide an energy savings up to 6-36% and outperforms other leading photonic interconnects by more than 10%-25% for adversial traffic via reconfiguration. The significant contributions of this work are as follows:

- We maximize the available bandwidth by reconfiguring the network at run time by monitoring the bandwidth availability and applying the reconfiguration algorithm without disrupting the on-going communication.
- We explore the design space (power-area-performance) of reconfiguring across multiple layers on both synthetic traffic (uniform, permutation) as well as on real application traces (Splash-2, PARSEC, SPEC CPU2006).
- We apply our reconfiguration algorithm to overcome channel faults by effectively sharing the bandwidth of the remaining healthy channels, thereby allowing the performance to degrade gracefully.

2. Related Work

Photonic interconnects is a technology-based solution for designing next generation communication fabric for future multicores. Most photonic interconnects adopt an external laser and on-chip modulators, called micro-ring resonators (MRRs). On application of voltage V_{on}, the refractive index of the MRR is shifted to be in resonance with the incoming wavelength of light which causes a 0 to appear at the end of the waveguide. Similarly, when no voltage is applied, the MRR is not in resonance and a 1 appears at the output. MRRs are used both at the transmitter (modulators) and receiver (filters) sides and have become a favorable choice due to smaller footprint (10 μm), lower power dissipation (0.1 mW), high bandwidth ($>$ 10 Gbps) and low insertion loss (1 dB) [25]. Complementary-metal oxide semiconductor (CMOS) compatible silicon waveguides allow for signal propagation of on-chip light. Waveguides with micron-size cross-sections (5.5 μm) and low-loss (1.3 dB/cm) have been demonstrated [25]. Recent work has shown the possibility of multiplexing 64 wavelengths (wavelength-division multiplexing) within a single waveguide with 60 GHz spacing between wavelengths, although the demonstration was restricted to four wavelengths [3, 25]. An optical receiver performs the optical-to-electrical conversion of data, and consists of a photodetector, a transimpedance amplifier (TIA), and a voltage amplifier [32, 14]. A recent demonstration showed that Si-CMOS-Amplifier has energy dissipation of about 100 fJ/bit with a data rate of 10 Gbps [32]. Thermal stability of MRRs is one of the major challenges causing a mismatch between the incoming wavelength and MRR resonance. Techniques ranging from thermal tuning (more power), athermal tuning (applicable only at fabrication), tuning free-spectral range with backend circuitry (more power) and current injection (smaller tuning range) have been proposed which offer different power consumption levels [7, 9].

On the architecture side, there has been several photonic interconnects that tackle several important issues including arbitration, inter-core communication and core-memory communication[29, 3, 23, 15, 28]. Vantrease et.al. [29] proposed a 3D stacked 256-core photonic interconnect to completely remove all electrical interconnect by designing an optical crossbar and token control. Due to sharing of resources, contention can be high as well as the cost and complexity of designing an optical crossbar for very high core counts. Firefly is an optoelectronic interconnect [23] that reduces the crossbar complexity of [29] by designing smaller optical crossbars connecting select clusters and implementing electrical interconnect within the cluster. In the more recent “macrochip" from Oracle [15], multiple many-core chips are integrated in a single package and propose multi-phase arbitration protocols for communication. FlexiShare [22] is an optical crossbar that combines the advantages of both Corona (single-read, multiple-write) and Firefly (multiple-read, single-write). While Flexishare is concerned with improving bandwidth in the time domain (more slots on more channels), R-3PO improves performance on both space and time domain with a gradient of bandwidth (different percentages). Recently, a 3D photonic interconnect called MPNoCs was proposed that uses multiple layers to create a crossbar with no optical waveguide crossover points [31]. In this work, we extend the 3D photonic interconnect design space by implementing a reconfiguration algorithm that dynamically re-allocates bandwidth from under-utilized to over-utilized links. Prior work on dynamic reconfiguration has been restricted to time slot re-allocation, time and space re-allocation and both power and bandwidth regulation in multiprocessor systems [13]. To the best of our knowledge, this is the first work to propose bandwidth reconfigurability across multiple layers for both improving performance and reliability.

3. R-3PO: Reconfigurable 3D Photonic On-Chip Interconnect

The R-3PO architecture consists of 256 cores, running at 5 GHz, in 64 tile configuration on a 400 mm^2 3D IC. As shown in Figure 1, 256 cores are mapped on a 8×8 network with a concentration factor of four, called a *tile*. From Figure 1(a), the bottom layer, called the *electrical* die, adjacent to the heat sink, contains the cores, caches and memory controllers. To utilize the advantage of a vertical implementation of signal routing, we propose the use of separate optical and core/cache systems unified by a single set of connector vias. The upper die, called the *optical* die, consists of the electro-optic transceivers layer which is driven by the cores via TSVs and four decomposed photonic crossbar layers. The electro-optic layer consists of all the front-end system drivers and the back-end receiver circuitry for photonics. Using TSVs, each tile will modulate the optical signal from an external laser using MRRs and route the signal to the appropriate destination tile. Layers 0-3 contain optical signal routing elements composed of MRRs and bus waveguides and electrical contact for other layers, if necessary. From fabrication perspective, the TSV approach is more tedious due to the maintenance

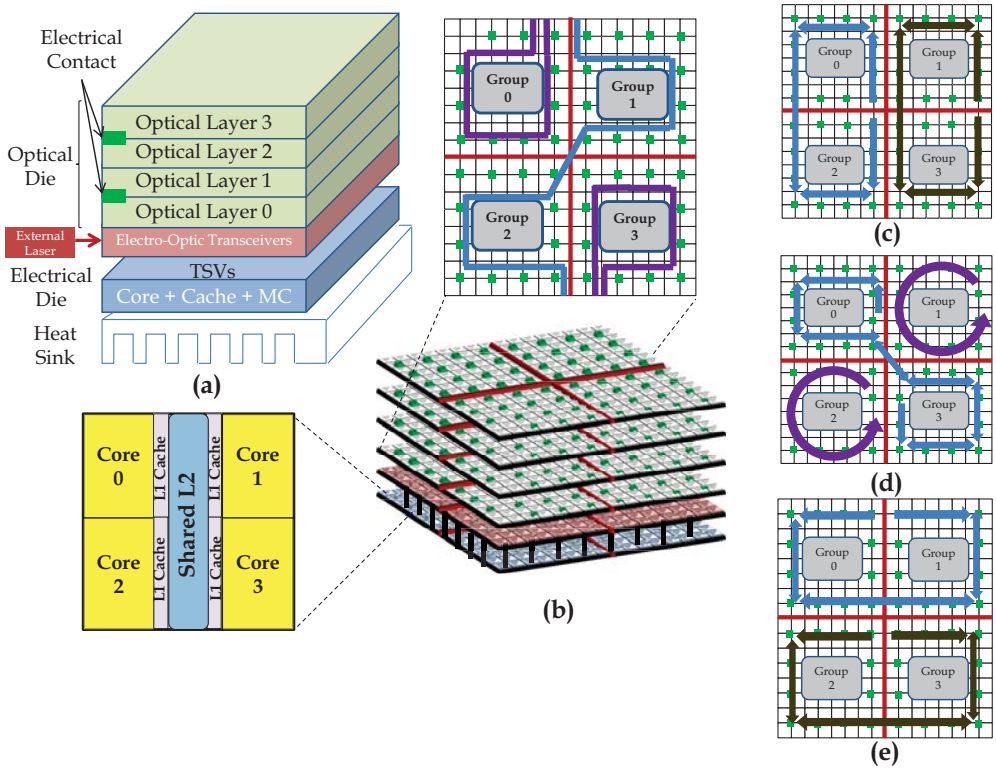


Figure 1: Proposed 256-core 3D chip layout. (a) Electrical die consists of the core, caches, the memory controllers and TSVs to transmit signals between the two dies. The optical die on the lower most layer contain the electro-optic transceivers and four optical layers. (b) 3D chip with four decomposed photonic crossbars with the top inset showing the communication among one group (layer 0) and the bottom inset showing the tile with a shared cache and 4 cores. The decomposition, slicing and mapping of the three additional optical layers: (c) optical layer 1, (d) optical layer 2 and (e) optical layer 3.

of precise alignment for electrical contacts (TSVs). An alternate technique of designing multiple layers is to incorporate racetrack configuration where electrical contact is restricted to the bottom layer (electro-optic layer), and the signal propagation to upper layers is via passive vertical coupling of MRRs. While the racetrack configuration eases fabrication, the technique could increase the laser power due to extra vertical coupling losses and complicate the thermal heating at the upper layers. Adding multiple ring resonators actually improves the filtering of the signal and reduces the crosstalk due to residual signals. Therefore, we propose to design multiple photonic ring resonators coupled in racetrack configuration to traverse multiple layers to prevent the over-reliance on TSVs.

