PROBE: Prediction-based Optical Bandwidth Scaling for Energy-efficient NoCs

Li Zhou and Avinash Karanth Kodi
School of Electrical Engineering and Computer Science
Ohio University, Athens, OH 45701
E-mail: { lz792711, kodi }@ohio.edu

Abstract—**Optical interconnect is a disruptive technology solution that can overcome the power and bandwidth limitations of traditional electrical Networks-on-Chip (NoCs). However, the static power dissipated in the external laser may limit the performance of future optical NoCs by dominating the stringent network power budget. From the analysis of real benchmarks for multicores, it is observed that high static power is consumed due to the external laser even for low channel utilization. In this paper, we propose PROBE: Prediction-based Optical Bandwidth Scaling for Energy-efficient NoCs by exploiting the latency/bandwidth trade-off to reduce the static power consumption by increasing the average channel utilization. With a lightweight prediction technique, we scale the bandwidth adaptively to the changing traffic demands while maintaining reasonable performance. The performance on synthetic and real traffic (PARSEC, Splash-2) for 64-cores indicate that our proposed bandwidth scaling technique can reduce optical power by about 60% with at most 11% throughput penalty.**

I. Introduction

As chip multiprocessors (CMPs) have become an important approach to achieve better performance and power efficiency in the many-core era, tens and even hundreds of cores will be integrated on a single chip [1]. An energy and area efficient NoC architecture is becoming increasingly important to meet the multicore performance requirements. Recent research has explored optical interconnect as an alternative to traditional electrical signaling, due to high bandwidth density, low propagation latency, and distance-independent power consumption [2], [4], [5], [6], [7]. Research has shown that optical NoCs could significantly improve the achieved bandwidth by 2-6X, and reduce the power consumption by more than 10X when compared to electrical networks [4], [5], [10].

Although the benefits of optical interconnects are significant, there are several challenges [3], [9]. Thermal sensitivity and process variations of silicon devices are two potential issues that directly affect the operation of optical devices and impact the performance and reliability of optical NoCs. Different from traditional NoCs which consume large dynamic power, static power consumption induced by external laser, and microring resonators (MRRs) tuning dominate the power budget of optical NoCs. Research has shown that 74% of network power can be attributed to the laser and tuning power in a conventional radix-32 optical crossbar [11]. Reducing static power consumption of optical channels will require latency and bandwidth trade-off in future designs.

Prior work has reduced power consumption by exploring the non-uniform traffic and unbalanced resource utilization commonly seen in multicore applications. By gathering the information on past resource usage, the network predicts link and buffer utilization for each channel, and dynamically re-allocates the on-chip resources or tunes the voltage or frequency of the link [12], [13], [14], [15], [16]. For those channels that have low link utilization, the under-utilized bandwidth will be re-allocated to those channels that request more bandwidth, or save power by decreasing the voltage and frequency levels of the link. Recently proposed R-3PO architecture also benefits from 3D stacking of multiple layers and reconfigurable interconnects [15]. The effectiveness of dynamic resource re-allocation largely depends on the effectiveness of traffic prediction and reconfiguration implementation.

Current research on optical NoCs has not yet solved the issue of static power consumption when bandwidth demands are low across the network. In optical NoCs, low channel utilization leads to the problem of over provisioned bandwidth. The external laser is always switched on even under low network traffic that consumes unnecessary static power which can reduce the benefits of nanophotonics. As network traffic is often unbalanced and vary with different applications over time, future multicore applications require the network to provide sufficient bandwidth for traffic burstiness and fluctuation under high network load, while consuming reasonable power under low network load.

To address the static power problem and improve the bandwidth efficiency in optical NoCs, we propose **PROBE: Prediction-based Optical Bandwidth Scaling for Energy-efficient NoCs**, a history-based dynamic bandwidth scaling (DBS) scheme based on past link utilization. We tune the bandwidth of each channel according to the network traffic to meet the performance requirements by dynamically shutting-off portions of the network. We design an efficient scheme with a lightweight traffic predictor to increase the channel utilization and reduce static optical power. The DBS scheme is evaluated using synthetic traffic and traces from Splash2 [26], PARSEC [27] and SPEC CPU2006 [28] benchmarks. Our simulation results show that the DBS scheme achieves about 60% optical power saving with at most 11% throughput penalty. In summary, the major contributions of this paper include:

- We propose a dynamic bandwidth scaling (DBS) scheme that tunes the channel bandwidth adaptively to reduce the optical power consumption.

- We present a two-level bandwidth control mechanism to set the channel bandwidth globally while collecting resource utilization and tuning ring resonators locally.
- We design a lightweight traffic prediction scheme, and propose three different modes (Performance, Balanced and Power-aware) which provide latency/bandwidth trade-off.

II. Hardware Architecture and Implementation

A. Architecture Design

We initially discuss the optical architecture that will be used to test and evaluate our proposed DBS scheme. Figure 1 shows the layout of our network design. We combine four cores together and connect them with a shared L2 cache, which we call a tile. The layout consists of 16 tiles in a grid fashion with 4 tiles in x and y-directions. All tiles in each direction are fully connected. Instead of using a single off-chip laser, the ith-tile is assigned with a ith laser that provides the optical signal for all the output channels of that tile in either x or y-direction. Each laser is associated with an off-chip voltage regulator (VR) that tunes the power supplied to the waveguide. For example, in Figure 1, L_0 is assigned to Tile 0 to provide the optical signal for the output channel of Tile 0 in x and y-directions, and VR_0 is used to tune the supply power of L_0.

