PROWAVES: Proactive Runtime Wavelength Selection for Energy-efficient Photonic NoCs

Aditya Narayan, *Student Member, IEEE,* Yvain Thonnart, *Member, IEEE,* Pascal Vivet, *Member, IEEE,* and Ayse K. Coskun, *Senior Member, IEEE,*

***Abstract*—2.5D manycore systems running parallel applications are severely bottlenecked by network-on-chip (NoC) latencies and bandwidth. Traditionally, network-on-chips are composed of electrical links that exhibit constrained bandwidth, increased energy consumption at high-speed communication, and long latencies. Photonic Network-on-Chips (PNoCs) have been shown to provide high bandwidth at low latencies and negligible data-dependent power. However, the power overheads of lasers, thermal tuning, and electrical-optical conversion present major challenges against wide-scale adoption of PNoCs. A primary factor that impacts PNoC power is the number of activated laser wavelengths in the system. Applications' dynamic bandwidth needs provide the opportunity to selectively deactivate laser wavelengths when there is a lower bandwidth demand to alleviate high PNoC power concerns. This paper analyzes dynamic PNoC activity of applications at runtime so as to select laser wavelengths depending on an application's bandwidth requirements. The paper then proposes *PROWAVES*, a proactive runtime wavelength selection policy that forecasts the bandwidth needs and activates the minimum laser wavelengths for each application phase. We develop a cross-layer simulation framework to model the system performance, PNoC power and transient thermal distribution in a manycore system with PNoCs. We compare *PROWAVES* with prior system-level policies and our simulation results on a 2.5D system demonstrate that *PROWAVES* provides 18% and 33% power savings with only 1% and 5% loss in performance respectively, compared to activating all laser wavelengths in the system.**

***Index Terms*—2.5D manycore systems, Photonic NoCs, Wavelength selection, ARIMA time-series forecasting, Thermal tuning**

I. Introduction

Emerging compute-heavy and data-intensive applications in the domains of artificial intelligence, cloud, and high-performance computing demand higher parallelism and larger data transfers compared to past applications. Following the growing need for higher data processing capabilities, there is a rising interest in building chips with a dense integration of hundreds or thousands of logic cores. However, such a dense integration of cores result in larger die sizes and reduced manufacturing yields, contributing to substantially higher fabrication costs [1].

2.5D integration of multiple smaller chiplets on a large interposer is developing into a promising technology. Stow *et al.* [2] demonstrate that 2.5D interposer-based technology have significant cost reductions and higher manufacturing yield compared to 2D systems. Their work also shows 2.5D systems have a higher thermal efficiency over 3D systems. 2.5D integration also allows for a larger reuse of different IPs, thereby paving the way for heterogeneous integration [3]. Several works explore the performance and energy benefits of 2.5D systems in academia and industry [4], [5], [6].

In 2.5D manycore systems, as the on-chip wirelength gets longer, performance and energy efficiency are severely hampered by the network-on-chip (NoC) latencies and bandwidth. With the emergence of CMOS-integrated silicon-photonic technology, photonic NoCs (PNoCs) have been shown to provide high bandwidth, low latencies and low energy-per-bit communications, making PNoCs attractive for 2.5D systems [7], [8], [9]. A conventional PNoC consists of a laser that emits optical signals onto an on-chip waveguide. Microring resonators (MRRs) at the transmitting (Tx) and receiving (Rx) sites are utilized for modulation and filtering of the optical signals, respectively. A photodiode at Rx converts the filtered optical signal to an electrical signal. In addition, multiple optical signals can be multiplexed onto the same waveguide, enabling wavelength division multiplexing (WDM) that provides high internal bandwidth. Numerous works have demonstrated the feasibility of integrating photodiodes [10], low-loss waveguides [11], grating couplers [12], and MRR modulators and filters [13] through the use of slightly adapted or unmodified CMOS process, thereby paving the way for efficient realization of PNoCs.

Wide-scale adoption of silicon-photonic technology, however, is hampered by the high PNoC power consisting of lasers, electrical-optical (EO) and optical-electrical (OE) conversion, and the thermal tuning of MRRs [14]. First, MRRs exhibit high sensitivities towards on-chip thermal variations (TV) and process variations (PV), resulting in high thermal tuning power. Second, the overall PNoC power increases to support high bandwidth needs of data-intensive applications. Though device-level solutions use closed loop monitoring mechanism to perform controlled local heat injection for thermal tuning [15], [16], [17], such solutions do not account for runtime characteristics of applications. There is a strong diversity in application's runtime bandwidth needs and resource utilization that result in highly application-specific power and thermal profiles. *To address these two challenges, we develop system-level policies to determine the minimum required number of laser wavelengths based on application-specific bandwidth needs and further leverage the benefits of a low-level thermal*

Manuscript received XXX, 2019; revised XXX. This work was funded partly by the CARNOT institute and the French National Program Programme d'Investissements d'Avenir, IRT Nanoelec under Grant ANR-10-AIRT-05. This work was also funded partly by NSF CCF-1716352.

A. Narayan and A.K. Coskun are with the Department of Electrical and Computer Engineering, Boston University, Boston, MA, 02215 USA.

Y. Thonnart and P. Vivet are with Université Grenoble Alpes, CEA-Leti, Grenoble, France

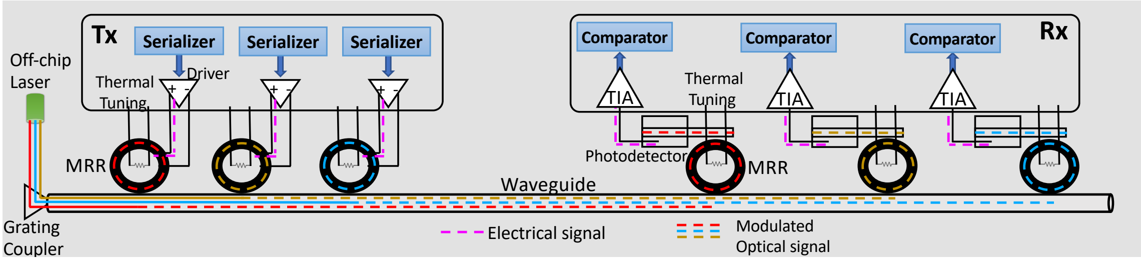


Fig. 1. An example silicon-photonic link. An off-chip laser emits 3 optical signals that are modulated by 3 MRRs at the Tx site and filtered by 3 MRRs at the Rx site.

control loop to perform thermal remapping with an eventual goal of reducing the overall PNoC power.

In our recent work [18], our wavelength selection policy, (*WAVES*, renamed as Static-Oracle WAVES (*SO-WAVES*) in this paper), introduces a mechanism to select the minimum number of laser wavelengths (λ_{min}) that can satisfy the average bandwidth needs of an application. In this paper, we demonstrate that the bandwidth needs of applications are highly dynamic during execution. Prior laser management policies, including *SO-WAVES*, are either agnostic to such dynamic behavior and/or perform offline characterization for wavelength selection [18], [19], [20], [21], [22], [23], [24], [25]. *Our observation that λ_{min} strongly correlates with the dynamic bandwidth needs of an application drives us to monitor these needs to predict PNoC activity and proactively select λ_{min} for each application phase.*

To this end, we first define a dynamic oracle wavelength selection policy (*DO-WAVES*). This policy determines the ideal minimum laser wavelength (λ_{min}) that is power-efficient for an application phase, given a performance loss threshold. Our goal is to design a dynamic runtime policy that achieves a λ_{min} as close as possible to the λ_{min} selected by *DO-WAVES*. Our specific contributions are as follows:

- We demonstrate that the bandwidth needs of an application are highly dynamic. Consequently, the number of wavelengths selected by our *SO-WAVES* policy [18] either over- or under-estimates the bandwidth needs during an execution phase. We quantify the average packet latency in the PNoC during each phase in the application execution, and show strong correlation between the average packet latency and the bandwidth needs during that phase.
- We design a dynamic runtime policy for wavelength selection, *PROWAVES*, which utilizes an AutoRegressive Integrated Moving Average (ARIMA) time-series predictor [26] to proactively select the minimum required wavelength, λ_{min}, for the next application phase. The predictor forecasts the average packet latency of the next application phase based on recent past trends. Using this forecasted average packet latency, *PROWAVES* then proactively activates or deactivates laser wavelengths to satisfy near-future bandwidth needs of that application. While *DO-WAVES* provides the ideal λ_{min} at each application phase, our practical implementation of *PROWAVES* is able to select λ_{min} that is within 12% on average of the λ_{min} selected by *DO-WAVES*.
- We develop a methodology that accounts for the PV and on-chip TV at each application phase. Our work is the first to model a low-level thermal control loop at the system-level to capture the effects of TV-induced shifts and enable MRR thermal remapping during each application phase. This dynamic thermal remapping enables *PROWAVES* to activate the best combination of λ_{min} that result in the lowest thermal tuning range.
- We design a simulation framework to evaluate *PROWAVES* against prior system-level policies on a 2.5D manycore system. Compared to a power scaling policy that uses a ridge regression model (*RR-PS*) [25], we demonstrate the benefits of modeling a thermal control loop that enables thermal remapping in *PROWAVES*. *PROWAVES* consumes $26.3W$ lower thermal tuning power than RR-PS. Furthermore, *PROWAVES* consumes an additional 10.2% and 16.4% lower PNoC power than our earlier *SO-WAVES* [18] for the same performance loss compared to the baseline.

The rest of this paper is organized as follows: Section II starts with a background on photonic communication. Section III provides an overview of related work. Section IV presents architectural details of the 2.5D manycore system used in our study. In Sec. V, we explain our experimental methodology. Section VI introduces our proposed *PROWAVES* policy. Section VII presents our evaluation results and Section VIII concludes this paper.

II. Background in Photonic Communication

Figure 1 illustrates an example of silicon-photonic link. A laser emits multiple optical signals with n different resonance wavelengths $\lambda_1, \lambda_2, .., \lambda_n$. In our work, we assume the laser is located off chip. The optical signals are carried by an optical fiber and coupled onto the on-chip waveguide using grating couplers. Several optical signals of different wavelengths can coexist within a single waveguide, allowing for WDM. At Tx, a cascade of n MRRs are designed to resonate at the same wavelength as the n optical signals from the laser. The output data at Tx is first serialized and then modulated onto one of the optical signals by an MRR that is resonating at the same wavelength as that optical signal. Depending on whether a logic 0 or 1 needs to be modulated onto the laser wavelength, an analog driver applies a corresponding voltage to the MRR. We discuss voltage modulation further in Sec. II-B. The modulated laser wavelength travels through the on-chip waveguide. At Rx, there is another set of n MRRs that are designed to resonate at the same wavelength as the n laser optical signals. The modulated optical signal in the waveguide is filtered out of the WDM bundle of channels by an MRR that is resonating at the same wavelength as that optical signal. The filtered optical signal is captured by a photodetector diode [27] that converts the optical signal into electrical signal. The detected electrical signal is amplified by a transimpedance amplifier (TIA), and read by a set of comparators that distinguishes the electrical signal as either logic 0 or logic 1.