3.1. Intra- and Inter-Group Communication

In the proposed 3D layout, we divide tiles into four groups based on their physical location. Each group contains 16 tiles. Unlike the global 64 $\times$ 64 photonic crossbar design in [29] and the hierarchical architecture in [23], R-3PO consists of 16 decomposed individual photonic crossbars mapped on four optical layers. Each photonic crossbar is a 16 $\times$ 16 crossbar connecting all tiles from one group to another (Inter-group). It is composed of Multiple-Write-Single-Read (MWSR) photonic channels, which requires lesser power than Single-Write-Multiple-Read (SWMR) channels described in [23]. A MWSR photonic channel allows multiple nodes the ability to write on the channel but only one node can read the channel. This channel design reduces power but requires arbitration as multiple nodes can write at the same time. On the other hand, a SWMR channel allows only one node the ability to write to the channel but multiple nodes can read the data. This channel design reduces latency as no arbitration is required but requires source destination handshaking protocol or else, the power to broadcast will be higher. We adopt MWSR and Token slot [29] in this architecture to improve the arbitration efficiency for the channel. Each waveguide used within a photonic crossbar has only one receiver which we define as the *home channel*. During communication, the source tile sends packets to their destination tile by modulating the light on the home channel of the destination tile. An off-chip laser generates the required 64 continuous wavelengths, $\bigwedge = \lambda_0, \lambda_1, \lambda_2 \ldots. \lambda_{63}$. Figure 1(b) shows the detailed floor plan for the first optical layer. For optical layer 0, a 32 waveguide bundle is used for communication between Groups 0 and 3 and two 16 waveguide bundles are used for communication within Groups 1 and 2. For inter-group communication between 0 and 3, the first 16 waveguide bundle is routed past Group 0 tiles so that any tile within Group 0 can transmit data to any destination tile in Group 3. Similarly, the next 16 waveguide bundle is routed past Group 3, so that any tile within Group 3 can communicate with a destination tiles located within Group 0. The bidirectional arrows illustrate that light travels in both directions and depends on which group is the source and destination. The remaining two independent waveguide bundles (16 waveguides) are used for intra-group communication for Groups 1 and 2 respectively. Therefore, we require a total of 64 waveguide bundle per layer. A detailed decomposition and slicing of the crossbar on the other three layers is shown in Figure 1(c-e).

3.2. Router Microarchitecture

Figure 2(a) shows the router microarchitecture in R-3PO for tile 0. Any packet generated from the L2 cache is routed to the input demux with the header directed towards RC (routing computation). The two MSBs are used to direct the packet to a one of the four sets of input buffers (IB_0 - IB_3) corresponding to each optical layer (0-3). For the second set of demuxes, the packet will utilize an unique identifier (that corresponds to the core number) to indicate the source of the packet to prevent any core from overwhelming the input buffers. Token (Request + Release) ensures that packets are transmitted from the IBs without collision and the MRRs are used to modulate the signal into the corresponding home channel. At the receiver, the reverse process takes place where the packet from the optical layer is converted into electronics and according to the unique identifier will find one set of buffers available. Token Control is used to prevent buffer overflow at the home node by checking the number of empty buffer slots. If the number of empty buffer slots falls below a certain threshold Buf_Th, then the destination tile will capture the circulating token and will not re-inject the token until the number of free slots increases to the threshold. Furthermore, the receiver of R-3PO does not require router computation for an incoming flit of a packet because, flit interleaving does not take place as an optical token is not re-injected until the whole packet is sent. The packets will then contend to obtain the switch (switch allocator (SA)) to reach the L2 cache. It should be noted that the proposed unique identifier is similar to virtual channel allocator, however we do not perform any allocation as the decision to enter any buffer is determined on the core number (source or destination). Figure 2(b) shows the proposed token control block. In the token control block, an optical token is only placed on the token inject waveguide when an optical token is present (high TR signal) and the buffer congestion (BC) signal is low. A low BC signal in this case represents a free buffer slot at the destination tile and a high BC signal represent that all the buffer slots are full at the destination tile. 2(c) shows the router pipeline. RC ensures that the packet is directed to the correct output port for both static and reconfigured communication. BWS writes the packet into the buffer slot. EO conversion takes place with appropriate buffer chain after the token is received. Optical transmission can take anywhere between 1-3 clock cycles running at 5 Ghz. OE conversion is repeated at the receiver, BWD writes the packet into the buffer slot and finally switch allocation (SA) ensures that the packet progresses into the L2 cache.

4. Reconfiguration

As future multicores will run diverse scientific and commercial applications, networks that can adapt to communication traffic at runtime will maximize the available resources while simultaneously improving the performance. Moreover, faults within the network or the channel can isolate healthy groups of tiles; with the natural redundancy available in the decomposed crossbar, we can take advantage of reconfiguration to overcome channel faults and maintain limited connectivity. To implement reconfiguration, we propose to include additional MRRs that can switch the wavelengths from different layers to create a reconfigurable network. Further, we also propose a reconfiguration algorithm to monitor traffic load and dynamically adjust the bandwidth by re-allocating excess bandwidth from under-utilized links to over-utilized links.

4.1. Bandwidth Re-Allocation

To illustrate with an example, consider a situation where tiles in Group 0 communicates only with tiles in Group 3. Figure 3 shows

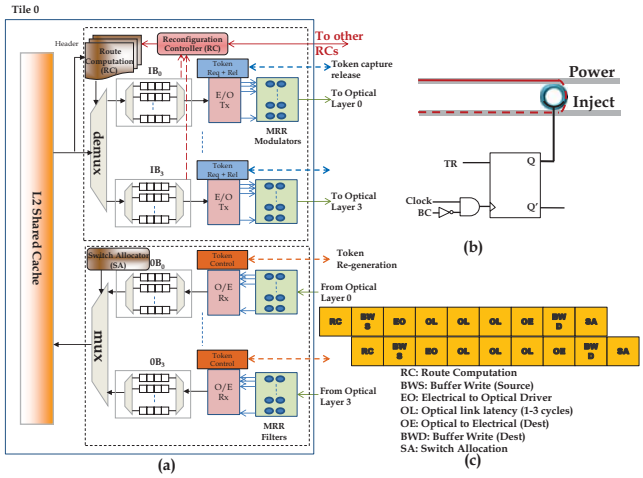


Figure 2: (a) Router microarchitecture, (b) token control and (c) router pipeline.

the reconfiguration mechanism. The *static* allocation of channel for communication are in layer 2 as shown in Figure 3(a). Suppose no tile within Group 1 (in layer 1) communicates with Group 3, then we can re-allocate the bandwidth from Group 1 to Group 0 to communicate with Group 3. To implement reconfiguration, however, we need to satisfy two important requirements: (1) There should be a source waveguide which should be freely available to start the communication on a source layer, and (2) there should be a destination waveguide which also should be freely available to receive the extra packets. As shown in Figure 3(b), as the two Groups 0 and 3 talk to each other, we have the first set of waveguides on layer 0 (generally used to communicate within the group) available, therefore this satisfies the first condition. As Group 1 does not communicate with Group 3, we can utilize the destination waveguide available in layer 1 and this satisfies the second condition. The signal originates on layer 0, switches to layer 1 to reach the destination. Note that this additional channel is available in addition to layer 2 static configuration, thereby doubling the bandwidth. Therefore, during reconfiguration Group 0 has doubled the bandwidth to communicate with Group 3 via layers 2 (static) and 1 (dynamic). Two different communication are disrupted when the reconfiguration occurs, namely, Group 0 in layer 0 can no longer communicate with itself and Group 1 in layer 1 can no longer communicate with Group 3.