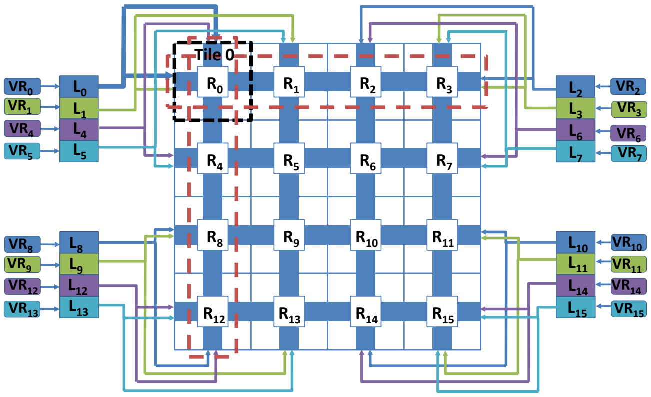


Fig. 1. Layout of the network design(R_i: Router connected to Tile i, L: laser, VR: voltage regulator).

Inter Tile Communication: The inter tile communication is same in either x or y-direction. One waveguide is assigned to each tile. By using multiple splitters, each waveguide splits into six channels that connect to the other six tiles in x and y-direction [7]. This design requires three channels per tile or a total of 12 channels for each direction. The waveguides are implemented in a U-shape, which allows for optical data to be modulated on the channels on the first pass and for the optical data to be received on the second pass.

B. Tunable Silicon Nanophotonic Device and Components

1) Tunable Power Ratio Splitter: Optical power splitters are essential components that distribute optical power. The passive splitter has a fixed power ratio which leads to low power efficiency. It is desirable to tune the power ratio dynamically to improve the network utilization. Using multimode interference (MMI) devices, tunable power splitters can be implemented by inducing phase change on top of the multimode section. Varying power-splitting ratios can be achieved by only slightly biasing the refractive index [20], [21], [22]. The driving voltage of the tunable power splitter is 0.9 V, tuning speed is 6 ns, and the access waveguide is only 5 μm pitch, thereby making these devices compatible with 22 nm Complementary metal-oxide-semiconductor (CMOS) process [21], [22]. In this paper, we use the tunable 1x2 power splitters with 3 dB (50 : 50) power ratio under unbiased driving voltage, and a tuning range up to 20 dB ($\sim$99% input power outputs in either of the channels).

2) Three-level Binary-tree-based Waveguide Design: First, we introduce our dynamic optical channel design which is based on the binary-tree waveguide prototype proposed in [19]. Figure 2(a) shows a channel between any two tiles. The channel consists of a three-level binary-tree waveguide with four branch waveguides. Optical signal is split into two parts at each Y-branch by using the tunable power splitter with an on-chip voltage regulator. α_i is the power ratio of the splitter at ith-level Y-branch, β_i is the output power of ith branch. By varying the driving voltage of each splitter, the power ratio α_i can be changed at run time. Assuming the excess loss for ith Y-branch is e_i, so $(1-e_i)$ is the effective input power that is split into two parts. We assume the same excess loss e for each Y-branch. The equations for each β_i and the power loss due to $m-1$ times splitting are given below [19]:

$$\begin{aligned}\beta_i &= \alpha_i(1-e)^i \prod_{k=1}^{i-1}(1-\alpha_k), 1 \le i \le m-1, \\ \beta_m &= (1-e)^{m-1} \prod_{k=1}^{m-1}(1-\alpha_k)\end{aligned} \tag{1}$$

$$power\_loss = -10\lg \sum_{i=1}^{m} \beta_i \tag{2}$$

where m is the number of branch waveguides which is 4 is our case.

With Dense Wavelength Division Multiplexing (DWDM) technique, up to 64 wavelengths can be transmitted within the same waveguide. The effective bandwidth of a optical interconnect is given by $B_w = W_N * W_{gN} * B_R$, where W_N is the number of wavelengths, W_{gN} is the effective number of waveguide branches, and B_R is the effective bit rate of the channel. With $W_N = 64$, $W_{gN} = 4$, and B_R = 5 Gb/s, we obtain a channel bandwidth of 1.28 Tb/s. By dynamically changing the driving voltage at the third splitter (from Figure 2(a)), we can transfer all the power from branch 4 to branch 3. Therefore, with $W_N = 64$, $W_{gN} = 3$, and B_R = 5 Gb/s, the channel bandwidth is tuned to 960 Gb/s. With varying

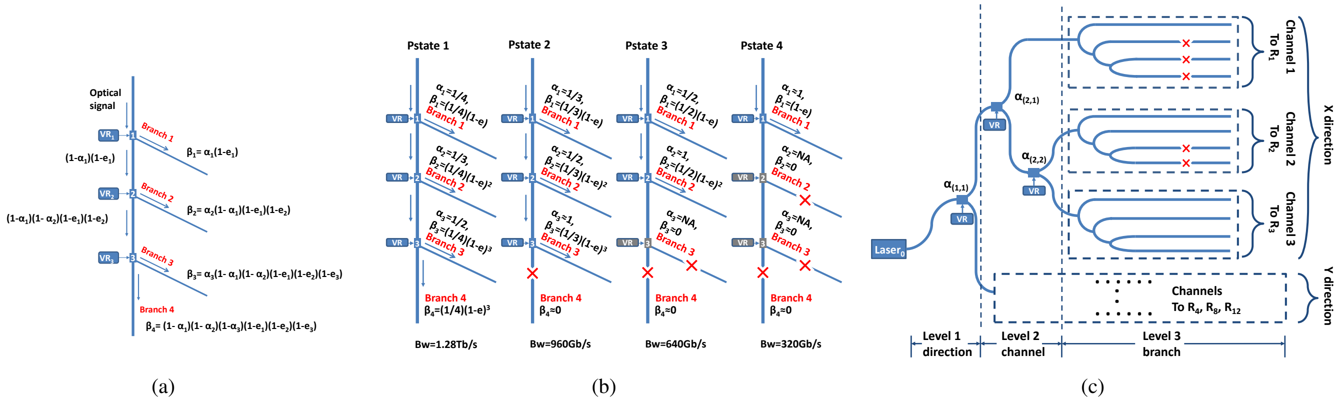


Fig. 2. Tunable channel and the three-level binary-tree-based waveguide. (a) Asymmetrically optical splitting of the channel. (b) Four power states of the channel using different power ratios of the splitters to accommodate the different bandwidth demands. (c) The structure of the waveguide assigned to tile 0, and the power states of the channels at cycle k.