A. Microring resonators (MRRs)

MRRs are integral components of WDM photonic communication. MRRs are utilized for narrow-band modulation and filtering of optical signals. An MRR utilizes a coupling mechanism to access the optical signal in a waveguide. MRRs are characterized by their small footprints, low modulation energies, wide working ranges and adjustable wavelength selectivity that allows for WDM [13]. When the coupled optical waves in an MRR loop builds up a round trip phase that is an integral multiple of 2π, the MRR is in resonance, diverting most of the optical power from the waveguide to the MRR. This phenomenon, called constructive interference, occurs when the MRR optical length is a whole number multiple of a particular laser wavelength. Therefore, an MRR is said to be in resonance with that laser wavelength when Eq. (2) holds true, where OPL denotes the optical path length, n_{eff} is the effective refractive index and r is the MRR radius:

$$OPL = n_{eff} \cdot (2\pi r), \tag{1}$$

$$\lambda = \frac{OPL}{m}, m = 1, 2, 3, ... \tag{2}$$

If a second waveguide is placed within coupling distance to the MRR, the waves also undergo constructive interference, thereby enabling the filtering of a given laser wavelength from a WDM waveguide.

The spectral parameters of an MRR are characterized by the free-spectral range (FSR), full-width at half maximum (FWHM) and the peak transmissions at resonance. FSR is the wavelength difference between two consecutive resonances, as depicted in Fig. 2. FWHM denotes the wavelength spectrum width at points that are the half the maximum power amplitude, and characterizes the sharpness of the resonance. It is related to the Q-factor, defined as the ratio between the resonant wavelength and the FWHM.

B. Voltage modulation

MRRs are doped to create a junction interfering with the optical path along the ring. This doping forms a p-n junction behaving in depletion mode i.e., electrons accumulated at the junction will be depleted when the junction is reverse biased, which increases n_{eff} and shifts the resonance wavelength to higher values. This phenomenon is fast, allowing for high-speed modulation up to $10-40Gbps$.

On the other hand, the doping can form a p-i-n junction, with an intrinsic zone along the MRR waveguide. In this case, the MRR behaves in injection mode i.e., forward biasing of the p-i-n junction injects carriers in the optical channel, thus reducing n_{eff} and shifting the resonance wavelength to lower values. This current-based injection is slower, operating around $100Mbps$. In addition, the resulting shift for p-i-n junctions is much larger than p-n junctions, which allows shifting of more than the FWHM. This property can be utilized to dynamically activate and deactivate MRRs as filters within a PNoC for routing of optical signals to different waveguides.

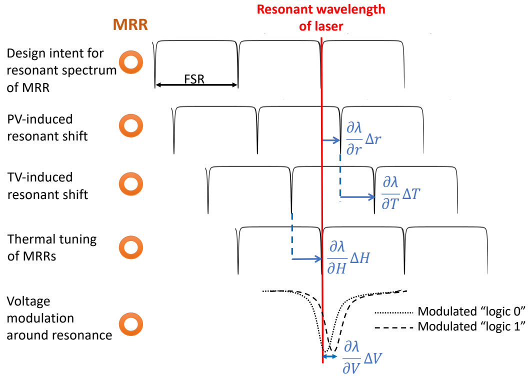

MRR
Resonant wavelength of laser
Design intent for resonant spectrum of MRR
FSR
PV-induced resonant shift
$\frac{\partial\lambda}{\partial r}\Delta r$
TV-induced resonant shift
$\frac{\partial\lambda}{\partial T}\Delta T$
Thermal tuning of MRRs
$\frac{\partial\lambda}{\partial H}\Delta H$
Voltage modulation around resonance
Modulated "logic 0"
Modulated "logic 1"
$\frac{\partial\lambda}{\partial V}\Delta V$

Fig. 2. MRRs are designed to resonate at wavelength of the laser. TV and PV induces shifts in the MRR resonant wavelength. The MRRs are supplied with heating power to tune back to laser wavelength.

C. Thermal and process sensitivity of MRRs

The non-idealities associated with CMOS fabrication process introduces variations in the thickness and width of the MRRs. These process variations have been demonstrated to induce shifts in the MRR resonant wavelengths. Krishnamoorthy *et al.* [28] show $1nm$ shift in MRR resonant wavelength for every $1nm$ of variation in MRR thickness or radius. As variations due to lithography, etch process and chemo-mechanical polishing are expected to be in the order of $10nm$, the PV-induced MRR resonant shifts become prominent. This implies that the exact MRR resonant wavelength cannot be fixed at design time, but needs to be accounted for each chip. These PV have a small random component and are mostly geometric variations. Hence, two distant MRRs across the die reticle can have an unknown resonant wavelength difference. Therefore, while it is possible to design MRRs grouped within about $1mm$ distance, it is not possible to predict the resonance variations between distant Tx and Rx MRRs.

TV introduces changes in the refractive index of an MRR, causing shifts in the resonant wavelength. The high thermo-optic coefficient of Si ($1.86 \times 10^{-4} K^{-1}$) makes it extremely vulnerable to TV-induced resonant wavelength shifts. The MRR resonant wavelength shift due to TV is given by Eq. (3):

$$\Delta\lambda_{shift} = \frac{d\lambda}{dT} \cdot \Delta T \quad . \tag{3}$$

In large manycore systems, high compute activity in the cores creates large TV and hot spots, which can reach high temperatures ($>85°C$) for compute-intensive applications. Moreover, these TV are not only temporal, but also spatial. Therefore, MRRs on the interposer can experience TV-induced resonance shifts between a completely cold state and peak activity, and distant MRRs can shift very differently depending on local activity.

Figure 2 shows an MRR that is designed to resonate at a laser wavelength. As the MRR is highly sensitive to PV and on-chip TV, the resonance peak of the MRR increases and the entire wavelength spectrum of the MRR shifts to the right.

D. Thermal tuning of MRRs

In order to compensate for PV- and TV-induced resonant wavelength shifts, the MRRs can be thermally tuned by controlled local heat injection as shown in Fig. 2. This is conventionally done via the Joule effect using resistive heaters inside the MRRs, thereby increasing the MRR wavelength. While Joule heating does not allow decreasing the resonant wavelength, it is always possible to heat up to a higher order of resonance. The maximum MRR wavelength shift is one FSR to align one of the MRR resonant peaks to the laser wavelength. The feasibility of a large wavelength shift by Joule heating is strongly dependent on MRR thermal sensitivity and the heater efficiency in terms of K/W.

Owing to WDM, it is possible to reduce the required amount of wavelength shift for an MRR. If n laser sources are used with resonant wavelengths evenly distributed within the FSR of an MRR, the maximum resonance shift required for any MRR is FSR/n to tune to the nearest laser wavelength. We can use n similar MRRs with small perimeter changes whose resonances are evenly distributed within an FSR. Thus, it is possible to tune all the MRRs to one laser wavelength each, forming a WDM group (either the Tx or Rx part of a WDM link), with no more than FSR/n shift for each MRR.

Thermal tuning with controlled local heat injection requires a closed-loop feedback system that monitors the MRR resonant shift due to TV and PV and the tuning required for an MRR to lock on to a laser wavelength. This is done by measuring the optical power on the drop port of the MRR with a photodiode. At the Rx, the optical power measurement is needed for signal demodulation, but we also incorporate power measurement at Tx to allow for monitoring and thermal tuning. Several techniques exist, either analog or digital, to close the feedback loop and derive a heating level from the optical monitoring of the drop port [16], [29], [17]. The heater maintains a fixed temperature within the MRR, so that the MRR resonance remains fixed to the laser wavelength.

However, a second level of control is needed, in order to handle the large temporal TV occurring at runtime. This control is responsible for wavelength locking to a different mapping of a WDM group. First, if the chip temperature increases close to the target MRR temperature, it is necessary to force a remapping to a higher order of resonance. Second, when the heater power gets higher, it may be possible to find a lower order of resonance that would require lower heater power. These remappings between n wavelengths are only possible if the heater efficiency is high enough that it can shift by more than FSR/n with some margin. As remapping requires larger amounts of thermally-controlled shifts, it is a relatively slow process of about $100\mu s$, but occurs less than once per second due to the thermal inertia of chips. Thonnart *et al.* [15], [30] provide more details on thermal tuning of MRRs, with silicon measurements, and wavelength remapping of a WDM link. Since the thermal control loop performs wavelength stabilization after MRR remappings, we assume negligible impact on device-level bit error rate (BER) arising from our system-level wavelength selection policies.

III. Related Work

2.5D-integrated systems with PNoCs have been extensively studied both in academia and industry because of their potential performance and thermal advantages. Oracle proposed the *Macrochip* [31], which integrates multiple manycore processors in a single package with different PNoC designs, providing low-power, higher inter-die communication bandwidth. *Galaxy* [32] is a multi-chip architecture that integrates multiple small chiplets through optical fibers and incorporates local electrical signaling for near-communication and photonic waveguides for distant intra-chiplet communication. Grani *et al.* [33] implemented a crossbar-based PNoC using arrayed waveguide grating router on a silicon interposer and demonstrated high bisection bandwidth at low energy-per-bit values.

A primary aspect in designing energy-efficient manycore systems with PNoCs is to address the tradeoff between achieving high bandwidth and reducing PNoC power consumption. Such tradeoffs are explored in various prior studies by enabling a higher number of channels for maximum aggregated bandwidth [20], via optimized wavelength allocation based on application task graph [21] or using an arbitration-free shared-channel PNoC [22].

There have been other system-level efforts to reduce the MRR thermal tuning power. *RingAware* [34] and *FreqAlign* [35] employ thread allocation and migration to reduce the thermal variations around communicating MRRs. *Aurora* [36] encompasses a cross-layer approach at the device, system and OS-level to control the thermal tuning power. Our approach is orthogonal to such runtime policies and can be combined with most prior studies to develop an energy-efficient PNoC.

Wavelength selection has also been studied in the context of power scaling in several related works. Winkle *et al.* [25] proposed a learning-based technique based on PNoC link utilization to determine the optimal number of laser wavelengths. Chen *et al.* [19] performed runtime wavelength selection on clos and butterfly network topologies based on the latency at each application phase. *R-3PO* [23] is a reconfigurable 3D-PNoC that monitors the bandwidth availability and performs runtime reconfiguration of PNoC bandwidth.