4.2. Design-Space Exploration

The objective of reconfiguration is to improve performance by re-allocating bandwidth from under-utilized to over-utilized links. The design space of reconfiguration is large as there can be several combinations across multiple layers. Figure 4 shows four possible combinations that we will evaluate as they cover most of the design space. Row-column matrix indicates the statically allocated communication. For example layer 0 - layer 0 shows three combinations G0 $<->$ G0, G1 $<->$ G2 and G3 $<->$ G3 i.e. group 0 communicates with itself, groups 1 and 2 communicate with each other and group 3 communicates with itself. The square (red) boxes show which layers can be used for reconfiguration and the arrow indicates the layers that can be used for reconfiguration. Figure 4(a) shows layer 0 can reconfigure and take away bandwidth from layer 1; similarly layer 1 can reconfigure and can take away bandwidth from layer 0. Layer

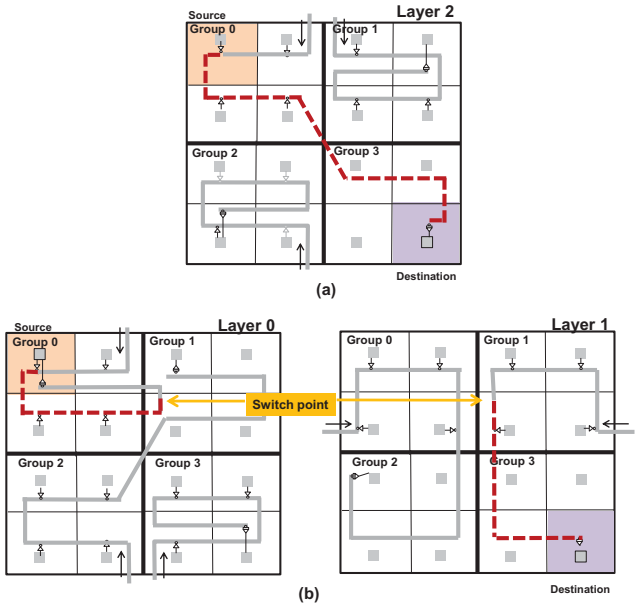


Figure 3: (a) Static communication between the source in Group 0 and destination in Group 3. (b) Illustration of reconfiguration between Groups 0 and 3 using partial waveguides from layers 0 and 1.

2 can take bandwidth from layer 3 and vice versa. This approach restricts to one additional layer that can be used for reconfiguration and we call this R-3PO-L1 (R-3PO-Limited to 1 Layer) and this restricted mechanism will reduce both the power consumption and area overhead. Figure 4(b) shows reconfiguration across one or two layers; however both layers have to be adjacent. Layer 0 can only reconfigure with layer 1, where as layer 1 can reconfigure with both layer 0 and layer 2 (adjacent). Adjacent layer reconfiguration is easier to implement as the next layer (above or below) will be used which improves on a single layer and we call this R-3PO-LA (R-3PO-Limited to adjacent layer). Figure 4(c) shows reconfiguration across two layers even if they are not adjacent and we call this configuration R-3PO-L2 (R-3PO-Limited to 2 Layers). This increases the power consumption as well as design fabrication as more TSVs will be needed. One side-effect of this reconfiguration is that as more layers are involved, there are more channels lost due to reconfiguration. This is primarily due to the fact that as additional waveguides are consumed, we are then restricting the number of layers that can be reconfigured. For adverse and embarrassingly parallel applications, this would be an interesting option as more layers can be used for reconfiguration. Figure 4(d) shows the complete reconfiguration, as any layer can go to any other layer, and we call this configuration R-3PO-L3 (R-3PO-All 3 Layers). This fully reconfigured design will need the most in terms of area overhead and also incur higher complexity in terms of fabrication as TSVs have to extend to all the layers.

4.3. Fault Tolerance

Fault tolerance occurs by allowing data from the faulty channel to be switched to an adjacent layer (channel) that communicates with the same destination. Figure 5 shows an example of how fault tolerance is implemented in R-3PO. In this example, the tiles in

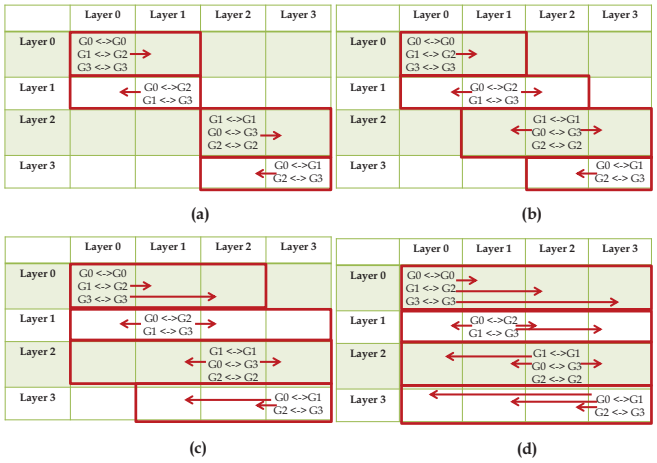


Figure 4: Various configurations evaluated: (a) R-3PO-L1 (R-3PO-Limited to 1 Layer), (b) R-3PO-LA (R-3PO-Limited to adjacent layer), (c) R-3PO-L2 (R-3PO-Limited to 2 Layers) and (d) R-3PO-L3 (R-3PO-All 3 Layers)

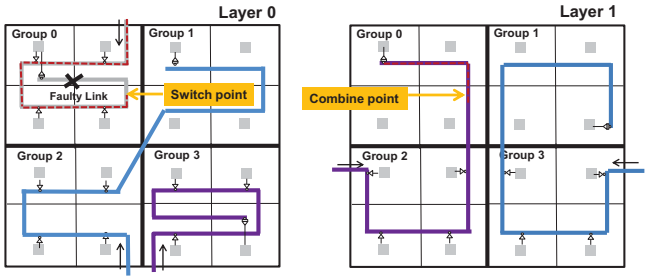


Figure 5: Fault tolerance in R-3PO.

Group 0 cannot communicate with Tile 0 because the optical receiver at Tile 0 is inoperable or faulty, thereby isolating Tile 0 from other tiles in Group 0. To detect a fault, we augment the reconfiguration algorithm and hardware counters to detect faulty links.[3] Once the fault is detected, data is re-routed to the adjacent layer waveguides that communicate with the same destination tile. After Group 0 detects that the communication to Tile 0 is faulty, any data originating from Group 0 will be switched to the waveguide in Layer 1 that communicates with Tile 0. In addition, we prevent a tile from Group 0 and a Tile from Group 2 from communicating to Tile 0 at the same time which requires the token sharing scheme to be updated. In this example, after the fault is detected, any tiles in Group 0 will need to capture the token that tiles in Group 2 use to communicate with Tile 0. This in essence increases the number of tiles that share a common link; therefore the bandwidth Group 2 utilizes to communicate with Tile 0 is now shared by all tiles in Group 0 to communicate with Tile 0. Reconfiguration allows bandwidth or channel sharing where the faulty channel can be bypassed by using bandwidth on adjacent layer.

4.4. Algorithm Implementation

We design our reconfiguration algorithm with the following objectives: (a) The algorithm should not be overly sensitive to traffic

[3]The design space of testing the functionality of the channel is vast as multiple sources or destination can be faulty; in this paper, we limit fault to the destination receiver which can be self tested by the home channel by transmitting a pinging packet.

fluctuations to prevent rapid changes in topology; (b) the algorithm should mostly work in the background, and (c) the algorithm should ensure that no tile is starved from bandwidth. To implement such a reconfiguration, we first take measurements that are available such as link utilization (Link_{util}) and buffer utilization (Buffer_{util}) using hardware counters [8]. This implies that each tile within a group will have four hardware counters (one for each of the three groups) that will monitor traffic utilization and provide the link and buffer information to Reconfiguration Controller (shown in 2). All these statistics are measured over a sampling time window called *Reconfiguration window* or phase, $R_W{}^t$, where t represents the reconfiguration time t. This sampling window impacts performance, as reconfiguring finely incurs latency penalty and reconfiguring coarsely may not adapt in time for traffic fluctuations. In our performance section, we show that we evaluated a number of PARSEC applications to determine the optimum size for R_W. For calculation of Link_{util} at configuration window t, we use the following equation:

$$Link_{util}^{t} = \frac{\sum_{cycle=1}^{R_W} Activity(cycle)}{R_W} \quad (1)$$

where $Activity(cycle)$ is 1 if a flit is transmitted on the link or 0 if no flit is transmitted on the link for a given cycle. For calculation of Buffer_{util} at configuration window t, we use the following equation:

$$Buffer_{util}^{t} = \frac{\sum_{cycle=1}^{R_W} Occupy(cycle)/Total_{buffers}}{R_W} \quad (2)$$

where $Occupy(cycle)$ is the number of buffers occupied at each cycle and $\text{Total}_{b}uffers$ is the total number of buffers available for the given link. When traffic fluctuates dynamically due to short term bursty behavior, the buffers could fill up instantly. This can adversely impact the reconfiguration algorithm as it tries to re-allocate the bandwidth faster leading to fluctuating bandwidth allocation. To prevent temporal and spatial traffic fluctuations affecting performance, we take a weighted average of current network statistics (Link_{util} and Buffer_{util}), so that the network will gradually re-allocate bandwidth. We calculate the Buffer_{util} as follows:

$$Buffer_{w}^{t} = \frac{\sum_{Buffer_{util}^{t} \times weight + Buffer_{util}^{t-1}}}{weight + 1} \quad (3)$$

where *weight* is a weighting factor and we set this to three in our simulations [26].