number of W_{gN}, we define four power states of the channel: P_{state} 1 (W_{gN} = 4, B_w = 1.28 Tb/s), P_{state} 2 (W_{gN} = 3, B_w = 960 Gb/s), P_{state} 3 (W_{gN} = 2, B_w = 640 Gb/s), and P_{state} 4 (W_{gN} = 1, B_w = 320 Gb/s). The proposed power states are based on the effective bandwidth of the channel and the implementation of the four power states is shown in Figure 2(b). As shown in Figure 2(b), a cross mark on a branch indicates no output power in such a branch. Reducing the number of branches requires less power from the laser. Moreover, the power loss due to splitting decreases as more branches are shut-off and the depth of the tree decreases. To further reduce the optical power, the microring resonators along the closed branches that are used to modulate/detect the signal can be temporarily powered off. Table I shows the channel bandwidth, splitter ratios and power loss for each of the four power states assuming a same access loss e (= 0.2 dB).

TABLE I
POWER STATES CONFIGURATION WITH BANDWIDTH AND CORRESPONDING SPLITTING RATIOS.

P_{state}	Bandwidth(Tb/s)	$\alpha 1$	$\alpha 2$	$\alpha 3$	power loss(dB)
1	1.28	1/4	1/3	1/2	0.49
2	0.96	1/3	1/2	1	0.39
3	0.64	1/2	1	NA	0.30
4	0.32	1	NA	NA	0.2

Based on the tunable channel, we design a three-level binary-tree-based waveguide that includes all output channels of a tile in x and y-directions. We present the structure of the waveguide including all output channels of the tile in Figure 2(c). As shown in Figure 2(c), laser 0 is used to provide the optical signal. At level 1, the waveguide splits into two direction waveguides, x and y-direction. At level 2, each x or y-direction waveguide splits into 3 channels, each connecting to a tile in the same direction. At level 3, each channel splits into 4 branch waveguides. In each level, we use $\alpha_{(i,j)}$ to indicate the power ratio of the splitter of jth Y-branch in ith-level, each splitter is associated with a voltage regulator to tune the power ratio at run time. We take x-direction as an example to calculate the power ratio of each splitter. At level 1, $\alpha_{(1,1)}$ is calculated by the output power in x-direction waveguide over effective input power from the laser. At level 2, $\alpha_{(2,1)}$ is calculated by the output power in channel 1 over the effective input power from x-direction waveguide, and $\alpha_{(2,2)}$ is the ratio of the output power in channel 2 over the total power in channels 2 and 3. As we assume the same receiver sensitivity for each receiver, we estimate the required power by the number of effective branches. At level 3, each channel is in one of the four power states, and the power ratio of each splitter is set according to its power state and pre-defined in Table I. The total splitting power loss is given below:

$$power\_loss = -10lg\frac{\sum_{i=1}^{m} on_i * (1-e)^{l_i}}{\sum_{i=1}^{m} on_i} \qquad (3)$$

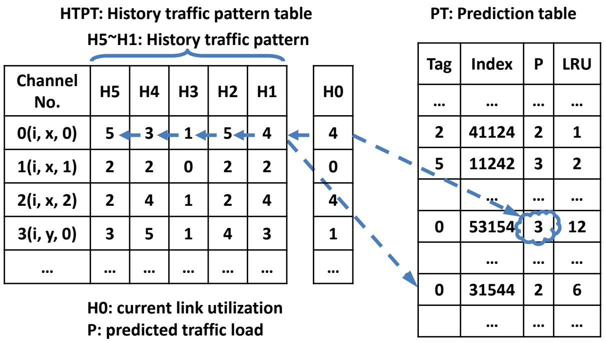
where m is the total number of leave branches of the three-level binary-tree, on_i equals 0 when branch i is closed, otherwise it equals 1, l_i is the level of the leave branch in the three-level binary-tree. In this paper, we apply the worst-case splitting power loss which is 0.98dB for the optical power estimation.

III. DYNAMIC BANDWIDTH SCALING (DBS) SCHEME

We introduce the scheme in two parts, first we depict our traffic statistics and prediction model, and second we discuss the dynamic tuning algorithm.

A. Traffic Statistics and Prediction

We adopt two traffic indicators such as link utilization ($Link_{util}$) and buffer utilization ($Buffer_{util}$) to track the network traffic load, and take measurements every cycle by using hardware counters [12]. Each hardware counter is associated with an optical transmitter. For each tile, there are three counters (one for each tile in each direction) to monitor the traffic utilization and provide the link and buffer information to a local Bandwidth Tuning Controller (BTC). All the statistics are measured over a sampling time called Reconfigurable



HTPT: History traffic pattern table

H5~H1: History traffic pattern

Channel No.	H5	H4	H3	H2	H1
0(i, x, 0)	5	3	1	5	4
1(i, x, 1)	2	2	0	2	2
2(i, x, 2)	2	4	1	2	4
3(i, y, 0)	3	5	1	4	3
…	…	…	…	…	…

H0
4
0
4
1
…

H0: current link utilization
P: predicted traffic load

PT: Prediction table

Tag	Index	P	LRU
…	…	…	…
2	41124	2	1
5	11242	3	2
	…	…	…
0	53154	3	12
	…	…	…
0	31544	2	6
	…	…	…

Fig. 3. Two-level history pattern traffic predictor. Channel 0(i, x, 0) denotes channel 0 in the x-direction of Tile i. H5 : H1 = (5, 3, 1, 5, 4) is the index for locating the predicted traffic load in prediction table (tag = 0, index = 53154, and p = 3 which predicts $Link_{util}$ between 0.4 $\sim$ 0.6) in last time interval. H0 is the current link statistic level which equals 4 ($Link_{util}$ between 0.6 $\sim$ 0.8). The predictor made a wrong prediction, so update the value P of the index(= 53154) with 4. Then make a left shift to H5:H0 as the arrows show in the figure, use new index H5 : H1 = (3, 1, 5, 4, 4) to find the predicted traffic load level (tag = 0, index = 31544, p = 2 ($Link_{util}$ between 0.2 $\sim$ 0.4)) for the next time interval.