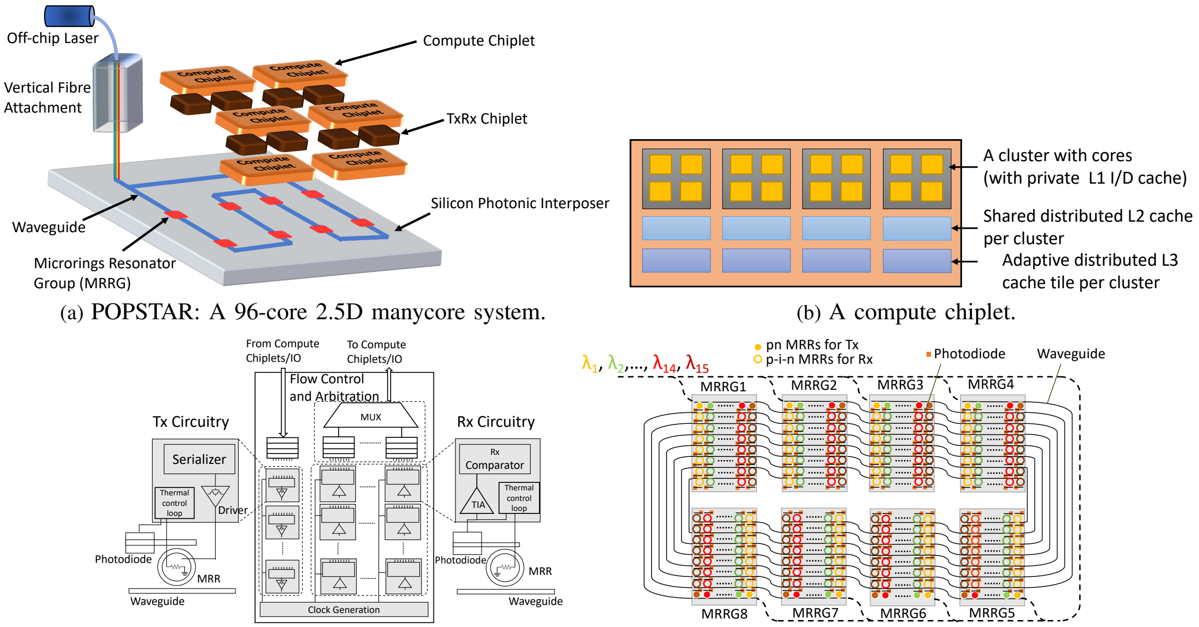
In contrast to prior system-level studies, the major distinguishing feature of *PROWAVES* comes from the awareness towards the underlying low-level thermal control loop. The low-level modeling of the thermal control loop enables us to evaluate the benefits of MRR thermal remapping, which reacts to the application's dynamic thermal profile. We evaluate the benefits of thermal control loop modeling in Sec. VII-A. Our work is also the first to consider the dynamic impact of on-chip TV and PV, leverage underlying device-level solutions, and design proactive system-level policies to improve the energy efficiency of 2.5D manycore systems with PNoCs. *PROWAVES* comes at a relatively low overhead in terms of latency and storage, and can be seamlessly integrated in a real system.

IV. System Architecture and Methodology

Our target system is a 2.5D manycore system of homogeneous cores, connected with PNoCs. Specifically, we focus on the Photonic Silicon inTerposer ARchitecture (*POPSTAR*) that was introduced in our recent work [18], [30].

A. Architecture overview

POPSTAR is a 2.5D-based manycore system with 96 cores organized in six compute chiplets and the electrical circuitry



(a) POPSTAR: A 96-core 2.5D manycore system.

(b) A compute chiplet.

(c) A TxRx chiplet.

(d) PNoC routing architecture and MRRG assignment.

Fig. 3. The 96-core POPSTAR architecture consists of six compute chiplets and eight TxRx chiplets integrated on a photonic interposer. An off-interposer laser emits 16 optical signals onto SWMR links on the interposer for communication.

for photonic communication organized in eight TxRx chiplets. The compute and the TxRx chiplets are stacked on a photonic interposer. Figure 3 illustrates the *POPSTAR* organization. An off-chip laser emits multiple optical signals that are carried onto the photonic interposer through a fiber attachment. Vertical grating couplers couple these optical signals between the waveguides on the interposer and the fiber attachment.

1) Compute chiplets: The 96 cores in *POPSTAR* are organized in six compute chiplets. The architecture of each compute chiplet is similar to the *TSARLET* chiplet [37]. *TSARLET* is a scalable cache-coherent chiplet architecture with full virtual memory support targeted for 3D integration onto an active interposer. Each chiplet is composed of four clusters with four cores in each cluster as shown in Fig 3b. In our study, we assume that the core architecture is similar to IA-32 core from the Intel Single-chip Cloud Computer (SCC) [38]. Each core has a private $16KB$ L1 I-cache and D-cache with full virtual memory support. We scale the dimensions of the IA-32 core to $28nm$, resulting in a single core area (including the L1 cache) of $0.93mm^2$. There is a shared distributed L2-cache with $256KB$ per cluster, and a distributed L3-cache, with 4 L3 tiles $(4 \times 1MB)$ per compute chiplet.

2) TxRx chiplets: Each TxRx chiplet is composed of the electronic circuitry for E-O and O-E conversion, as shown in Fig. 3c. Each of the six compute chiplet accesses the PNoC on the interposer through a TxRx chiplet. Two of the TxRx chiplets are connected to the external peripherals, IOs, and memory controllers. For each Tx MRR, there is a serializer and a modulation driver. Additionally, for each Rx MRR, there is a filter bias, a TIA and comparators. An analog thermal loop [15], detects the photodiode current, compares it with a reference bias current and supplies heating power to thermally tune the MRRs so that the detected photocurrent is equal to the reference current. This thermal loop locks an MRR to the nearest activated laser wavelength. Finally, FIFO queues and multiplexers handle the flow control and communicate with the compute chiplets using local 2.5D passive connections.

3) PNoC architecture: The PNoC on the interposer handles the inter-chiplet transfers consisting of the data and coherence traffic between an L1 and an L2 cache that are on separate chiplets, and an L2 and an L3 cache that are on separate chiplets. The interface between the compute and TxRx chiplet is serialized before injection in the PNoC depending on the number of activated laser wavelengths. The data rate of each laser wavelength is $12Gbps$, resulting in a peak aggregate bandwidth of $1.5Tbps$ on the interposer.

The global PNoC toplogy connecting the TxRx chiplets is a Single-Writer Multiple Reader (SWMR) topology [30]. The SWMR topology is mapped onto a U-shaped spiral of waveguides on the photonic interposer. Each TxRx chiplet owns a communication channel, spanning all activated laser wavelengths, where it writes its data to transmit. This channel passes by all TxRx chiplets along the spiral. Data is routed to the appropriate TxRx chiplet by electrical control lines from the transmitting TxRx chiplet driving the Rx MRRs of the receiving TxRx chiplet. The flow control of the packets is achieved by the FIFO buffers in each TxRx chiplet. As each channel is written by a single dedicated TxRx chiplet, in-network contention is prevented.

4) Microring resonators (MRRs): MRRs are responsible for modulating and filtering the optical signals. We use p-n junction MRRs for Tx high-speed modulation and p-i-n junction MRRs for Rx filtering. Each MRR has a radius of $10\mu m$, and designed around a center wavelength of $1310nm$ with an FSR of $10.8nm$. Thermal tuning of MRRs is achieved via dedicated local heaters that are fabricated with titanium.

5) Microring resonator group (MRRG): A group of MRRs consisting of Tx MRRs and Rx MRRs is organized under-

TABLE I
NOTATIONS USED

Notation	Description	Value
C	Number of TxRx chiplets (and waveguides)	8
λ_{tot}	# available laser wavelengths	16
λ_{act}	# activated laser wavelengths	1,2, .. 16
λ_{min}	# laser wavelengths required for an application	1,2, .. 16

TABLE II
POWER CONSUMPTION OF DIFFERENT ELEMENTS [14]

Component	Active Power		Idle Power	
	Notation	Value (mW)	Notation	Value (mW)
Laser (wall-plug)	P_L	30		0
Serializer	$P_{srl,a}$	3	$P_{srl,i}$	1
Driver	P_{drv}	3		0
Rx Comparator	$P_{cmp,a}$	1	$P_{cmp,i}$	0.33
TIA	P_{TIA}	2		0
Arbitration and Flow Control	$P_{arb,a}$	32	$P_{arb,i}$	10

neath a TxRx chiplet, termed as microring resonator group (MRRG). Each TxRx chiplet accesses the PNoC through a single MRRG. For every MRRG, there is one Tx waveguide and seven Rx waveguides coming from the other MRRGs forming a spiral of SWMR links as depicted in Fig. 3d. An MRRG consists of 16 WDM bundle of MRRs, with each WDM bundle operating at a different laser wavelength. A WDM bundle consists of a single Tx MRR and seven Rx MRRs. The Tx MRR in a WDM bundle modulates data on a laser wavelength. For data filtering, the seven Rx MRRs per WDM bundle are utilized to receive data from the seven other TxRx chiplets on that laser wavelength. This results in an aggregate of 16 Tx MRRs and 112 Rx MRRs in each MRRG.

B. Power consumption of the PNoC

The major components of power consumption along a PNoC include the electronic circuitry for E-O and O-E conversion, laser and the thermal tuning of MRRs. Table I lists the notations used in our work. In this section, we evaluate the power consumption as a function of λ_{act}. Table II displays the active and idle power consumption of different elements. These power values are determined from the post-layout simulations using PrimeTime power analysis [14].