After each $R_W{}^t$, each tile will gather its link statistics (Link_{util} and Buffer_{util}) from the previous window $R_W{}^{t-1}$ and send to its local reconfiguration controller (RC) for analysis. We assume that Tile 0 of every group gathers the statistics from the remaining tiles and this can be few bytes of information that is periodically transmitted. Next, when each RC_i, ($\forall$ i = 0, 1, 2, 3), has finished gathering link and buffer statistics from all its hardware controllers, each RC_i will evaluate the available bandwidth for each link depending on the $\text{Link}_{util}{}^{t-1}$ and $\text{Buffer}_{util}{}^{t-1}$ and will classify its available bandwidth into a different thresholds β_{1-4} corresponding to 0%, 25%, 50% and 90%. We never allocate 100% of the bandwidth as the source group may have new packets to transmit to the destination tile before the next R_W. RC_i will send link information (availability) to its neighbor RC_j (j $\neq$ i). If RC_j needs the available bandwidth, RC_j will notify the source and the destination RCs so that they can switch the MRRs and inform the tiles locally of the availability. Once the source/desitnation RCs have switched their reconfiguration MRRs, RC_i will notify RC_j that the bandwidth is available for use. On the other hand, if a node within RC_i that throttled its bandwidth requires it back due to increase in network demand, RC_i will notify that it requires the bandwidth back and afterwards will deactivate the corresponding MRRs. The above reconfiguration completes a three-way handshake where RC_i first notifies RC_j, then RC_j notifies RC_i that RC_j will use the additional bandwidth, and finally RC_i notifies RC_j that the bandwidth can be used. Table 1 shows the reconfiguration algorithm in R-3PO.

Table 1: Reconfiguration Algorithm used in R-3PO.

Step 1:	Wait for Reconfiguration window, $R_W{}^t$
Step 2:	RC_i sends a request packet to all local tiles requesting $Link_{Util}$ and $Buffer_{Util}$ for previous $R_W{}^{t-1}$
Step 3:	Each hardware counter sends $Link_{Util}$ and $Buffer_{Util}$ statistics from the previous $R_W{}^{t-1}$ to RC_i
Step 4(a):	RC_i classifies the link statistic for each hardware counter as: If Link_{util} = 0.0 Not-Utilized: Use β_4 If $\text{Link}_{util} \leq \text{L}_{min}$ Under-Utilized: Use β_3 If $\text{Link}_{util} \geq \text{L}_{min}$ and $\text{Buffer}_{util} < \text{B}_{con}$ Normal-Utilized: Use β_2 If $\text{Buffer}_{util} > \text{B}_{con}$ Over-Utilized: Use β_1
Step 4(b):	Faulty links detected by RC_i are eliminated from reconfiguration; Token sharing updated to bypass the faulty link
Step 5:	Each RC_i sends bandwidth available information to RC_j, (i$\neq$j)
Step 6:	If RC_j can use any of the free links then notify RC_i of their use, else RC_j will forward to next RC_j
Step 7a:	RC_i receives response back from RC_j and activates corresponding microrings
Step 7b:	RC_j notifies the tiles of additional bandwidth and RC_i notifies RC_j that the additional bandwidth is now available
Step 8:	Goto Step 1

5. Performance Evaluation

In this section, we evaluate the performance, power-efficiency and impact of faulty channel in R-3PO when compared to competing electrical interconnects and photonic interconnects.

5.1. Simulation Setup

Our cycle-accurate simulator models in detail the router pipeline, arbitration, switching and flow control. An aggressive single cycle electrical router is applied in each tile and the flit transversal time is one cycle from the local core to electrical router [18]. As the delay of Optical/Electrical (O/E) and Electrical/Optical (E/O) conversion can be reduced to less than 100 ps [29], the total optical transmission latency is determined by physical location of source/destination pair (1 - 3 cycles) and two additional clock cycles for the conversion delay. We assume an input buffer of 16 flits with each flit consisting of 128

bits. The packet size is 4 flits which is sufficient to fit a complete cache line of 64 bytes. We assume a supply voltage V_{dd} of 1.0 V and a router clock frequency of 5 Ghz [29, 23]. We compare R-3PO architecture to three other crossbar-like photonic interconnects, Corona [29], Firefly [23], MPNoCs [31]; and two electrical interconnects (mesh and Flattened Butterfly) [12]. We implement all architectures such that four cores (one tile) are connected to a single router. We assume token slot for both R-3PO and Corona to pipeline the arbitration process to increase the efficiency. We use Fly_Src routing algorithm [23] for Firefly architecture, where intra-group communication via electrical mesh is implemented first and then inter-group via photonic interconnects. For a fair comparison, we ensure that each communication channel in either electrical or optical network is 640 Gbps with 64 wavelengths. We also evaluate the performance by restricting the channel bandwidth to 16/8 wavelengths and communication bandwidth limited to 160/80 Gbps. For each network, we ensure that identical bandwidth is maintained for each link in our network, thereby providing equal bandwidth between each source and destination pairs, whether it be electrical or optical networks.

For open-loop measurement, the packet injection rate is varied from 0.1 to 0.9 of the network capacity, and packets are injected according to the Bernoulli process based on the given network load. The simulator was warmed up under load without taking measurements until steady state was reached (up to 1000 cycles). Then a sample of injected packets were labeled during a measurement interval (1000 to 10,000). The simulation was allowed to run until all the labeled packets reached their destinations. We consider both uniform as well as permutation traffic such as bit-complement (bitcomp), bit-reversal (bitrev), transpose, butterfly, neighbor and prefect shuffle traffic patterns for 256-cores.

For closed-loop measurement, we collect traces from real applications using the full execution-driven simulator SIMICS from WindRiver with the memory package GEMS enabled [20]. We evaluate the performance of 64-core versions of each network on Splash-2 [30], PARSEC [6] and SPEC CPU2006 workloads. We assume a 2 cycle latency to access the L1 cache (64 KB, 4-way), a 4 cycle latency to access the L2 cache (4MB, 16-way), cache line size of 64 bytes and a 160 cycle latency to access the main memory. For Splash-2 traffic, we assume the following kernels and workloads: FFT (16K particles), LU (512 $\times$ 512 with a block size of 16 $\times$ 16), Radix (1 Million integers), Ocean (258 $\times$ 258), and Water (512 Molecules). We consider six PARSEC applications with medium inputs (blackscholes, facesim, fluidanimate, freqmin, and streamcluster) and two workloads from SPEC CPU2006 (bzip and hmmer). We ran several benchmarks of PARSEC and Splash-2 to determine the optimum size of R_W by varying the simulation cycles. While initially the performance improved with increasing window size as more statistics are available which enable better decision making; at very large window sizes, the performance diminishes as the algorithm cannot react fast enough to take advantage of the reconfiguration algorithm. Our simulation results show that 1300 cycles for R_W showed the best performance. We assume a 100 cycle latency for the reconfiguration to take place after each R_W (three-way handshake delay). It should be noted that the reconfiguration latency is only incurred by those links that already are lightly loaded and, therefore do not experience a significant delay.