Windows, R_w^t, where t presents the reconfiguration time. This sampling window impacts performance, as reconfiguring finely incurs latency penalty and reconfiguring coarsely may not adapt in time for traffic fluctuations [15]. In our performance section, we show that by utilizing network simulations to determine the optimum size for R_w. For calculation of $Link_{util}$ and $Buffer_{util}$ at configuration time t, we use the following equations [12], [15]:

$$Link_{util}^{t} = \frac{\sum_{t=1}^{R_w^t} Activity(t)}{R_w^t} \tag{4}$$

$$Buffer_{util}^{t} = \frac{\sum_{t=1}^{R_w^t} Occupancy(t)/Buffer_{size}}{R_w^t} \tag{5}$$

where Activity(t) is 1 if flit traverses the link in cycle t, or is 0 if no flit is transmitted on the link for a given cycle. Occupancy(t) is the number of buffer occupied at each time t, and $Buffer_{size}$ is the total number of buffers for the given link. As the bandwidth of the channel will vary over the time, the $Link_{util}^t$ is normalized to the full bandwidth when storing, and adjusted based on current bandwidth when predicting.

1) Design of Traffic Prediction: We implement two predictors, and choose the prediction from either predictors according the network traffic.

First Predictor: The first predictor is used under low traffic variation, named weighted traffic predictor. By taking a weighted average of current network statistics ($Link_{util}$ and $Buffer_{util}$) with past network statistics, the network will gradually tune the bandwidth to prevent temporal and spatial traffic fluctuations affecting performance. We calculate the link and buffer prediction ($Link_{util,pred}$ and $Buffer_{util,pred}$) as follows:

$$Predict_{util} = \frac{Past_{util} * weight + Current_{util}}{weight + 1} \tag{6}$$

where weight is a weighting factor and we set this to three in our simulations [23].

Second Predictor: However the traffic burstiness and sudden fluctuations will make the traffic trends hard to predict [17]. With the observation of that application behavior is repetitive, in this paper we extend the prior work in [16] to apply a two-level predictor which is called history pattern-based traffic predictor to obtain better prediction for the link utilization. The prediction of channel traffic will be chosen from the second predictor under high traffic variation, or from the first predictor under low traffic variation. Implementing the second predictor, each local BTC tracks the $Link_{util}$ of corresponding channels from previous k time intervals, and uses them to index a history pattern look-up table for traffic prediction. We first define five traffic load levels for the $Link_{util}$ by equally dividing the scale 0.0 $\sim$ 1.0 into 5 parts. Each part indicates a traffic load level, from the lowest level 1 ($Link_{util}$ between 0.0 $\sim$ 0.2) to the highest level 5 ($Link_{util}$ between 0.8 $\sim$ 1.0). We set k to five in our history pattern table design. The two-level predictor has two tables as shown in Figure 3. In each entry of history traffic pattern table, channel i(j, x/y, k) denotes channel i ($i = 1, 2, 3, ..., 6$) which is jth-channel ($j = 0, 1, 2$) in the x or y-direction of Tile i, and i is used as the tag in prediction table. H5 $\sim$ H1 are the quantized $Link_{util}$ levels for previous 5 time intervals R_w^{t-1} to R_w^{t-5} for the given link. H0 is the current $Link_{util}$ of the time interval R_w^t. In each entry of prediction table, tag is used to differentiate the channels. Index is used to look up the table for traffic prediction, and is a combination of H5 $\sim$ H1. P is the predicted $Link_{util}$. The replacement policy used in the prediction table is LRU. After each time interval, local BTC first compares the H0 of the channel with the predicted $Link_{util}$ for a given index(H5 : H1), update P if the predictor made a wrong prediction. Then the numbers from H5 to H0 make a left shift to get a new index for look-up in the prediction table to find the predicted traffic load for the next time interval R_w^{t+1}. If the entry is not found in prediction table, then H0 is used as the predicted traffic load instead. An example is shown in Figure 3.

2) Predictor Selector: The local BTC automatically choose the predicted link utilization from either the weighted traffic predictor or the history pattern traffic predictor according the traffic fluctuation. The first predictor, weighted traffic predictor is chosen under low traffic variation, and the second level history pattern traffic predictor is chosen under high traffic variation. We apply the saturating counter which is used in branch prediction for each link to select the traffic predictor according to the level of traffic variation. When current predictor gives wrong predictions twice, the local BTC will start to choose the prediction from the other predictor.

B. Algorithm Implementation

Based the predict link and buffer utilization, we introduce our dynamic bandwidth scaling algorithm as below: we design a two-level control system including 1) local BTC which is

TABLE II
RECONFIGURATION ALGORITHM FOR DBS.

Step 1:	Wait for Reconfiguration window, R_w
Step 2:	Each local BTC_i send a request packet to its local tile hardware counters for $Link_{util}$ and $Buffer_{util}$ from previous R_w^{t-1}
Step 3:	Each hardware counters sends $Link_{util}$ and $Buffer_{util}$ for previous R_w^{t-1} to its local BTC_i

Step 4a:	Each local BTC_i classifies the link statistic for each hardware counter as:
	if $Link_{util,pred} < Link_{util,lower}$ Under -Utilized: Decrease bandwidth
	if $Link_{util,lower} \leq Link_{util,pred} \leq Link_{util,upper}$ and $Buffer_{util,pred} \leq Buffer_{con}$ Normal-Utilized: Remain bandwidth
	if $Link_{util,pre} > Link_{util,upper}$ or $Buffer_{util,pred} > Buffer_{con}$ Over-Utilized: Increase bandwidth