In order to filter out the optical signal from waveguide at each Rx site, the input laser needs to provide the minimum power so that the optical power at Rx site of the longest SWMR link is above the photodetector sensitivity. We account for all the sources of power loss as the optical signal traverses through the waveguide, including the effect of PV on each MRR, and determine the worst-case power loss along the longest SWMR link in the waveguide. The laser source power of a single wavelength (P_L) should be higher than the sum of this worst-case power loss and the photodiode sensitivity. We calculate this value as $30mW$. The overall laser power, P_{laser}, for λ_{act} laser wavelengths can then be expressed as:

$$P_{laser} = P_L \cdot C \cdot \lambda_{act} \ . \tag{4}$$

The next contributor to the PNoC power is electronic circuitry for E-O and O-E conversion. We break down the overall EOE power consumption into the power consumed by the Tx circuitry, Rx circuitry and the logic for arbitration and flow control. In the TxRx chiplet, the serializer, comparators and arbiters are clocked for precise timing control. The number of elements that are active in the Tx, Rx and arbitration circuitry is a strong function of λ_{act}. We calculate the overall EOE power across all TxRx chiplets as follows:

$$P_{Tx} = P_{drv} \cdot \lambda_{act} + P_{srl,a} \cdot \lambda_{act} + P_{srl,i} \cdot (\lambda_{tot} - \lambda_{act}) \ , \tag{5}$$

$$P_{Rx} = P_{TIA} \cdot \lambda_{act} + P_{cmp,a} \cdot \lambda_{act} + P_{cmp,i} \cdot (\lambda_{tot} \cdot C - \lambda_{act}) \ , \tag{6}$$

$$P_{arb} = P_{arb,a} \cdot \frac{\lambda_{act}}{\lambda_{tot}} + P_{arb,i} \cdot \frac{\lambda_{tot} - \lambda_{act}}{\lambda_{tot}} \ , \tag{7}$$

$$P_{EOE} = C \cdot (P_{Tx} + P_{Rx} + P_{arb}) \ . \tag{8}$$

The heating power required to thermally tune the MRR back to resonance is another major power contributor of the overall PNoC power. As described in Sec. II-D, the analog thermal control detects the photodetector current, compares with the reference and supplies the appropriate heating power. The power of this feedback logic circuitry for tuning power is $0.15mW$ and is neglected in comparison to the heating power. In our system, we use the MRR thermal sensitivity of 78pm/K and a heater efficiency of $120pm/mW$. Given the small area footprint of an MRRG, we assume the transient temperatures within an MRRG to be same. Thus, all the MRRs within an MRRG undergo the same TV-induced resonance shift. However, due to variable chip activity across different compute chiplets, the thermal gradient across different MRRGs might be very high. Equation (9) calculates the overall wavelength shift ($\Delta\lambda_{shift}$) for an MRR, where ΔT is the difference between the MRRG temperature and the ambient temperature and $\Delta\lambda_{shift,PV}$ is the PV-induced wavelength shift:

$$\Delta\lambda_{shift} = \frac{d\lambda}{dT} \cdot \Delta T + \Delta\lambda_{shift,PV} \ . \tag{9}$$

In a single MRRG, one Tx MRR and seven Rx MRR are heated for every activated laser wavelength. We calculate the total heating power, P_{heat}, in Eq. (11) by aggregating the heating power of the MRRs over all the MRRGs, where $\Delta\lambda_{heat}$ is the required wavelength shift to the nearest laser wavelength for an MRR:

$$\Delta\lambda_{heat} = \frac{FSR}{\lambda_{tot}} - (\Delta\lambda_{shift} \ mod \ \frac{FSR}{\lambda_{tot}}) \ , \tag{10}$$

$$P_{heat} = \sum_{i=1}^{C} \sum_{r=1}^{C \cdot \lambda_{act}} \frac{\Delta\lambda_{heat_{ir}}}{\frac{d\lambda}{dH}} . \tag{11}$$

Since most of the heat flux from the heater flows through the substrate, thermal coupling is less than $0.02K/mW$ between adjacent MRRs [30].

V. SIMULATION FRAMEWORK

To evaluate the performance and power consumption of *POPSTAR* with different wavelength selection policies, we set up a simulation framework that is composed of a performance simulator, logic core power calculator, PNoC power model and a thermal simulator. Figure 4 depicts our toolflow. For simulating the performance of the multithreaded applications, we use Sniper [39] by modeling the architectural details of *POPSTAR*. We use McPAT [40] to estimate the logic power. We feed the logic and the PNoC power traces to HotSpot [41], [42] for transient thermal simulations.

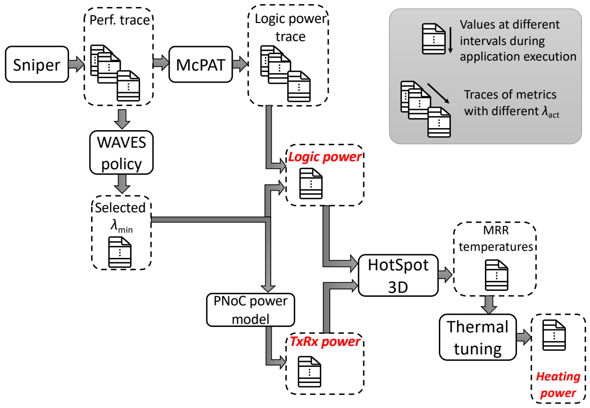


Fig. 4. Simulation framework for modeling performance, power and temperature of *POPSTAR* and performing wavelength selection.

A. *Performance simulation*

For our experiments, we use a diverse set of applications from PARSEC [43] (*swaptions, blackscholes*), UHPC [44] (*shock*), HPCCG [45] (*hpccg*) and NAS Parallel Benchmark [46] (*ep, is, mg, ft, bt, lu, sp*). We vary the inter-chiplet bandwidth to model the link bandwidth corresponding to λ_{act} in the system. In our experiments, we fast-forward the sequential instruction phase of all the applications. We execute each application for 10 billion instructions in the region of interest (ROI) or the full ROI if the application finishes earlier than 10 billion instructions. We collect performance traces, including the PNoC metrics (number of packets transferred in the PNoC and total queue delay) in Sniper at every 100 million instruction from the beginning of ROI. The size of a single interval, unless otherwise mentioned, is 100 million instructions. To study the benefits of wavelength selection with different system utilization, we run each application with varying number of thread counts (24, 48, 72 and 96 threads).

B. *Power simulation*

We feed the performance statistics from Sniper as input to McPAT at each interval. To perform the calibration of McPAT power numbers, we estimate the average power in McPAT obtained from all our experiments. We calculate the calibration factor by scaling the McPAT average power to the published average power of the IA32 core ($0.83W$ at $V_{DD} = 0.85V$, $f = 533MHz$) [38], and use this factor to scale the McPAT power numbers. We assume that the idle cores are put to sleep and consume negligible power.

We calculate the laser and EOE power using at each interval using our analytical PNoC power model, as described in Eq. (4)-(8). The aggregate heating power required to thermally tune all the MRRs is calculated using Eq. (10)-(11). Sec. V-C explains the thermal simulation to determine MRR temperatures. We model the local MRRG PV as a gaussian distribution with a standard deviation of $100pm$ [30].

Since the leakage power component is strongly dependent on temperature, we implement a temperature-dependent leakage power model in our thermal simulator. We extract the linear leakage power model from the published data for Intel $22nm$ Ivy Bridge processors [47], [35]. Since our operating range covers a limited range of temperature, we assume the leakage current dependence on temperature to be linear [48].

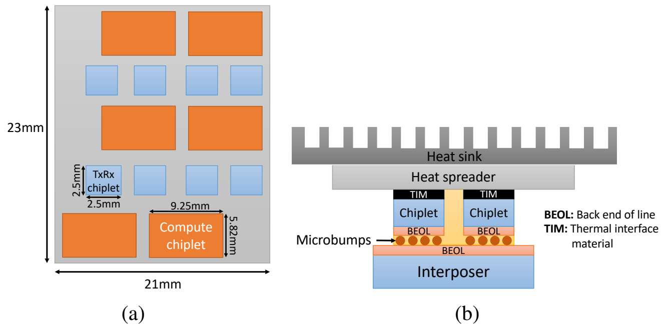


Fig. 5. Layout and cross-sectional view of *POPSTAR*

TABLE III
MATERIAL PROPERTIES AND DIMENSIONS

Layer	Thickness (μm)	Thermal conductivity ($W/m.K$)	Specific heat ($J/kg.K$)	Density (kg/m^3)
Heat sink	6900	400	396	8960
Heat spreader	1000	400	396	8960
TIM	10	6.8	900	1300
Chiplets	750	150	700	2330
BEOL	10	145	612	4237

Microbump	Pitch=40,
	diameter=20
		0.86	846	2689

Interposer	750	150	700	2330

C. *Thermal simulation*

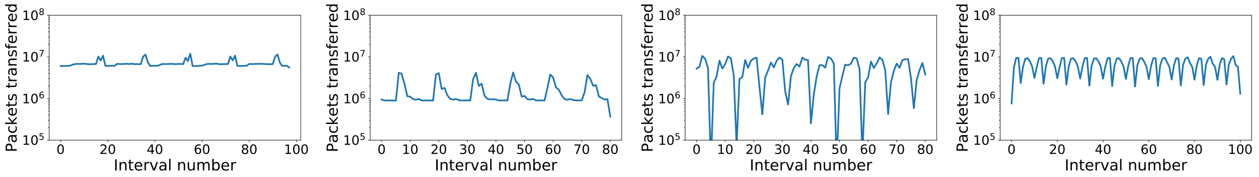
We use the 3D extension of HotSpot [41], [42] to determine the transient thermal profile of each MRRG. In order to obtain an accurate thermal map, we calibrate the HotSpot temperatures to temperatures obtained from Project Sahara [49], which is a sign-off thermal tool from Mentor, a Siemens Business, for simulating detailed 3D circuits within its package and board. We obtain HotSpot temperatures to be within 2% error margin of Project Sahara on average. Figure 5 shows the floorplan and cross-sectional view of *POPSTAR* that we model in HotSpot. We use a grid size of 64×64 for our thermal simulations in HotSpot. Table III shows the material properties of all layers in *POPSTAR* that are used in our thermal simulations.

We use an ambient temperature of $310K$, and assume that the MRRs are designed to resonate at laser wavelengths at this temperature. Transient thermal simulations are initialized with temperatures from a steady state simulation. As we incorporate the temperature-dependent leakage model that in turn changes the power traces, we run each transient thermal simulation for a second round to ensure convergence of temperature.

VI. WAVELENGTH SELECTION IN PNOCS

A. *Need for wavelength selection*

1) Impact of λ_{act} on PNoC power: The peak aggregate bandwidth of a PNoC is the product of λ_{act} and the modulation bit rate of a single laser wavelength. For applications with high inter-chiplet communication, it is intuitive that a higher λ_{act} provides increased communication bandwidth, and therefore, is desirable for higher performance. However, as evident from the Equations 4-10, the power consumption along a PNoC increases with λ_{act}. Therefore, it is essential to address this bandwidth-power tradeoff and activate the minimum number of laser wavelengths, λ_{min}, that sufficiently caters to the required bandwidth needs of an application.



(a) *bt* (b) *ep* (c) *shock* (d) *lu*

Fig. 6. Inter-chiplet packets transferred during application execution. Each interval consists of 100million instructions. Applications have phases where a higher number of packets are transferred compared to other phases and these phases exhibit periodic behavior.