5.2. Simulation Results

5.2.1. Splash-2, PARSEC and SPEC CPU2006 for 64 Cores: We analyze the speed-up for few selected Splash-2, PARSEC and SPEC CPU2006 applications [30] for 64/16/8 wavelengths, where the speed-up is normalized to mesh architecture. From Figure 6, all R-3PO configurations show a speedup of 2.5 - 3X over electrical mesh, 10-40% improvement over Flattened-Butterfly and Firefly architectures, 22-18% over MPNoC and Corona architectures for 64 wavelengths. The performance gains over electrical and electro-optic networks are derived primarily due to the decomposed crossbars which enable increased traffic outflows from the router into four difference optical crossbars. Further performance improvement over photonic crossbars such as Corona and MPNoC are due to the reconfiguration algorithm which takes advantage of the idle communication channels. Within the four different configurations of R-3POs, the best performing configuration is R-3PO-L3 which provides the maximum flexibility by reconfiguring all the optical layers. For 64 wavelengths, the performance improvements provided by R-3PO-L3 and R-3PO-LA is 6-8% for streamcluster and bzip over R-3PO-L1.

Figure 7 shows the performance of various networks on Splash-2, PARSEC and SPEC CPU2006 benchmarks for 16 wavelengths. From Figure 7, all R-3PO configurations show a speedup of 2.3 - 3.5X over electrical mesh, 17-40% improvement over Flattened-Butterfly and Firefly architectures, 32-18% over MPNoC and Corona architectures for 16 wavelengths. When the number of wavelengths is reduced, the performance improvements over 64 wavelengths are primarily due to the reconfiguration algorithm as the additional bandwidth has more of an impact on the speedup. Figure 8 shows the performance of various networks for 8 wavelengths. From Figure 8), all R-3PO configurations show a speedup of 2.1 - 3.6X over electrical mesh, 17-62% improvement over Flattened-Butterfly and Firefly architectures, 42-18% over MPNoC and Corona architectures for 8 wavelengths. When the resources are further constrained, the bandwidth is stressed where the re-allocated bandwidth via reconfiguration can alleviate performance. Clearly, the performance gains increases dramatically when we reduce the bandwidth and the reconfiguration algorithm can assist in improving the performance. For LU, water, streamcluster and facesim benchmarks, R-3PO-L2 and R-3PO-L3 show over a 10% increase in performance when compared to R-3PO-L1. From the figure, the average speed provided by R3P0-LA, R-3PO-L2, and R-3PO-L3 over R-3PO-L1 ranges from about 1% to as high as 10%. Multiple configurations of R-3PO provide different performance gains and the speedup increases with reduced bandwidth via reconfiguration.

5.2.2. Synthetic Traffic: 256 Cores The throughput for all synthetic traffic traces for 256-core implementations are shown in Figure 9 and is normalized to mesh network (for Uniform, the mesh has a throughput of 624 GBytes per sec). R-3PO-L1 has about a 2.5 $\times$ increase in throughput over Corona for uniform traffic due to the decomposition of the photonic crossbar. The decomposed crossbars allow for a reduction in contention for optical tokens as now a single token is shared between 16 tiles instead of 64 tiles as in Corona. Firefly slightly outperforms R-3PO-L1 for uniform traffic due to the contention found in the decomposed photonic crossbars. Moveover, Firefly uses a SWMR approach for communication which does not require optical arbitration. From the figure, R-3PO-L1 slightly outperforms Corona for bit-reversal and complement traffic traces. This is due to lower contention for optical tokens in the decomposed crossbars. R-3PO-L1 significantly outperforms mesh for the bit-reversal, matrix-transpose and complement traffic patterns. In these traffic patterns, packets need to traversal across multiple mesh routers which in turn increases the packet latency and thereby reduces the through-

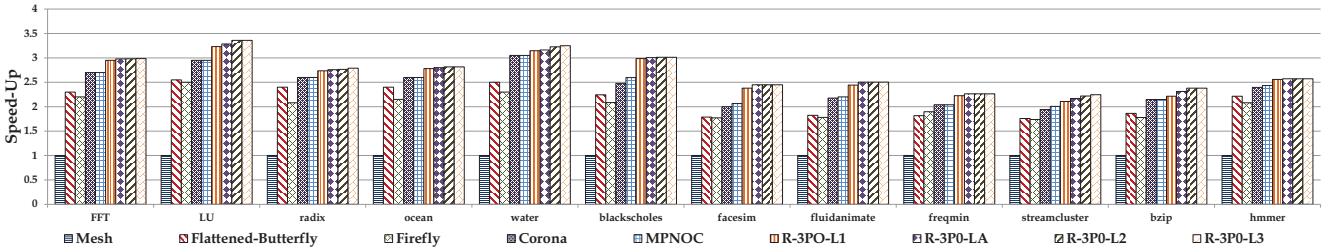


Figure 6: Speed-up for 64-core using SPLASH-2, PARSEC and SPEC CPU2006 traffic traces using 64 wavelengths.

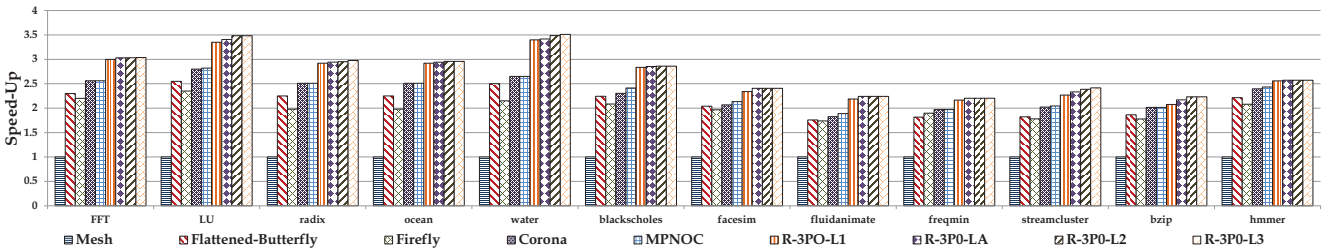


Figure 7: Speed-up for 64-core using SPLASH-2, PARSEC and SPEC CPU2006 traffic traces using 16 wavelengths.

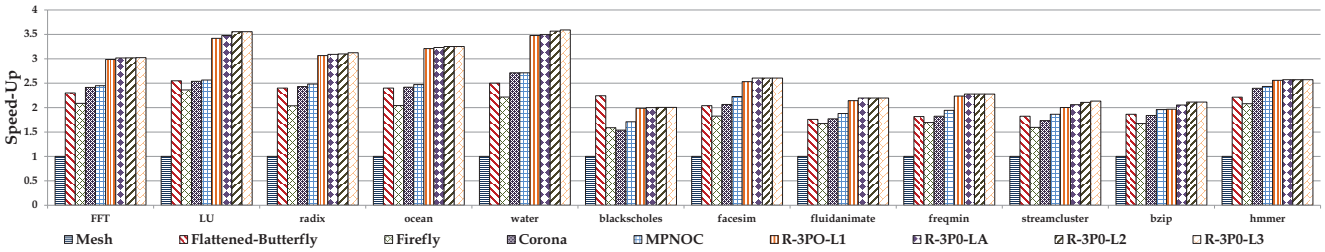


Figure 8: Speed-up for 64-core using SPLASH-2, PARSEC and SPEC CPU2006 traffic traces using 8 wavelengths.

put. When R-3PO-L1 is compared to Firefly, R-3PO-L1 outperforms Firefly by 2.5 $\times$. In Firefly, most traffic patterns will require packets to travel on several electrical routers and then an optical link to reach the destination. R-3PO-L3 is able to out perform R-3PO-L1 for complement, matrix-transpose and perfect shuffle traffic traces. These permutation traffic traces exhibit adversial patterns which will benefit R-3PO-L1. In complement traffic, R-3PO-L1 has about a 55% increase in performance when compared to R-3PO-L1.

5.2.3. Fault Tolerance We evaluated the performance degradation when 10%, 25% and 50% of the channels are faulty. These faults were randomly inserted such that they do not coincide with the reconfiguration window cycle. We assume that every tile checks the home channel working once at the beginning of the R_w and if there are any faults, bandwidth sharing is enabled where bandwidth from other healthy channels is re-allocated. Figure 10 shows the performance degradation for R-3PO-L1 for 64 wavelengths. The results show that with 10%, 25% and 50% link failures, performance degrades by 5%, 10-15% and 20-40% respectively. While the reconfiguration algorithm kicks in within a couple of iterations (in worst case scenario), the loss primarily arises from the sharing of channel which increases the latency for both faulty as well as non-faulty communication. The results show that reconfiguration algorithm can bypass the faults by efficiently sharing the link bandwidth with some performance degradation. While the fault model assumes a high fault rate (10% - 50%), with adequate process development and monolithic integration, variation-induced fault rates are actually much lower [2]. Our analysis assumes worst-case fault rate for the system evaluation with reconfiguration.