Step 4b:	Each local BTC_i sends bandwidth adjustment information to the corresponding off-chip global BTC_j. Power off parts of the ring resonators if it decides to lower the bandwidth, or remains current configuration until a respond arrives
Step 5a:	Each global BTC_i checks the current power state of each corresponding local BTC_i, decides to upper or lower the bandwidth of corresponding channel according to requests from local BTC_is and the next power state each corresponding local BTC_i should be
Step 5b:	Each global BTC_i looks up the power ratio configuration table, then tunes the power supply of each corresponding laser and sends a response packet to each corresponding local BTC_i
Step 6:	Each local BTC_i enable portion of the ring resonators according to the response from the corresponding global BTC_i
Step 7:	Go to Step 1

located at each tile and 2) global BTC which is located near each off-chip laser. Each global BTC controls six local BTCs of a tile whose output channels are fed by the laser assigned to the global BTC. When varying the bandwidth of the channel, local BTC is responsible to enable/disable the MMRs and signal conversion back-end circuitry. Each global BTC maintains the power states of all corresponding channels, and collects the bandwidth requests from local BTCs. According to the bandwidth requests, each global BTC calculates the power ratio of each splitter, and is responsible to tune the laser and splitters. Table II shows the reconfiguration algorithm in PROBE.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance, power efficiency and overall overhead of proposed DBS scheme through simulations using synthetic traffic patterns and traces form real traffic.

A. Simulation Setup

A cycle-accurate network simulator is developed based on the Booksim [18] and modified to implement the baseline architecture and the proposed DBS algorithms. 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 [24]. The total delay of Electrical/Optical (E/O) and Optical/Electrical (O/E) conversion is reported less than 100ps [5], and is modeled as part of the link traversal. The nanophotonic transmission latency is amount to 1-2 cycles based on the physical location of source/desination pair. We assume a supply voltage V_{dd} of 1.0 V and a router clock frequency of 5 GHz [5], [6]. We assume an input buffer of 16 flits organized with 2 virtual channels, and each flit consisting of 256 bits. Dimension order routing algorithm and wormhole flow control are employed in our simulation. For the throughput analysis, packets are assumed to be single-flit packets.

We use full execution-driven simulator SIMICS with GEMS [25] integrated to extract traffic traces from real application in Splash-2, PARSEC and SPEC CPU2006 benchmarks. For Splash-2 traffic, the assumed kernels and workloads are as follows: FFT (16K particles), LU (512x512 with a block size of 16x16), Radiosity (Largeroom), Raytrace (Teapot), Radix (1 Million integers), Ocean (258x258), FMM (16K particles), and Water (512 Modules), seven PARSEC applications with medium inputs (blackscholes, facesim, fluidanimate, freqmin, streamcluster, ferret, and swaptions) and three workloads from SPEC CPU2006 (bzip, gcc base, and hmmer).

We ran several simulations to determine the optimum reconfiguration window size of R_w by varying the size from 100 to 5000 simulation cycles. We evaluated the latency and normalized power dissipation for random uniform traffic. While initially the performance improved with increasing window size as the decreasing influence of the transition latency penalty and more statistics are available which enable better prediction; at very large window sizes, the performance diminishes as the insensitive of the traffic load change. Our simulation results show that 1000 cycles for R_w showed the best performance. We assume a 100 cycles latency for the reconfiguration to take place after each R_w. It should be noticed that the reconfiguration latency only brings bandwidth delay when the channel request more bandwidth, and the local BTC waits for the signals from the global BTC indicating the laser tuning is done.

B. Power Model

The optical power budget is the sum of the laser power and the power dissipated in the MRRs. We adopt the nanophotonic devices and loss values from [4], [8], [9] and listed in Table III to calculate the required optical power. Based on the parameters, we layout the waveguides and estimate the required laser power. In this paper, we assume a flat thermal model that requires ring resonator heating power. The total laser power for the full bandwidth is 10.8 W, for ring heating is 25.85 W. The power consumption by an external laser is: $P_{laser} = \eta * I_{bias} * V_{bias}$ where η is the wall-plug laser efficiency, I_{bias} is the average driving current, and V_{bias} is the driving voltage [12]. According to the variable total optical power loss along the waveguide and the receiver sensitivity requirement, we provide power of the laser by varying the driving voltage using off-chip voltage regulator and power sensor.

TABLE III
OPTICAL POWER LOSSES FOR SELECT OPTICAL COMPONENTS.

Component	Value	Unit
Laser efficiency	30%	
Coupler(Fiber to Waveguide)	1	dB
Waveguide	1	dB/cm
Splitter(Total in Worst-case)	0.98	dB
Non-Linearity	1	dB
Ring Insertion & scattering	1e-2∼1e-4	dB
Ring Drop	1.0	dB
Waveguide Crossings	0.05	dB
Photo Detector	0.1	dB
Ring Heating	26	μW/ring
Ring Modulating	500	μW/ring
Receiver Sensitivity	-26	dBm

C. Synthetic Workload

We defined three PROBE modes to provide the flexibility to improve the latency/bandwidth trade-off. Figure 4 presents the power-latency product under light (0.1 pakcets/node/cycle), medium (0.3 packets/node/cycle), and heavy (0.45 packets/node/cycle) injection rates for uniform random traffic with five different target bandwidth boundaries ($Link_{util,lower} \sim Link_{util,upper}$ in Table II). Intuitively, higher bandwidth boundary lead to more aggressive scaling of channel bandwidth, which result in larger delays and lower optical power consumption. Our objective is to increase the resource utilization by providing sufficient bandwidth and obtain acceptable performance. Based on the results, we define three bandwidth utilization boundaries as Performance Mode (0.2 $\sim$ 0.4), Balanced Mode (0.4 $\sim$ 0.6), and Power-aware Mode (0.6 $\sim$ 0.8).