2) Bandwidth characteristics of applications: Figure 6 illustrates the number of network packets transferred in the PNoC at various intervals during an application's execution. Therefore, an application's bandwidth needs during its execution is highly dynamic and periodic. Since an application's bandwidth needs is highly dynamic, executing the entire application with a single λ_{min} may not yield the maximum PNoC power savings. In order to unearth the maximum benefits of wavelength selection, we argue for the need of a dynamic wavelength selection that monitors the bandwidth needs at different intervals and activates the λ_{min} for each interval.

B. Static Oracle WAVES (SO-WAVES)

SO-WAVES [18] is a static wavelength selection policy that determines the λ_{min} for an application based on offline analysis using a set performance loss threshold (L_{thr}). The performance loss is calculated from the case where all the laser wavelengths in the system are activated, i.e. $\lambda_{act} = \lambda_{tot}$. We set an L_{thr} that is deemed acceptable for a system. In Sec. VII, we experiment with different L_{thr} values i.e., 1% and 5%. *SO-WAVES* selects the minimum $\lambda_{act} = \lambda_{min}$ that provides a system performance within L_{thr} of the system performance obtained from $\lambda_{act} = \lambda_{tot}$.

Once we determine the λ_{min} for an application, we account for the TV and PV and activate the best λ_{min} that result in the lowest overall thermal tuning range. Our recent work [18] presents the TV and PV-aware selection of the best λ_{min}.

C. Dynamic Oracle WAVES (DO-WAVES)

A major limitation of *SO-WAVES* stems from the fact that it does not account for the varying trends in bandwidth requirements during the runtime execution of an application. As explained in Sec. VI-A2 and illustrated in Fig. 6, the bandwidth needs is highly dynamic over an application execution. Therefore, selecting a single λ_{min} for the entire application execution leaves much of the power benefits from wavelength selection under-utilized. We exploit this observation to dynamically select the λ_{min} for each interval of an application.

In Dynamic Oracle WAVES (*DO-WAVES*), we use the same L_{thr} from *SO-WAVES* policy to determine the $\lambda_{act} = \lambda_{min}$ at each interval. It is imperative to note that this policy is not realistic as it assumes accurate knowledge of the future execution trends to select the optimal λ_{min} at each interval of an application. Our goal is to design a proactive policy that can closely match the λ_{min} of *DO-WAVES*. We implement *DO-WAVES* for the sake of comparing our proactive policy against the best hypothetical case of operation.

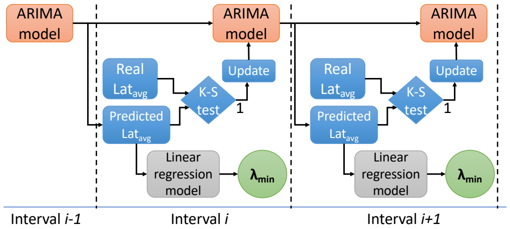


Fig. 7. Flow of *PROWAVES*. Every interval, the ARIMA model forecasts the Lat_{avg}. The linear regression model selects the λ_{min} from the forecasted Lat_{avg}. A K-S test is applied to update the ARIMA model in case of divergence.

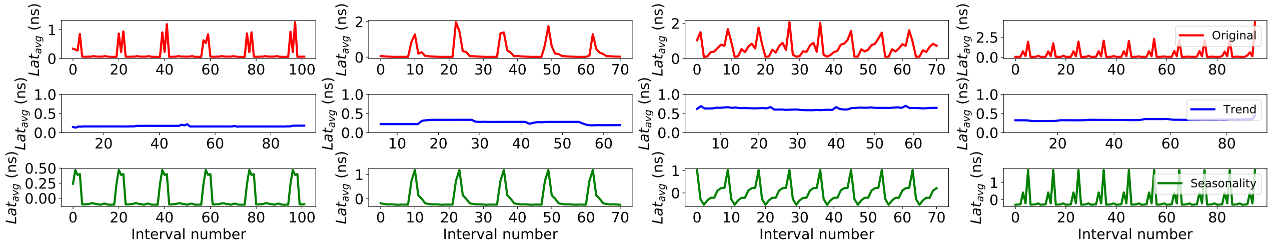
D. Proactive Dynamic WAVES (PROWAVES)

The need for dynamic wavelength selection motivates designing a policy that can proactively select λ_{min} for each interval during an application execution, based on the bandwidth needs. A higher traffic of chiplet-to-chiplet network packets during an application interval will necessitate a higher inter-chiplet bandwidth to ensure minimal queue latency in the FIFO output buffers. The average packet latency (Lat_{avg}), therefore, is a strong indicator of the minimum bandwidth required in order to keep the PNoC out of saturation. Figure 10a shows the plot of $log_{10}(Lat_{avg})$ vs λ_{min} selected by *DO-WAVES*, illustrating linear increase in λ_{min} with log-linear increase in Lat_{avg}. For an application interval i, we define Lat_{avg_i} as follows, where T_{queue_i} is the aggregate queue latency of all packets, and N_{p_i} is the total number of inter-chiplet packets transferred during that interval i,

$$Lat_{avg_i} = \frac{T_{queue_i}}{N_{p_i}}. \tag{12}$$

In order to select λ_{min} in an efficient way, we develop a forecasting methodology to predict Lat_{avg} at each interval. We propose a policy, *PROWAVES*, that can proactively select λ_{min} at each interval based on the forecasted Lat_{avg}. Figure 7 illustrates the flow of our proposed *PROWAVES* policy.

1) ARIMA-based time series forecasting: We utilize an autoregressive integrated moving average (ARIMA) [26] predictor to forecast Lat_{avg} for each application interval by utilizing the past trends in Lat_{avg}. ARIMA models are autocorrelation models in a time series that are extensively utilized in many fields for time series forecasting. ARIMA models typically require the time series to be stationary i.e., the time series should be devoid of trends and/or seasonality. In a time series, trend denotes the increasing or decreasing behavior and seasonality represents a cyclic variation. Figure 8 depicts the trends and seasonality in the Lat_{avg} time series for



(a) *bt* (b) *ep* (c) *shock* (d) *lu*

Fig. 8. Trends and seasonality in the Lat_{avg} time series for applications running 96 threads.

applications. It is evident that the Lat_{avg} time series exhibits minimal variations in trends, but strong seasonality, therefore it is non-stationary. A common strategy to convert such a non-stationary time series to stationary is to compute the difference between consecutive observations, also known as differencing. In ARIMA, an autoregression model (AR) consists of forecasting the variable using a linear combination of past values of the variable. Futhermore, a moving average model (MA) in ARIMA uses past forecast errors in a regression-like model. An $ARIMA(p,d,q)$ model is represented as follows, where p is the order of autoregressive part, d is the degree of first differencing involved and q is the order of moving average:

$$\begin{aligned} Lat'_{avg_i} = & c + \phi_1 \cdot Lat'_{avg_{i-1}} + \cdots + \phi_p \cdot Lat'_{avg_{i-p}} \\ & + \theta_1 \cdot \epsilon_{i-1} + \cdots + \theta_q \cdot \epsilon_{i-q} + \epsilon_i. \end{aligned} \tag{13}$$

Here, Lat'_{avg_i} is the differenced time series of Lat_{avg}, ϵ_k is the forecast error of interval k, and θ and ϕ are the coefficients of moving average and autoregressive parts respectively.

2) Building an ARIMA model: We obtain the best-fitting ARIMA model using the Akaike information criterion (AIC) [50]. The AIC provides an estimate of the relative quality of the statistical models for a given dataset, thereby enabling a comparison of the goodness of fit of the model on the dataset. AIC estimates the relative information lost by the ARIMA model, i.e., the less information the model loses, the higher the quality of that model. The AIC criterion function of $ARIMA(p,d,q)$ model is defined as:

$$AIC = n \ln{(\sigma_g)^2} + 2(p+q+1). \tag{14}$$

We use the starting 30 intervals of an application execution to build the initial ARIMA model. We select the number of intervals empirically based on our application set to build a good-fit ARIMA model and this number can be altered for different applications. We perform a grid search over a range of p, d and q values, starting with an $ARIMA(1,0,0)$ model. We increase the order and use the $ARIMA(p,d,q)$ that receives the lowest AIC value. Figure 9 shows the real and forecasted values of Lat_{avg} for certain applications.

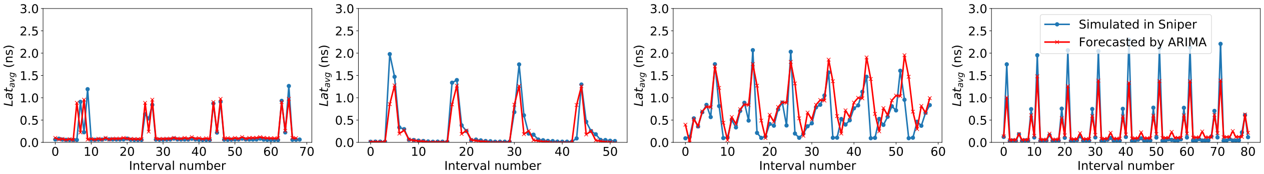
3) Remodeling ARIMA with K-S test: A time-series may exhibit a behavior that is very different than the initial training phase, and this gives rise to inaccurate forecasting. This necessitates the need for remodeling the initial $ARIMA(p,d,q)$ model. We incorporate a goodness-of-fit test to detect the divergence of the real data from the ARIMA predicted data. The Kolmogorov-Smirnov (K-S) [51] test is integrated in the ARIMA model to detect the similarity between the predicted data by the ARIMA model and the real dataset. If the K-S test fails, i.e., marked by 1 in Fig. 7, we conclude that the current ARIMA model is not suitable for time-series forecasting. We then rebuild the ARIMA model, by grid-searching again over the range of p,d and q values, and updating the order of the ARIMA model that results in the lowest AIC value.

4) Selecting λ_{min} from Lat_{avg}: We devise a methodology to correlate the predicted Lat_{avg} from ARIMA model to the λ_{min} selected by *DO-WAVES*. We collect the λ_{min} selected by *DO-WAVES* for different intervals from a training application set, as mentioned in Sec. VI-C. For these same intervals, we collect the Lat_{avg}. We then build a linear regression model by plotting the $log_{10}(Lat_{avg})$ against λ_{min} selected by *DO-WAVES*. Figure 10 shows the scatterplot of $log_{10}(Lat_{avg})$ vs λ_{min} selected by *DO-WAVES*. We fit a line through these points, such that 90% of the points are above this line to ensure that the bandwidth needs at an interval is always satisfied. We use this linear regression model in *PROWAVES* to select the λ_{min} using the forecasted Lat_{avg} from ARIMA model.