5.3. Energy Comparison

The energy consumption of a photonic interconnect can be divided into two parts, electrical energy and optical energy. Optical energy consists of the off-chip laser energy and on-chip MRRs heating energy. In what follows, we first discuss the electrical energy and then optical energy consumption.

5.3.1. Electrical Energy Model The electrical energy dissipated includes the energy of the link, router and back-end circuitry for optical transmitter and receiver. We use ORION 2.0 [11] to obtain the energy dissipation values for an electrical link and router and modified their parameters for 22nm technology according to ITRS. We assume all electrical links are optimized for delay and the injection rate to be 0.1. Moreover, we include the energy dissipated in both planar and vertical links (communicating with all layers). Furthermore, we incorporate the power dissipated within the router buffers, except for virtual channel allocation. The energy for planar link is conservatively obtained as 0.15 pJ/bit for Firefly, 0.075 pJ/bit for mesh, and

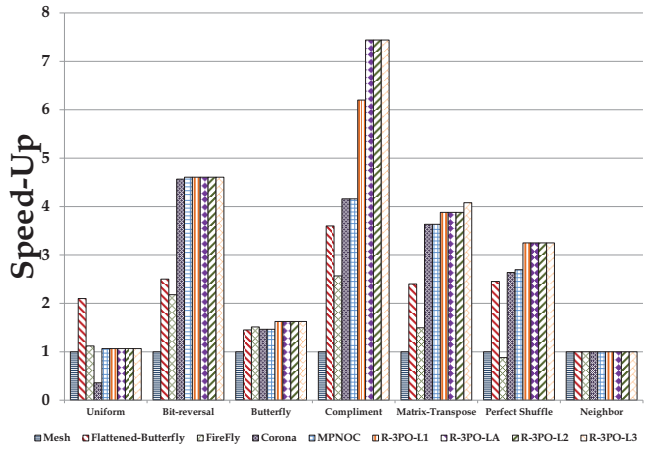


Figure 9: Simulation results showing normalized saturation throughput for seven traffic patterns for 256 cores.

0.15 pJ/bit per router bypass for Flattened-Butterfly under low swing voltage signalling [11]. The link energy dissipation depends on the location of the source and destination for Flattened-Butterfly. For a 10-layer chip, the vertical via is determined as ∼100- 200μm [15], which is significantly less than planar links. As a result, the energy consumption in vertical links are very small. We neglect it when we calculate our electrical link power model. We calculate the energy dissipated for a 10 × 10 router to be 0.42 pJ/bit, 8 × 8 router to be 0.30 pJ/bit [11], and 5 × 5 router will be 0.22 pJ/bit [11]. This is the energy dissipated per hop of communication. For each bit of optical transmission, we need to provide electrical back end circuit for transmitter and receiver. We assume the O/E and E/O converter energy is 100fJ/b, as predicted in [17]. For RC power dissipation, each RC optimizes performance by analyzing few bits (2-16) of information every 1300 cycles. While the data packets are as large as 512 bits of information, the static and dynamic power impact of the reconfiguration controller is negligible in comparison to the actual data movement.

5.3.2. Optical Energy and Loss Model The optical power budget is the sum of the laser power and the power dissipated in the MRRs. The laser power is determined by $P_{laser} = P_{rx} + C_{loss} + M_s$ where P_{laser} is the required laser power, P_{rx} is the receiver sensitivity, C_{loss} is the channel loss and M_s is the system margin. In order to perform an accurate comparison with the other two optical architectures, we use the same optical device parameters and loss values provided in [3, 1], as listed in Table 2. In this paper, we assume a flat thermal model that requires ring resonator heating power. However, this power can be lower as heating power can be shared by an array of rings [21], however this depends strongly on the actual layout of the ring resonators. Recent work has also demonstrated that flat thermal profile may not be practical and could increase off-resonance coupling losses [19, 15]. In this work, we show a preliminary analysis of the ring heating power which is a conservative model and more aggressive models can reduce this power [21, 19]. In addition, we assume a BER of 10^{-12} for each optical link and the Signal-Noise-Ratio (SNR) is given by [27]

$$BER = \frac{1}{2} - \frac{1}{2} erf(0.354\sqrt{SNR}) \quad (4)$$

Table 2: Electrical and optical power losses for select optical components.

Component	Value	Unit
Laser efficiency	5	dB
Coupler (Fiber to Waveguide)	1	dB
Waveguide	1	dB/cm
Splitter	0.2	dB
Non-Linearity	1	dB
Ring Insertion & scattering	1e-2 - 1e-4	dB
Ring Drop	1.0	dB
Waveguide Crossings	0.5	dB
Photo Detector	0.1	dB
Ring Heating	26	μW/ring
Ring Modulating	500	μW/ring
Receiver Sensitivity	-26	dBm

Table 3: Electrical power dissipation for various photonic interconnects.

	Corona	Firefly	R-3PO	Mesh
Link(electric)	-	0.15pJ/b	-	75fJ/b
Router	0.22pJ/b	0.30pJ/b	0.22pJ/b	0.22pJ/b
O/E, E/O	100fJ/b	100fJ/b	100fJ/b	-
Optical loss	-25.2dB	-17.6dB	-16dB	-
Power(λ)	0.81mW	0.14mW	0.10mW	-
Laser power	13.6W	2.4W	6.1W	-
Ring heating	26W	6.5W	27.5W	-

and the minimum power for a given SNR is

$$SNR = \frac{Po \cdot \eta}{NEP \cdot \sqrt{f}}, \quad (5)$$

where η is the quantum efficiency of the detector, NEP is the Noise-Equivalent-Power, and f is the transmission frequency [27]. Using the above equations, we determined the SNR in R-3PO to be 176.42.

Based on the energy model discussed in the previous section, we calculate the energy parameters of all four architectures as shown in Table 3. We test uniform traffic with 0.1 injection rate on the all architectures and obtain the energy per-bit as shown in Figure 11. Although Firefly has $\frac{1}{4}$ as many MRRs as Corona and R-3PO, which results in $\frac{1}{4}$ energy consumption per bit on ring heatings, it still consumes more energy than R-3PO and CORONA due to the overhead of electrical routers and links. In general, R-3PO saves 6.5%, 23.1%, 36.1% energy per bit compared to Corona, Firefly, and mesh respectively. R-3PO has a slight increase in power dissipation over MPNoCs due to the additional MRRs required for reconfiguration. The total network power for each application varied between 4 Watts to 6 Watts for 64-core simulation and 16 Watts to 24 Watts for 256-core simulation.

5.3.3. Laser Power Variations The optical losses shown in Table 3 are mostly conservative estimates that may not reflect the actual losses in future photonic devices. Figures 12(a) and 12(b) illustrate the impact on laser power when four optical parameters, namely receiver sensitivity, ring filter loss, wavelengths and waveguides are changed. We choose these four parameters from Table 3 as we believe they will have the greatest impact on the total laser power. We evaluate the variation in laser power with receiver sensitivity and the number of

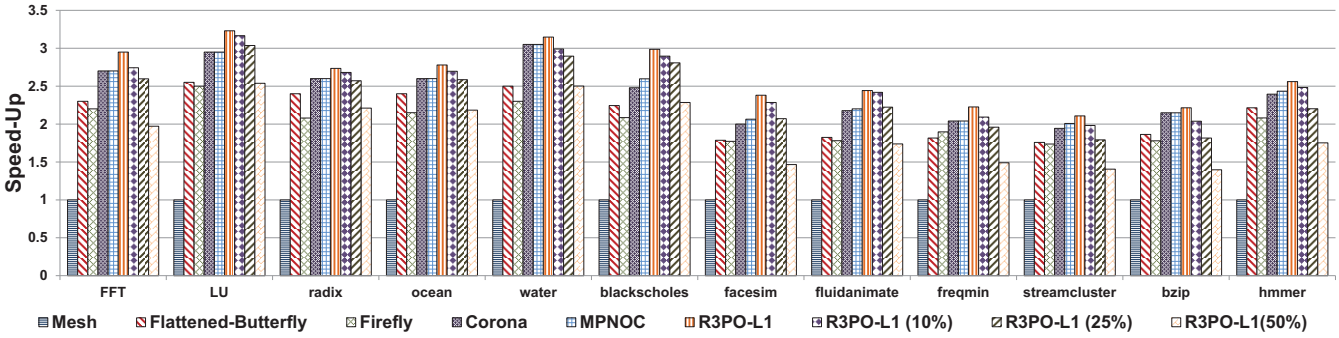


Figure 10: Speed-up for 64-core using SPLASH-2, PARSEC and SPEC CPU2006 traffic traces using 64 wavelengths for R-3PO-L1 with 10%, 25% and 50% faults in the channel.