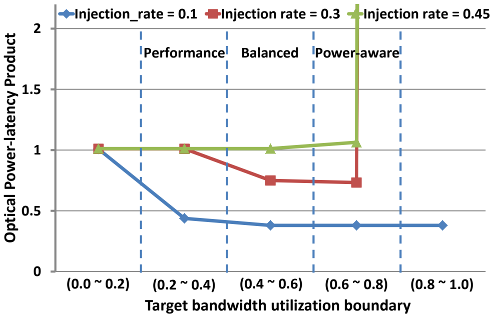


Fig. 4. Three modes for latency/bandwidth trade-off.

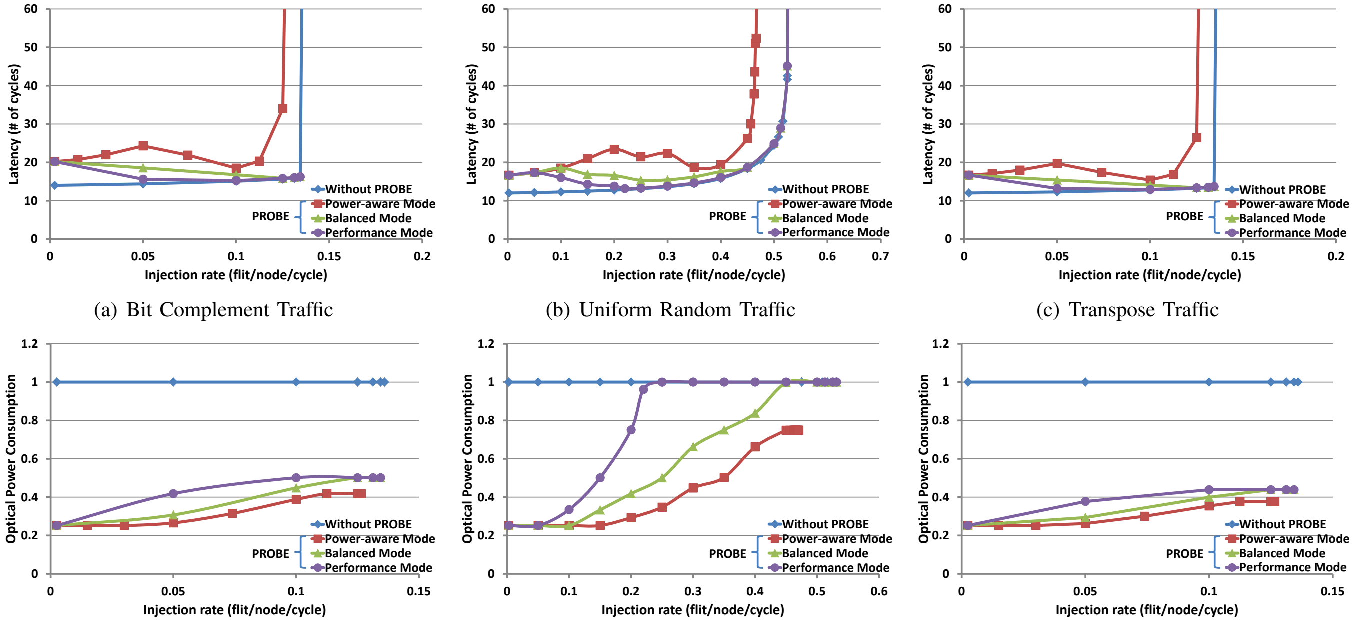
In Figure 5, we compare the latency/throughput and normalized optical power consumption under bit complement, uniform random, and transpose. It shows that our proposed scheme saves significant static optical power while only losing a moderate performance. When the network works at low injection rate, DBS scheme provides minimum bandwidth to obtain the maximum power saving for all traffic patterns under the three modes. However, the zero-load latency only increases a little bit due to the optical signaling. For example, under very low network traffic, most channels work under P_{state} 4, the zero-load latency of the three traffics only increases 5 cycles but the network obtains about 75% optical power saving.

With the increasing traffic load, DBS scheme provides more bandwidth gradually to mitigate the increased latency. The states of the channels switch from P_{state} 4 to P_{state} 1 step by step due to the increasing bandwidth demand. We take uniform random traffic (from Figure 5(b)(e)) as an example to explain the differences between the three modes. Performance and balanced modes start to increase the channel bandwidth to reduce the latency after 0.05 and 0.1 flit/node/cycle respectively, while power-aware mode still maintains the channel bandwidth to optimize the optical power saving until 0.15 flit/node/cycle. After 0.21 flit/node/cycle, performance mode provides high bandwidth and ensures the network performance without any latency penalty. Balanced mode saves at most 50% optical power before 0.45 flit/node/cycle with less then 3 cycles latency penalty when compared with performance mode. Power-aware mode achieves higher bandwidth utilization and more power saving by sacrificing minor performance. The non-smooth latency curve of power-aware mode is due to the fluctuation of the $Link_{util}$ near the bandwidth boundary. With heavy network traffic closing to the congestion point, performance and balanced mode has the same latency and throughput when compared to the baseline, while power-ware mode saves 25% optical power with about 11% throughput penalty.

For traffic with little locality such as bit complement and transpose permutation traffic as shown in Figure 5(a)(d)(c)(f), the load/latency curves have the same trend with uniform random traffic, but more optical power savings. DBS scheme provides minimum bandwidth to those channels that are idle. For example, in bit complement traffic performance and balanced modes save about 55% and 58% optical power respectively even after network congestion without any performance penalty.