E. Thermal remapping of MRRs with wavelength selection

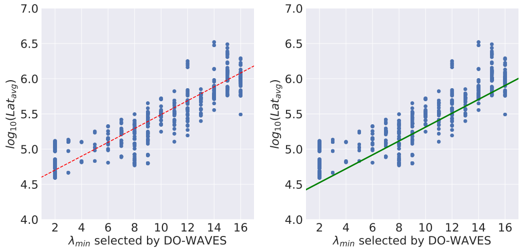
During application execution, when λ_{act} laser wavelengths are activated, there are λ_{act} Tx MRRs and 7 × λ_{act} Rx MRRs in an MRRG that are locked on to the λ_{act} laser wavelengths. The thermal control loop supplies the heating power to maintain these MRRs at resonance. However, there are two factors that shifts the MRR resonance, which then requires a remapping of the MRRs to the laser wavelengths: (1) A large temperature drift introduces a resonance shift greater than FSR/λ_{tot}, (2) λ_{act} is changed during application execution by *PROWAVES*.

Figure 11 shows the thermal remaping of MRRs. When the MRR shift increases over the tuning range of the heaters, the computation is temporarily halted. An on-chip LUT is polled to determine the set of λ_{act} laser wavelengths that result in the lowest thermal tuning power [18]. The thermal control loop then supplies the heating power to lock the MRRs to these new λ_{act} laser wavelengths. This is shown in Fig. 11, where the MRRs resonate at different laser wavelengths after remapping. Similarly, when *PROWAVES* increases or decreases the λ_{act} during application execution, the on-chip LUT is polled to identify the new set of MRRs that needs to be mapped to the selected λ_{act}.



(a) *bt* (b) *ep* (c) *shock* (d) *lu*

Fig. 9. Real and forecasted Lat_{avg} for applications running 96 threads using ARIMA model.



(a) (b)

Fig. 10. Scatterplots of Lat_{avg} vs λ_{min} selected by *DO-WAVES* for $L_{thr} = 5\%$. (a) shows the line with least mean square error, (b) shows the line such that 90% of the points are above the line.

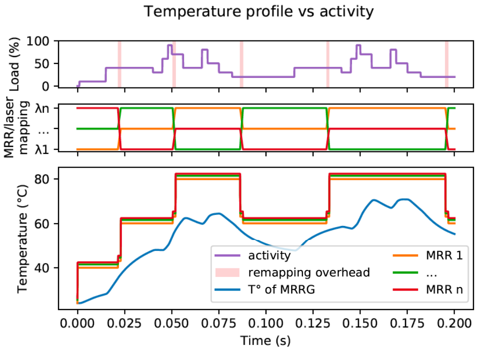


Fig. 11. Thermal remapping of MRRs to λ_{act}. As chip activity varies during execution, the thermal profile of MRRGs varies, causing MRRs within an MRRG to map to different laser wavelengths.

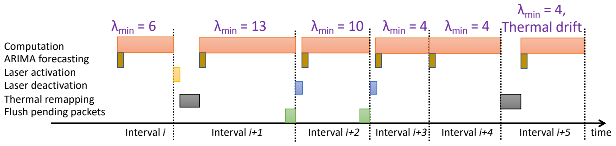


Fig. 12. Latency overhead of *PROWAVES*. Increasing λ_{min} involves laser activation ($2ns$), thermal remapping ($100\mu s$). Decreasing λ_{min} involves laser deactivation ($2ns$) and flushing pending packets ($100ns - 1\mu s$), which are hidden in the computation time.

F. Hardware cost of PROWAVES

Implementation of our proposed *PROWAVES* policy on a 2.5D system comes at a minimal hardware cost. The hardware performance counters are polled every interval to read out PNoC activity statistics, i.e., the number of inter-chiplet packets transferred and overall queue time. An initial ARIMA model is created using these statistics from the training interval and the model parameters (p, d, q) are stored. On average, this ARIMA model is created in $72ms$ for an application. This ARIMA model is utilized to determine the λ_{min} for the next interval in parallel with the execution of the current interval. We observe that ARIMA forecasting takes less than 0.1% of the execution time of an interval and, therefore, is always hidden in the execution time of the current interval.

Once λ_{min} for the next interval is determined, we need to activate the best combination of λ_{min} among a total of $\binom{\lambda_{tot}}{\lambda_{min}}$ combinations. For each combination, we store the heating power of all MRRGs for an operating temperature range in a lookup table (LUT). This LUT holds floating point values of heating power for 8 MRRGs, λ_{tot} wavelengths and temperature range of $300-380K$ ($0.5K$ precision). Since the memory footprint of this LUT is estimated as $400kB$, it can be stored on-chip. At runtime, depending on the thermal profile at the end of an interval, we poll this LUT and exhaustively search across all the laser combinations to determine the best combination of λ_{min}. As the worst-case LUT access time is $\leq \binom{\lambda_{tot}}{\lambda_{tot}/2} \cdot C$ lookups and additions, this latency is hidden within the thermal remapping latency ($100\mu s$).

G. Switching overheads of wavelength selection

The performance overhead of *PROWAVES* come from the latency associated with increasing and decreasing the laser wavelengths during application execution. The ARIMA model predicts λ_{min} for the next interval in parallel with the execution of the current interval. MRR remapping, if required for the next interval, is performed at the end of the current interval. When λ_{min} is increased, the latency comprises of the laser power-on latency and the thermal remapping of the new WDM group of MRRs to the activated laser wavelengths. Simon *et al.* [52] demonstrate that laser switching takes $2ns$ with relatively low drift. Once the laser wavelengths are activated, the thermal control loop remaps the MRRs to the activated laser wavelengths in $100\mu s$ [15]. Therefore, activating additional laser wavelengths during an application execution introduces a latency overhead of $100\mu s$.

When λ_{min} is decreased, the next application interval requires deactivation of certain laser sources ($2ns$ [52]). We observe that there is no additional WDM group of MRRs that needs to be tuned to the new set of laser wavelengths, therefore, the MRR thermal remapping during laser deactivation is not necessary. We simply release the heating power on the MRRs that were communicating via the deactivated laser wavelengths, and maintain the heating power on the remaining MRRs [1]. However, decreasing λ_{min} at runtime requires flushing the pending packets on deactivated laser wavelengths. We measure the worst-case completion of pending packets in the

[1] Note that when deactivating laser wavelengths, we do not perform LUT lookup to select λ_{min}, as the LUT lookup requires MRR thermal mapping to a new set of λ_{min} with a remapping cost of $100\mu s$. So the activated λ_{min} may not be the best combination that result in lowest thermal tuning power.

PNoC to be $100ns - 1\mu s$, and this latency is hidden in the next application interval. Hence, the overall latency of decreasing λ_{min} is negligible.

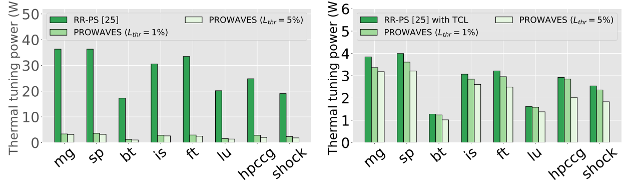
VII. Evaluation results

We evaluate the power benefits of *PROWAVES* against a power scaling technique based on a ridge regression model (*RR-PS*) [25]. With a feature set consisting of network metrics and L1/L2 cache misses, *RR-PS* predicts the number of packets transferred in the PNoC. Using the predicted number of packets, *RR-PS* calculates the minimum number of laser wavelengths that can support the network packets. A major limitation of *RR-PS* comes from the lack of TV and PV modeling, and the resulting thermal tuning power. We evaluate the two major benefits of *PROWAVES*: (1) the modeling of the low-level thermal control loop that enables MRR thermal remapping, and (2) the selection of best λ_{min} for every interval.

We also evaluate the power benefits and performance overhead of wavelength selection against a baseline case that activates all λ_{tot} laser wavelengths ($\lambda_{act} = \lambda_{tot}$). We demonstrate the benefits of *PROWAVES* policy in comparison to a static wavelength selection policy, *SO-WAVES* [18]. We conduct experiments with varying system utilization by running the application with different thread counts. In all experiments, we use L_{thr} values of 1% and 5% to demonstrate the user flexibility of setting the performance loss threshold and exploring the PNoC bandwidth-power tradeoffs.

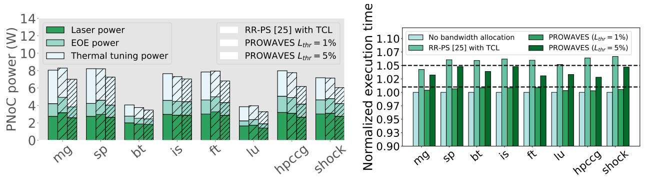
A. Thermal tuning power savings with PROWAVES

Figure 13a shows the thermal tuning power in *POPSTAR* with *PROWAVES* and a TV and PV unaware policy, *RR-PS* [25]. The impact of TV and PV-induced resonance shift is more prominently observed in *RR-PS*. Due to the lack of a control loop for thermal tuning in *RR-PS*, all the MRRs need to be tuned to the designed laser wavelengths. Therefore, the average case tuning range for a random PV distribution across MRRG in *RR-PS* is $FSR/2$. In contrast, the presence of a control loop for thermal tuning in *PROWAVES* enables thermal remapping to the nearest activated laser wavelength, resulting in a worst-case tuning range of FSR/λ_{act}. Since we model the low-level thermal control loop at the system-level, we are able to capture the benefits of thermal remapping and significantly reduce the overall thermal tuning power. Compared to *RR-PS*, *PROWAVES* consumes $24.6W$ and $26.3W$ lower thermal tuning power with an L_{thr} of 1% and 5% respectively. Thus, modeling of the thermal control loop is essential to evaluate system-level power benefits.



(a) (b)

Fig. 13. Thermal tuning power comparison between *RR-PS* and *PROWAVES*. In (a), *RR-PS* does not model thermal control loop that enables thermal remapping, as initially proposed in [25]. In (b), *RR-PS* is updated to include a thermal control loop model as *PROWAVES*, but does not select best λ_{min} that accounts for PV.



(a) (b)

Fig. 14. (a) Breakdown of the overall PNoC power and (b) Normalized system performance with *RR-PS* and *PROWAVES*. Execution times are normalized to a system with no bandwidth allocation ($\lambda_{act} = \lambda_{tot}$). The dotted line indicates the performance loss threshold (L_{thr}).