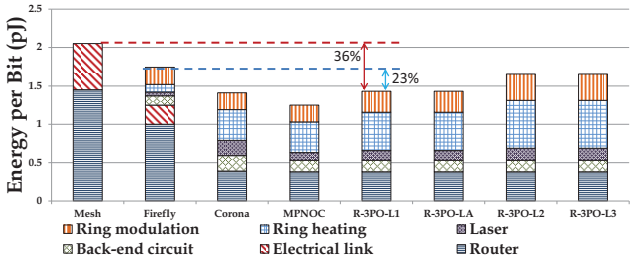


Figure 11: Average energy per-bit for electrical and photonic interconnects.

wavelengths in Figure 12(a). It should be noted that the bandwidth for each wavelength configuration is maintained at 640 Gbps in order to evaluate the laser power variation. Figure 12(a) shows that the laser power increases as the receiver sensitivity decreases because power per bit increases at the receiver. For example, a 6 dBm decrease in receiver sensitivity (-20 dBm) would result in a 4× increase in total laser power. Clearly, the receiver sensitivity has the greatest impact on the total network power for R-3PO, a low sensitivity receiver can increase the external laser power. Figure 12(b) shows the variations in laser power with the ring filter loss and the number of waveguides. From the figure, the increase in total laser power from waveguide losses has greater impact than the ring filter losses. This is due to the optical signal traversing several centimeters before arriving at the photodetector. For example, a 0.5 dB increase in waveguide loss (1.8 dB/cm) would more than double the total laser power.

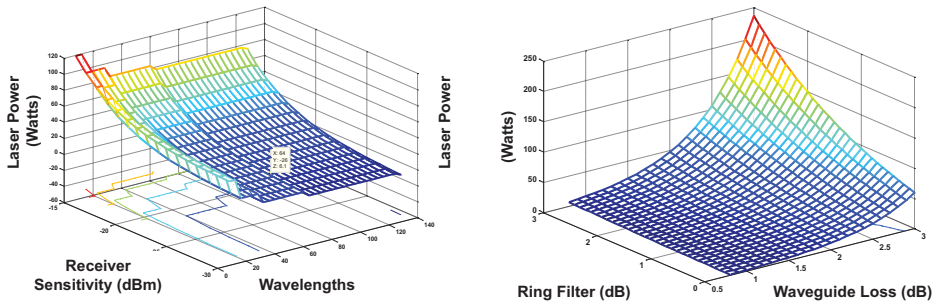
5.3.4. R-3PO Energy-Delay Product: In this paper, we propose different configurations of R-3PO that have different degrees of reconfiguration (increases bandwidth) and dissipate different energy. As such, the increase in performance due to more reconfiguration options may come at the price of higher energy dissipation. Figure 13 evaluates the energy-delay product (EDP) for all R-3PO configurations using the Splash-2, PARSEC, and SPEC CPU 2006 benchmarks. From the figure, it can be seen that R-3PO-L1 and R-3PO-LA have the least EDP. This is due to the fact that the slight increase in energy per bit over MPNoC and R-3PO-L1 is offset by the increase in performance over other networks. On the other hand, R-3PO-L2 and R-3PO-L3 have the highest EDP among all optical networks. This is obvious as these two networks have the highest energy per bit and there is only a slight increase in performance when compared to R-3PO-L1, R-3PO-LA and Corona. Mesh has the highest EDP as it is the worst network in terms of performance and has the highest energy per bit. When Flattened-Butterfly and Firefly are compared, their EDP have similar values. Firefly consumes lesser energy than Flattened-Butterfly although its performance is also proportionally lower than Flattened-Butterfly.

5.4. Area Analysis

In this subsection, we analytically compare the optical and electrical area overhead of R-3PO to Firefly [23] and Corona [29] photonic interconnects. For the optical area overhead, we consider the area required for all waveguides, MRRs and photodetectors. For the electrical layer, we consider the area required for all routers, electrical links and electrical receiver circuitry. Table 4 shows the area overhead of both optical and electrical components used in the area calculation. From Table 4, each router and electrical link values were obtained from Orion 2.0 by scaling 32 nm technology values to 22 nm technology. From our evaluation, we observe that both Corona and Firefly require 10% more optical area than R-3PO. This may be counter-intuitive, but R-3PO uses decomposed crossbars that permit waveguides in R-3PO to be shorter than the long serpentine waveguides used in both Corona and Firefly. In terms of electrical layer area overhead, R-3PO consumes 4X more electrical area than Corona. As each tile is connected to four optical layers to facilitate inter-group communication, each tile in turn should have the ability to receive four signals instead of one as in Corona. However, when R-3PO is compared to Firefly in terms of electrical area overhead, Firefly consumes about 75% more area. In Firefly, the electrical network can simultaneously receive from seven sets of optical receivers at once due to SWMR organization. R-3PO combines both MWSR (Corona) and SWMR (Firefly) communication channels, thereby increasing the communication channels to each tile while reducing the optical area overhead. For the different configurations of R-3PO, the additional increase in area overhead when compared to MPNOC is marginal as a single MRR can be used to switch all wavelengths from one layer to the other [5]. For example, the increase in area overhead for R-3PO-L1 is less than 1% and the increase in area overhead for R-3PO-L3 is about 1%. Table 5 shows the total area overhead for each network.

6. Conclusions

In this paper, we propose R-3PO that uses emerging photonic interconnects and 3D stacking to reduce the optical power losses found in 2D planar on-chip networks by decomposing a large 2D photonic



(a) (b)

Figure 12: (a)Laser power variations as a function of (a) receiver sensitivity and the number of wavelengths, and (b) ring filtering loss and the number of waveguides.

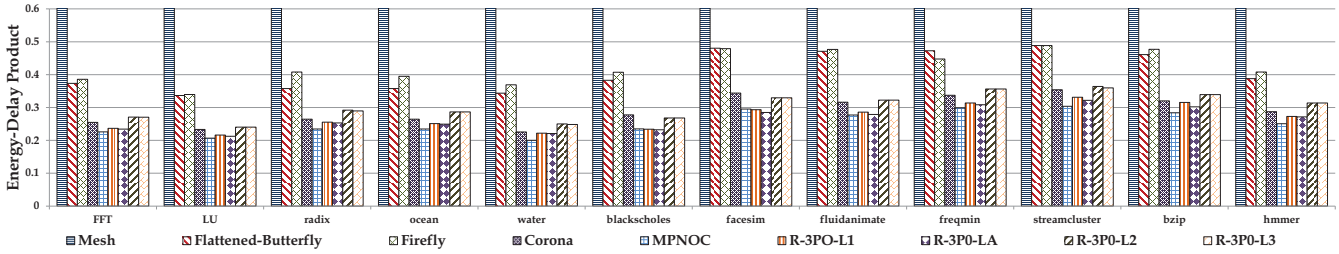


Figure 13: Simulation speed-up for 64-core using SPLASH-2, PARSEC and SPEC CPU2006 traffic traces using 8 wavelengths.

Table 4: Electrical and optical area overhead for select electrical and optical components

Component	Area
Electrical Link	0.0085 (mm^2)
Router (8×8)	0.128 (mm^2)
Photodetector receiver circuitry	0.02625 (mm^2)
Microring resonator	100(μm^2)
Photodetector	100(μm^2)
Waveguide	5.5 μm

Table 5: Electrical and optical area overhead for various networks (mm^2).

Network	Electrical	Optical
Firefly	712.25	78.5
Corona	107	78.5
R-3P0	407	70.9

crossbar into multiple smaller crossbars. In addition, we proposed a reconfiguration algorithm that maximizes the available bandwidth through run-time monitoring of network resources and dynamically re-allocating channel bandwidth. The reconfiguration algorithm improves performance by dynamically load balancing the network bandwidth and provides fault tolerance by bypassing faulty channels. For 64-core reconfigured network, our simulation results showed that the performance can be further improved by 10%-25% for Splash-2, PARSEC and SPEC CPU2006 benchmarks, where as simulation results for 256-core chip indicate a performance improvement of more than 25% while saving 6%-36% energy when compared to state-of-the-art on-chip electrical and optical networks.