D. Trace-Based Workload

Network traces from Splash-2, PARSEC, and SPEC CPU2006 benchmarks are used to evaluate the performance of proposed scheme. We compare the execution time and optical power consumption for our proposed scheme with full channel bandwidth. In Figure 6, the execution times of performance mode among all the benchmarks are quite close to the baseline. However, performance mode provides optical power savings due to the long-term low network traffic. For $facesim$, $swaps$, $barnes$ and lu which have almost the same execution time with the baseline, they still save about 59% $\sim$ 71% optical power. Balanced mode achieves about 70% average optical power saving over all the benchmarks with acceptable 11% execution time penalty on average, except for $ferret$ which takes 20% more time to execute. Power-aware mode obtains about 72% average optical power saving, however it takes 25% more execution time on average. For $ferret$, $streamcluster$ and $hmmer$ traffic, although the optical power savings are over 70%, the 40% more execution time penalty are too much. This could be improved by define the high traffic load channels



(a) Bit Complement Traffic

(b) Uniform Random Traffic

(c) Transpose Traffic

(d) Bit Complement Traffic

(e) Uniform Random Traffic

(f) Transpose Traffic

Fig. 5. Load-latency curve (a, b, c) and normalized optical power consumption (d, e, f) for Bit Complement, Uniform Random and Transpose traffic.

with balanced or performance modes to reduce the latency. We use Geometric Mean (GM) to evaluate all the three modes of our prosed scheme. Performance, balanced and power-aware mode has 3.5%, 10.1% and 24.3% execution penalty respectively, and optical power saving for the three modes are 64%, 69% and 72% respectively.

E. Design Cost of PROBE

The design cost of the two-level predictor has been reported in [16]. It takes around 0.017% of space in TILE64 and the estimated energy consumption per access is about 0.0039 nJ which is quite small and tolerable.

On-chip voltage regulator [29] is applied to tune the driving voltage of the power splitters and lasers. Since the power ratio of the splitters in the last level have four pre-defined ratios, so only four voltage regulators are required. We assign a voltage regulator for each power splitter in the first and second level. Given the 0.9 V tuning range, we estimate the power consumption for all 52 voltage regulators to be 0.8 W, and the total area is 0.0312 mm^2 by scaling to 22 nm. The tuning speed of voltage regulator is assumed to be 20 ns which is 40 networks cycles. The power tuning of off-chip lasers require 16 voltage regulators with 0.25 W estimated power consumption. So the total power consumption of voltage regulator modules and tunable splitters are estimated to be 1.05 W which only introduces about 3% power overhead compared to the optical power consumption.

The scalability of PROBE design could be achieved in different ways. One method of scaling the design is to group the lasers and combine their waveguides by increasing the number of levels (see in Figure 2(c)). For example, by grouping two lasers together, the total number of lasers could be cut in half. We leave this for future work.

V. Conclusions

In this paper, we propose a dynamic bandwidth scaling method that reduces the optical power consumption of optical channels based on past network traffic and resource utilization. To implement the proposed scheme, we design a binary-tree-based waveguide based on the tunable power splitter. We evaluate our scheme with synthetic traffic pattern and traces from Splash-2, PARSEC and SPEC CPU2006 benchmarks. The power is greatly reduced with acceptable performance penalty. For the synthetic traffic patterns, the power saving up to 75% compared with full bandwidth network and 11% loss in throughput at most. For the real traffic traces under balanced mode, it could save at average 68% optical power with at average 12% increase in execution time. Our design is cost-efficient and can be applied to other optical NoCs by re-designing the waveguide structures.

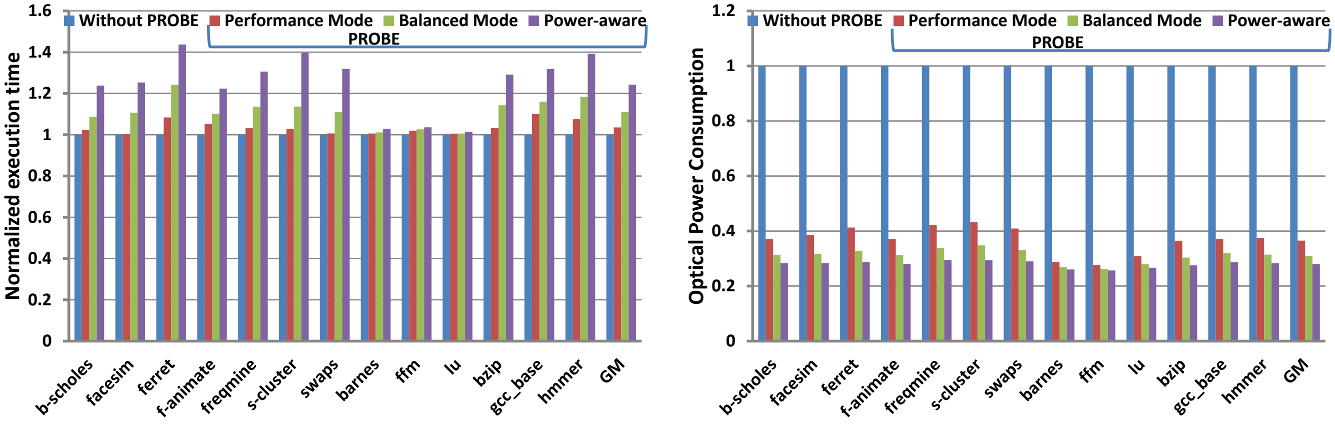
Acknowledgment

We thank the anonymous reviewers for their excellent feedback. This work was partially supported by the National Science Foundation grants CCF-0915418, CCF-1054339 (CAREER) and ECCS-1129010.

References

[1] J. Howard et al. “A 48-Core IA-32 Message-Passing Processor with DVFS in 45nm CMOS,” *In IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* pp. 108-109, 2010

[2] A. Shacham, K. Bergman, and L. P. Carloni, “Photonic Networks-on-Chip for Future Generations of Chip Multiprocessors,” *In IEEE Transactions on Computers*, vol. 57, no. 9, pp. 1246-1260, Sep, 2008.



(a) Normalized Execution Time

(b) Normalized Optical Power Consumption

Fig. 6. Simulation normalized execution time and optical power consumption of 64-core with different bandwidth for Splash-2, PARSEC, and SPEC CPU2006 benchmarks (GM: Geometric Mean).

[3] R. K. Dokania and A. B. Apsel, “Analysis of Challenges for On-Chip Optical Interconnects,” *In Proceedings of the 19th ACM Great Lakes symposium on VLSI*, pp. 275-280, 2009.