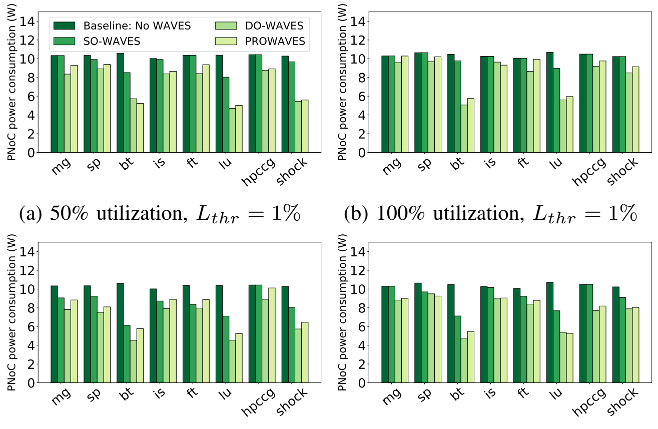
Since thermal control loop is essential for thermal remapping and significantly reduces the thermal tuning power, we incorporate its modeling in *RR-PS* in Figure 13b. This modeling enables us to isolate the specific benefits of wavelength selection. *PROWAVES* accounts for the impact of PV-induced resonant shift which varies across MRRs in an MRRG, and across different MRRGs. *PROWAVES* activates the best combination of laser wavelengths to reduce the impact of PV-induced resonant shifts as opposed to *RR-PS with TCL*, which always selects a fixed set of laser wavelengths for an interval. This finer level of wavelength selection in *PROWAVES* reduces the thermal tuning power by 7.1% and 22.01% for L_{thr} of 1% and 5% respectively, as compared to *RR-PS*.

B. PNoC power and performance comparison

Fig 14a provides a breakdown of the overall PNoC power, consisting of laser, EOE circuitry, and thermal tuning power. Fig 14b compares the system performance of *RR-PS* and *PROWAVES*. The thermal control loop modeling is included with *RR-PS* to evaluate the overall PNoC power. With a tight L_{thr} of 1%, we observe that *PROWAVES* consumes similar PNoC power as compared to *RR-PS*. However, due to the set L_{thr}, the performance of *PROWAVES* is always within the 1% threshold and consistently better than *RR-PS*, which has a performance loss close to or above 5% than baseline. If L_{thr} in *PROWAVES* is relaxed to 5%, we observe average power savings of 29.25% over *RR-PS*. The increased power savings with *PROWAVES* come from the selection of best combination of λ_{min} accounting for PV. The improved system performance in *PROWAVES* shows that ARIMA time-series forecasting predicts the PNoC activity better than ridge regression model. Thus, the flexibility of setting L_{thr} enables the user to trade off performance for higher power savings in *PROWAVES*.

C. Power savings with different WAVES policies

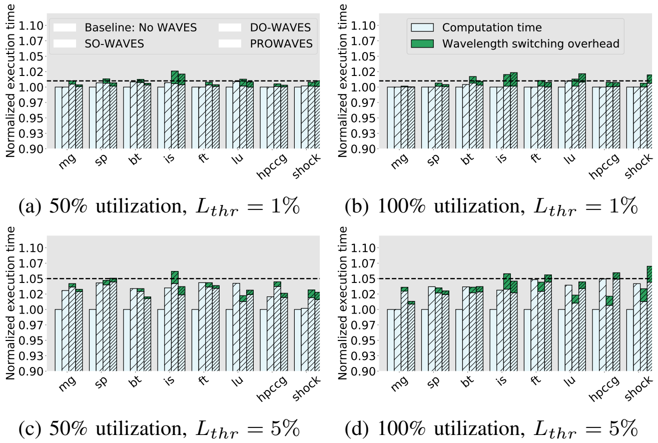
Figure 15 shows the PNoC power consumption with different wavelength selection policies under varying system utilization. *SO-WAVES* [18] consumes 8.6% and 21% lower PNoC power on average than the baseline case for an L_{thr} of 1% and 5% respectively. *DO-WAVES* is able to uncover additional PNoC power savings in the system by activating lower λ_{min} during phases of low bandwidth needs. This is in contrast to *SO-WAVES* that activates a single λ_{min} during the entire application execution. As a result, for L_{thr} of 1% and 5%, *DO-WAVES* provides 34.4% and 40.7% reduction in PNoC power than the baseline. In comparison, for L_{thr} of 1%



(a) 50% utilization, $L_{thr} = 1\%$ (b) 100% utilization, $L_{thr} = 1\%$

(c) 50% utilization, $L_{thr} = 5\%$ (d) 100% utilization, $L_{thr} = 5\%$

Fig. 15. PNoC power consumption of *POPSTAR* with different WAVES policies under varying system utilization and L_{thr}.



(a) 50% utilization, $L_{thr} = 1\%$ (b) 100% utilization, $L_{thr} = 1\%$

(c) 50% utilization, $L_{thr} = 5\%$ (d) 100% utilization, $L_{thr} = 5\%$

Fig. 16. Normalized execution time and wavelength switching overhead with different WAVES policies for varying system utilization and L_{thr}. The dotted line indicates L_{thr}.

and 5%, the system with *PROWAVES* consumes 18% and 33% lower PNoC power than the baseline and 10.2% and 16.4% lower PNoC power than *SO-WAVES* respectively. The power savings with *PROWAVES* is within 12% on average than the theoretical minimum that is achievable with *DO-WAVES*. We analyze this deviation in power savings of *PROWAVES* from *DO-WAVES* due to ARIMA forecasting and linear regression model inaccuracies in Sec. VII-E and VII-F.

We also evaluate the power benefits of wavelength selection policies with different system utilization. The power savings obtained from *PROWAVES* lowers with increasing system utilization. With a higher number of threads per chiplet, there is an increased inter-chiplet network traffic, resulting in higher bandwidth requirements. Consequently, a higher λ_{min} is selected to satisfy the high bandwidth needs.

D. Performance overhead of WAVES

Figure 16 shows the execution time of applications with the baseline case ($\lambda_{act} = \lambda_{tot}$) and under different policies, normalized to the baseline case. For each application, we calculate the wavelength switching overhead of *PROWAVES* by determining the count of thermal remappings arising due to laser activation or a large thermal drift during the execution.

TABLE IV
SUMMARY OF MODELING PARAMETERS AND RESULTS

	RR-PS [25]	SO-WAVES [18]	PROWAVES
Static/dynamic policy	Dynamic	Static	Dynamic
Model	Ridge regression	Offline	ARIMA
Thermal control loop modeling	No	Yes	Yes
Process variation modeling	No	Yes	Yes
Power savings over baseline	13%	8.6%[a] 21%[b]	18%[a] 33%[b]

Latency overhead [a]1% L_{thr}, [b]5% L_{thr}

On average, this switching overhead is computed to be only 0.73% of the overall execution time for *PROWAVES*. Since we calculate the λ_{min} for *PROWAVES* by comparing only the computation time with the performance loss threshold, the overall execution time including the wavelength selection overhead occasionally violates the set L_{thr}.

Compared to the execution time of *SO-WAVES*, the dynamic selection in *PROWAVES* is able to provide better performance at lower PNoC power, leading to much lower PNoC energy compared to *SO-WAVES*. In *PROWAVES*, higher λ_{min} is selected during periods of high bandwidth needs and a lower λ_{min} is selected during periods of lower bandwidth needs. In contrast, since *SO-WAVES* only selects a single λ_{min} throughout the application execution, this λ_{min} is roughly averaged. Therefore, during periods of high bandwidth needs, *SO-WAVES* falls short of selecting the optimal λ_{min}. Similarly, during periods of low bandwidth needs, *SO-WAVES* overestimates and selects a higher λ_{min} than required. Table IV summarizes the results and modeling parameters of *PROWAVES* compared to *RR-PS* [25] and *SO-WAVES* [18].

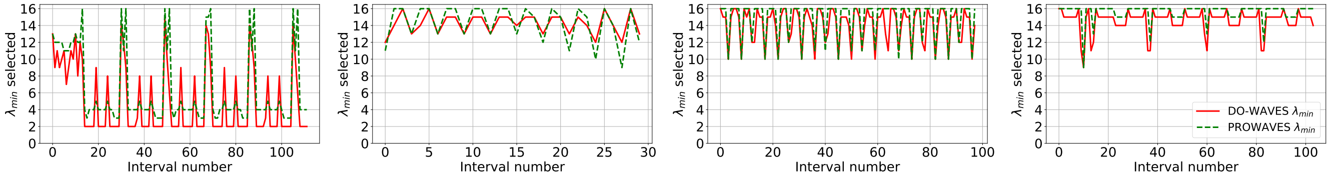
E. ARIMA forecasting accuracy

Figure 17 illustrates the deviation of λ_{min} selected by our proposed *PROWAVES* policy from *DO-WAVES*. This deviation in the selected λ_{min} and the resultant lower power savings in *PROWAVES* can be attributed primarily to two major reasons. First, the Lat_{avg} predicted by the ARIMA model does not have a 100% forecasting accuracy. Second, the linear regression model used to correlate the predicted Lat_{avg} to the *DO-WAVES* λ_{min} has inaccuracies that further contribute to a slightly different λ_{min}.

Figure 9 illustrates the real simulated values of Lat_{avg} on Sniper and the predicted Lat_{avg} values by ARIMA. We calculate the mean squared error of predicted Lat_{avg} as $0.019ns^2$. Thus, our ARIMA predictor with K-S test has an automated process of forming the model with 96.3% accuracy. In addition, as depicted in Fig. 17, the ARIMA model captures the seasonality in the Lat_{avg} time series.

F. Linear regression model accuracy

We analyze the selected λ_{min} against the training data in the linear regression model. We obtain a high R-squared value of 0.916, justifying a good fit of data. Moreover, we observe a low p-value, indicating a strong evidence against the null hypothesis. Specifically, the low p-value demonstrates



(a) *bt* (b) *is* (c) *sp* (d) *mg*

Fig. 17. Comparison of λ_{min} selected by *DO-WAVES* and *PROWAVES* for $L_{thr} = 5\%$. During periods of high bandwidth needs, a higher λ_{min} is activated, and during periods of lower bandwidth needs, a lower λ_{min} is activated.

that changes in the predictor's value (Lat_{avg}) are related to changes in the response (λ_{min}) variable. This shows that Lat_{avg} is statistically significant in predicting λ_{min}.

We observe from Fig. 17 that *PROWAVES* selects a λ_{min} that is at least equal or higher than the λ_{min} selected by *DO-WAVES*. The λ_{min} selected by *PROWAVES* is always higher than the λ_{min} selected by *DO-WAVES* by 0.84 on average with a standard deviation of 1.36. Therefore, at the cost of slightly lower power savings, the performance with *PROWAVES* is always better than *DO-WAVES*.