Acknowledgment

This research was partially supported by NSF awards, ECCS-0725765, CCF-0915537, CCF-0915418, CCF-1054339 (CAREER) and ECCS-1129010 and by the IR/D program while Ahmed Louri was serving at the National Science Foundation.

References

[1] J. Ahn, M. Fiorentino, R. G. Beausoleil, N. Binkert1, A. Davis, D. Fattal, N. P. Jouppi, M. McLaren, C. M. Santori, R. S. Schreiber, S. M. Spillane, D. Vantrease, and Q. Xu, "Devices and architectures for photonic chip-scale integration," *Applied Physics A: Materials Science and Processing*, vol. 95, no. 4, pp. 989–997, June 2006.

[2] K. Aisopos, C.-H. O. Chen, and L.-S. Peh, "Enabling system-level modeling of variation-induced faults in networks-on-chip," in *48th Design Automation Conference (DAC)*, 2011.

[3] 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. Stojanovi, and K. Asanovic, "Building manycore processor-to-dram networks with monolithic silicon photonics," in *Proceedings of the 16th Annual Symposium on High-Performance Interconnects*, August 27-28 2008.

[4] R. G. Beausoleil, P. J. Kuekes, G. S. Snider, S.-Y. Wang, and R. S. Williams, "Nanoelectronic and nanophotonic interconnect," *Proceedings of the IEEE*, vol. 96, no. 2, pp. 230–247, February 2008.

[5] A. Biberman, K. Preston, G. Hendry, N. Sherwood-Droz, J. Chan, J. S. Levy, M. Lipson, and K. Bergman, "Photonic network-on-chip architectures using multilayer deposited silicon materials for high-performance chip multiprocessors," *J. Emerg. Technol. Comput. Syst.*, vol. 7, pp. 1–25, July 2011.

[6] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The parsec benchmark suite: Characterization and architectural implications," in *Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques*, October 2008.

[7] N. L. Binkert, A. Davis, N. P. Jouppi, M. McLaren, N. Muralimanohar, R. Schreiber, and J. H. Ahn, "The role of optics in future high radix switch design," in *ISCA*, 2011, pp. 437–448.

[8] X. Chen, L.-S. Peh, G.-Y. Wei, Y.-K. Huang, and P. Pruncal, "Exploring the design space of power-aware opto-electronic networked systems," in

11th International Symposium on High-Performance Computer Architecture (HPCA-11), February 2005, pp. 120–131.

[9] M. Georgas, J. Leu, B. Moss, C. Sun, and V. Stojanovic, "Addressing link-level design tradeoffs for integrated photonic interconnects," in *CICC*, 2011, pp. 1–8.

[10] B. Grot, J. Hestness, S. W. Keckler, and O. Mutlu, "Express cube topologies for on-chip interconnects," in *Proceedings of the International Symposium on High-Performance Computer Architecture (HPCA)*, 2009, pp. 163–174.

[11] A. B. Kahng, B. Li, L.-S. Peh, and K. Samadi, "Orion 2.0: A fast and accurate noc power and area model for early-stage design space exploration," in *in Proceedings of Design, Automation & Test in Europe Conference & Exhibition*, Nice, France, April 20-24 2009, pp. 423–428.

[12] J. Kim, W. J. Dally, and D. Abts, "Flattened butterfly: Cost-efficient topology for high-radix networks," in *Proceedings of 34th Annual International Symposium on Computer Architecture(ISCA)*, June 2007, pp. 126 – 137.

[13] A. K. Kodi and A. Louri, "Energy-efficient and bandwidth reconfigurable photonic networks for hpc systemss," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 17, pp. 384–395, April 2011.

[14] S. J. Koester, C. L. Schow, L. Schares, and G. Dehlinger, "Ge-on-soi-detector/si-cmos-amplifier receivers for high-performance optical-communication applications," *Journal of Lightwave Technology*, vol. 25, no. 1, pp. 46–57, January 2007.

[15] P. Koka, M. O. McCracken, H. Schwetman, X. Zheng, R. Ho, and A. V. Krishnamoorthy, "Silicon-photonic network architectures for scalable, power-efficient multi-chip systems," in *Proceedings of the International Symposium on Computer Architecture (ISCA)*, June 2010.

[16] P. Koonath and B. Jalali, "Multilayer 3-d photonics in silicon," *Opt. Express*, vol. 15, pp. 12 686–12 691, 2007.

[17] A. V. Krishnamoorthy, R. Ho, X. Zheng, H. Schwetman, J. Lexau, P. Koka, G. Li, I. Shubin, and J. E. Cunningham, "Computer systems based on silicon photonic interconnects," in *Proceedings of the IEEE*, vol. 97, no. 7, June 2009, pp. 1337–1361.

[18] 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 *ICCD 2007*, October 2007.

[19] Z. Li, M. Mohamed, X. Chen, E. Dudley, K. Meng, L. Shang, A. R. Mickelson, R. Joseph, M. Vachharajani, B. Schwartz, and Y. Sun, "Reliability modeling and management of nanophotonic on-chip networks," *IEEE Trans. VLSI Syst*, vol. 20, pp. 98–111, 2012.

[20] M. Martin, D. Sorin, B. Beckmann, M. Marty, M. Xu, A. Alameldeen, K. Moore, M. Hill, and D. Wood, "Multifacet's genreal execution-driven multiprocessor simulator (gems) toolset," *ACM SIGARCH Computer Architecture News*, no. 4, pp. 92–99, November 2005.

[21] C. Nitta, M. Farrens, and V. Akella, "Addressing system-level trimming issues in on-chip nanophotonic networks," in *Proceedings of the 17th International IEEE Symposium on High Performance Computer Architecture*, 2011, pp. 122–131.

[22] Y. Pan, J. Kim, and G. Memik, "Flexishare: Channel sharing for an energy-efficient nanophotonic crossbar," in *Proceedings of the 36th annual international symposium on High Performance Computer Architecutre (HPCA)*, 2010, pp. 1–12.

[23] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, and A. Choudhary, "Firefly: Illuminating future network-on-chip with nanophotonics," in *the Proceedings of the 36th annual International Symposium on Computer Architecture*, 2009.

[24] K. Preston, S. Manipatruni, A. Gondarenko, C. B. Poitras, and M. Lipson, "Deposited silicon high-speed integrated electro-optic modulator," *Opt. Express*, vol. 17, pp. 5118–5124, 2009.

[25] N. Sherwood-Droz, K. Preston, J. S. Levy, and M. Lipson, "Device guidelines for wdm interconnects using silicon microring resonators," in *Workshop on the Interaction between Nanophotonic Devices and Systems (WINDS), co located with Micro 43*, December 5th 2010, pp. 15–18.

[26] V. Soteriou, N. Eisley, and L.-S. Peh, "Software-directed power-aware interconnection networks," *ACM Trans. Archit. Code Optim.*, vol. 4, March 2007.

[27] T. H. Szymanski, "Optical link optimization using embedded forward error correcting codes," *Jornal of Selected Topics in Quantum Electronics*, vol. 9, no. 2, pp. 647–656, March/April 2003.

[28] D. Vantrease, N. Binkert, R. Schreiber, and M. H. Lipasti, "Light speed arbitration and flow control for nanophotonic interconnects," in *MICRO 42: Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, 2009, pp. 304–315.

[29] D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. Jouppi, M. Fiorentino, A. Davis, N. Binker, R. Beausoleil, and J. H. Ahn, "Corona: System implications of emerging nanophotonic technology," in *Proceedings of the 35th International Symposium on Computer Architecture*, June 2008, pp. 153–164.

[30] S. Woo, M. Ohara, E. Torrie, J. Singh, and A. Gupta, "The splash-2 program: Characterization and methodological considerations," 1995, pp. 24–36.

[31] X. Zhang and A. Louri, "A multilayer nanophotonic interconnection network for on-chip many-core communications," in *Proceedings of the Design and Automation Conference (DAC)*, June 2010.

[32] X. Zheng, F. Liu, J. Lexau, D. Patil, G. Li, Y. Luo, H. Thacker, I. Shubin, J. Yao, K. Raj, R. Ho, J. Cunningham, and A. Krishnamoorthy, "Ultra-low power arrayed cmos silicon photonic transceivers for an 80 gbps wdm optical link," in *in Optical Fiber Communication Conference*, March 2011.