[4] C. Batten et al. “Building Manycore Processor-to-DRAM Networks with Monolithic Silicon Photonics,” *In Proceedings of the 16th Annual Symposium on High-Performance Interconnects (HOTI)*, August, 2008.

[5] D. Vantrease et al. “Corona: System Implications of Emerging Nanophotonic Technology,” *In Proceedings of the International Symposium on Computer Architecture (ISCA)*, pp. 153-164, 2008.

[6] P. Yan, K. Prabhat, K. John, M. Gokhan, Z. Yu, and C. Alok, “Firefly: Illuminating Future Network-on-Chip With Nanophotonics,” *In Proceedings of International Symposium on Computer Architecture (ISCA)*, 2009.

[7] R. Morris and A. Kodi, “Exploring the Design of 64- and 256-Core Power Efficient Nanophotonic Interconnect,” *In IEEE Journal of Slected Topics in Quantum Electronics*, vol. 16, no. 5, pp. 1386-1393, 2010.

[8] J. Ahn et al. “Devices and Architectures for Photonic Chip Scale Integration,” *Applied Physics A: Materials Science and Processing*, vol. 95, no. 4, pp. 989-997, June 2006.

[9] A. Joshi et al. “Silicon-photonic Clos Networks for Global On-Chip Communication,” *In Proceedings of the 2009 3rd ACM/IEEE International Symposium on Networks-on-Chip (NOCS)*, pp. 124-133, 2009.

[10] C. Batten, A, Joshi, V. Stojanovic, and K. Asanovic, “Designing Chip-Level Nanophotonic Interconnection Networks,” *In IEEE Journal of Emerging and Selected Topics in Circuits and Systems*, vol. 2, pp. 137-153, 2012.

[11] Y. Pan, J. Kim, and G. Memik, “Flexishare: Channel Sharing for An Energy-efficient Nanophotonic Crossbar,” *In International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 1-12, 2010.

[12] X. Chen, L-S. Peh, G-Y. Wei, Y-K. Huang, and P. Prucnal, “Exploring the Design Space of Power-Ware Opto-electronic Network Systems,” *International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 120-131, 2005.

[13] A. K. Kodi and A. Louri, “Performance Adaptive Power-aware Reconfigurable Optical Interconnects for High-performance Computing (HPC) Systems,” *In Proceedings of the 2007 ACM/IEEE conference on Supercomputing*, pp. 6:1-6:12, 2007.

[14] C. A. Nicopoulos, D. Park, J. Kim, N. Vijaykrishnan, M. S. Yousif, and C. R. Das, “ViChaR: A Dynamic Virtual Channel Regulator for Network-on-Chip Routers,” *In Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 333-346, 2006.

[15] R. Morris, A. K. Kodi, and A. Louri, “Dynamic Reconfiguration of 3D Photonic Networks-on-Chip for Maximizing Performance and Improving Fault Tolerance,” *In Proceedings of the 45th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2012.

[16] Y. S-C. Huang, K. C-K. Chou, C-T King, “Application-Driven End-to-End Trafc Predictions for Low Power NoC Design,” *In IEEE Transactions on Very Large Scale Integration System*, pp. 1-10, 2012.

[17] P. Bogdan, R. Marculescu, “Non-Stationary Trafc Analysis and Its Implications on Multicore Platform Design,” *In IEEE Transactions on Computer-Aided Design of Integrated Circuits and System*, Vol. 30, No. 4, 2011.

[18] W. J. Dally and B. Towles, “Principles and Practices of Interconnection Networks,” *Morgan Kaufmann Publishing Inc.*, 2004.

[19] B. Z. Fu, Y. H. Han, H. W. Li, and X. W. Li, “Accelerating Lightpath Setup Via Broadcasting in Binary-Tree Waveguide in Optical NoCs,” *In Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, pp. 933-936, 2010.

[20] J. Leuthold, and C. H. Joyner, “Multimode Interference Couplers with Tunable Power Splitting Ratios,” *Journal of Lightwave Technology*, vol. 19, no. 5, 2001.

[21] R. Thapliya, T. Kikuchi, and S. Nakamura, “Tunable Power Splitter Based on An Electro-optic Multimode Interference Device,” *Journal of Applied optics*, vol. 46, no. 19, 2007.

[22] R. Thapliya, S. Nakamura, and T. Kikuchi, “High Speed Electro-Optic Polymeric Waveguide Devices with Low Switching Voltages and Thermal Drift,” *In Proceedings of the Conference on Optical Fiber Communication (OFC)*, Feb, 2008.

[23] V. Soteriou, N. Eisley, and L.-S. Peh, “Software-directed Power-aware Interconnection Networks,” *ACM Trans. Archit. Code Optim*, vol. 4, March 2007.

[24] A. Kumar, P. Kundu, A. P. Singh, L.-S. Peh, and N. K. Jha, “A 4.6 Tbits/s 3.6 GHz Single-cycle NoC Router with a Novel Switch Allocator in 65nm CMOS,” *In Proceedings of the 25th International Conference on Computer Design (ICCD)*, pp. 63-70, October 2007.

[25] M. M. Martin et al. “Multifacets General Execution-driven Multiprocessor Simulator (gems) Toolset,” *ACM SIGARCH Computer Architecture News*, vol. 33, no. 4, pp. 92-99, 2005.

[26] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, “The SPLASH-2 Programs: Characterization and Methodological Considerations,” *ACM SIGARCH Computer Architecture News*, vol. 23, pp. 24-36, May 1995.

[27] 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*, pp. 72-81, October 2008.

[28] J. L. Henning, “SPEC CPU Suite Growth: An Historical Perspective,” *ACM SIGARCH Computer Architecture News*, vol. 35, pp. 65-68, March 2007.

[29] S. Kose, E. G. Friedman, S. Tam, S. Pinzon and B. McDermott, “An Area Efficient On-chip Hybrid Voltage Regulator,” *In Proceedings of the 13th International Symposium on Quality Electronic Design (ISQED)*, pp. 398-403, 2012.