VIII. Conclusion

PNoCs are promising alternatives as a high-bandwidth, low-latency and improved energy-efficient on-chip communication technology. However, the practical implementation of PNoCs in large manycore systems is limited due to the high PNoC power consumption with increasing number of activated laser wavelengths. Device-level solutions such as low-level control loop to address thermal sensitivity of MRRs seem promising; however, the growing diversity in application behavior demand the need of system-level policies running on top of such device-level solutions. In this paper, we propose a runtime wavelength optimization strategy to address the PNoC power-bandwidth tradeoff in large manycore systems. Our proposed wavelength selection technique, *PROWAVES*, proactively selects and activates the minimum number of laser wavelengths for an application interval based on the bandwidth needs during that interval. We demonstrate the benefits of *PROWAVES* on a 2.5D-integrated PNoC manycore system with a detailed account of TV- and PV-induced wavelength shifts and MRR remapping due to wavelength selection. *PROWAVES* reduces PNoC power by 18% and 33% on average with only 1% and 5% loss in performance, respectively, compared to activating all the laser wavelengths in the system. Furthermore, compared to a state-of-the-art power scaling policy based on ridge regression, a system with *PROWAVES* consumes $26.3W$ lower thermal tuning power due to the modeling of the thermal control loop, which enables thermal remapping.

References

[1] P. P. Gelsinger, "Microprocessors for the new millennium: Challenges, opportunities, and new frontiers," in *International Solid-State Circuits Conference. Digest of Technical Papers, San Francisco, CA, USA*, 2001, pp. 22–25.

[2] D. Stow, I. Akgun, R. Barnes, P. Gu, and Y. Xie, "Cost analysis and cost-driven IP reuse methodology for SoC design based on 2.5 D/3D integration," in *Proc. International Conference on Computer-Aided Design, Austin, TX, USA*, 2016, p. 56.

[3] X. Zhang *et al.*, "Heterogeneous 2.5D integration on through silicon interposer," *Applied Physics Reviews*, vol. 2, no. 2, p. 021308, 2015.

[4] A. Kannan, N. E. Jerger, and G. H. Loh, "Enabling interposer-based disintegration of multi-core processors," in *Proc. International Symposium on Microarchitecture, Waikiki, HI, USA*, 2015, pp. 546–558.

[5] I. Akgun, J. Zhan, Y. Wang, and Y. Xie, "Scalable memory fabric for silicon interposer-based multi-core systems," in *Proc. International Conference on Computer Design, Scottsdale, AZ, USA*, 2016, pp. 33–40.

[6] J. Macri, "AMD's next generation GPU and high bandwidth memory architecture: FURY," in *Proc. Hot Chips Symposium, Cupertino, CA, USA*, 2015, pp. 1–26.

[7] P. Dong *et al.*, "Reconfigurable 100 Gb/s silicon photonic network-on-chip," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 1, pp. A37–A43, 2015.

[8] G. A. Fish and D. K. Sparacin, "Enabling flexible datacenter interconnect networks with WDM silicon photonics," in *Proc. Custom Integrated Circuits Conference, San Jose, CA, USA*, 2014, pp. 1–6.

[9] Z. Wang *et al.*, "CAMON: Low-cost silicon photonic chiplet for many-core processors," *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, 2019.

[10] L. Virot *et al.*, "Germanium avalanche receiver for low power interconnects," *Nature communications*, vol. 5, p. 4957, 2014.

[11] J. Cardenas, C. B. Poitras, J. T. Robinson, K. Preston, L. Chen, and M. Lipson, "Low loss etchless silicon photonic waveguides," *Optics express*, vol. 17, no. 6, pp. 4752–4757, 2009.

[12] M. T. Wade *et al.*, "75% efficient wide bandwidth grating couplers in a 45nm microelectronics CMOS process," in *Proc. Optical Interconnects Conference, San Diego, CA, USA*, 2015, pp. 46–47.

[13] W. Bogaerts *et al.*, "Silicon microring resonators," *Laser & Photonics Reviews*, vol. 6, no. 1, pp. 47–73, 2012.

[14] R. Polster, Y. Thonnart, G. Waltener, J.-L. Gonzalez, and E. Cassan, "Efficiency optimization of silicon photonic links in 65-nm CMOS and 28-nm FDSOI technology nodes," *IEEE Trans. on Very Large Scale Integration Systems*, vol. 24, no. 12, pp. 3450–3459, 2016.

[15] Y. Thonnart *et al.*, "A 10Gb/s Si-photonic transceiver with 150μW 120μs-lock-time digitally supervised analog microring wavelength stabilization for $1Tb/s/mm^2$ die-to-die optical networks," in *Proc. International Solid-State Circuits Conference, San Francisco, CA, USA*, 2018, pp. 350–352.

[16] M. Rakowski *et al.*, "A 4×20Gb/s WDM ring-based hybrid CMOS silicon photonics transceiver," in *Proc. International Solid-State Circuits Conference - Digest of Technical Papers, San Francisco, CA, USA*, 2015, pp. 1–3.

[17] C. Sun *et al.*, "A 45nm CMOS-SOI monolithic photonics platform with bit-statistics-based resonant microring thermal tuning," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 4, pp. 893–907, 2016.

[18] A. Narayan, Y. Thonnart, P. Vivet, C. F. Tortolero, and A. K. Coskun, "WAVES: Wavelength selection for power-efficient 2.5D-integrated photonic NoCs," in *Proc. Design, Automation & Test in Europe Conference & Exhibition. Florence, Italy*, 2019.

[19] C. Chen and A. Joshi, "Runtime management of laser power in silicon-photonic multibus noc architecture," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 19, no. 2, pp. 3 700 713–3 700 713, 2013.

[20] M. Bahadori *et al.*, "Energy-bandwidth design exploration of silicon photonic interconnects in 65nm CMOS," in *Proc. Optical Interconnects Conference, San Diego, CA, USA*, 2016, pp. 2–3.

[21] J. Luo, C. Killian, S. L. Beux, D. Chillet, O. Sentieys, and I. O'connor, "Offline optimization of wavelength allocation and laser power in nanophotonic interconnects," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 14, no. 2, p. 24, 2018.

[22] A. Zulfiqar, P. Koka, H. Schwetman, M. Lipasti, X. Zheng, and A. Krishnamoorthy, "Wavelength stealing: an opportunistic approach to channel sharing in multi-chip photonic interconnects," in *Proc. International Symposium on Microarchitecture, Davis, CA, USA*, 2013, pp. 222–233.

[23] R. Morris, A. K. Kodi, and A. Louri, "Dynamic reconfiguration of 3D photonic networks-on-chip for maximizing performance and improving fault tolerance," in *Proc. International Symposium on Microarchitecture, Vancouver, BC, Canada*, 2012, pp. 282–293.

[24] A. Narayan, Y. Thonnart, P. Vivet, A. Joshi, and A. K. Coskun, "System-level evaluation of chip-scale silicon photonic networks for emerging data-intensive applications," in *Proc. Design, Automation & Test in Europe Conference & Exhibition. Grenoble, France*, 2020.

[25] S. Van Winkle, A. K. Kodi, R. Bunescu, and A. Louri, "Extending the power-efficiency and performance of photonic interconnects for heterogeneous multicores with machine learning," in *Proc. International Symposium on High Performance Computer Architecture, Vienna, Austria*, 2018, pp. 480–491.

[26] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[27] S. Lischke *et al.*, "High bandwidth, high responsivity waveguide-coupled germanium pin photodiode," *Optics express*, vol. 23, no. 21, pp. 27 213–27 220, 2015.

[28] A. V. Krishnamoorthy *et al.*, "Exploiting CMOS manufacturing to reduce tuning requirements for resonant optical devices," *IEEE Photonics Journal*, vol. 3, no. 3, pp. 567–579, 2011.

[29] K. Yu *et al.*, "A 24Gb/s 0.71pJ/b Si-photonic source-synchronous receiver with adaptive equalization and microring wavelength stabilization," in *Proc. International Solid-State Circuits Conference - Digest of Technical Papers, San Francisco, CA, USA*, 2015, pp. 1–3.

[30] Y. Thonnart *et al.*, "POPSTAR: A robust modular optical NoC architecture for chiplet-based 3D integrated systems," in *Proc. Design, Automation & Test in Europe Conference & Exhibition, Grenoble, France*, 2020.

[31] 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 *ACM SIGARCH Computer Architecture News*, vol. 38, no. 3, 2010, pp. 117–128.

[32] Y. Demir, Y. Pan, S. Song, N. Hardavellas, J. Kim, and G. Memik, "Galaxy: A high-performance energy-efficient multi-chip architecture using photonic interconnects," in *Proc. International Conference on Supercomputing, Munich, Germany*, 2014, pp. 303–312.

[33] P. Grani, R. Proietti, V. Akella, and S. B. Yoo, "Design and evaluation of AWGR-based photonic NoC architectures for 2.5 D integrated high performance computing systems," in *Proc. International Symposium on High Performance Computer Architecture, Austin, TX, USA*, 2017, pp. 289–300.

[34] T. Zhang, J. L. Abellán, A. Joshi, and A. K. Coskun, "Thermal management of manycore systems with silicon-photonic networks," in *Proc. Design, Automation & Test in Europe Conference & Exhibition, Dresden, Germany*, 2014, p. 307.

[35] J. L. Abellán *et al.*, "Adaptive tuning of photonic devices in a photonic NoC through dynamic workload allocation," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 5, pp. 801–814, 2017.

[36] Z. Li *et al.*, "Aurora: A cross-layer solution for thermally resilient photonic network-on-chip," *IEEE Trans. on Very Large Scale Integration Systems*, vol. 23, no. 1, pp. 170–183, 2015.

[37] E. Guthmuller *et al.*, "A 29 Gops/Watt 3D-ready 16-core computing fabric with scalable cache coherent architecture using distributed L2 and adaptive L3 caches," in *Proc. European Solid State Circuits Conference, Dresden, Germany*, 2018.

[38] J. Howard *et al.*, "A 48-core IA-32 message-passing processor with DVFS in 45nm CMOS," in *Proc. International Solid-State Circuits Conference, San Francisco, CA, USA*, 2010, pp. 108–109.

[39] T. E. Carlson, W. Heirman, and L. Eeckhout, "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation," in *Proc. International Conference for High Performance Computing, Networking, Storage and Analysis, Seatle, WA, USA*, 2011, p. 52.

[40] S. Li *et al.*, "McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures," in *Proc. International Symposium on Microarchitecture, New York, NY, USA*, 2009, pp. 469–480.

[41] K. Skadron *et al.*, "Temperature-aware microarchitecture," in *Proc. International Symposium on Computer Architecture, San Diego, CA, USA*, 2003, pp. 2–13.

[42] J. Meng, K. Kawakami, and A. K. Coskun, "Optimizing energy efficiency of 3D multicore systems with stacked DRAM under power and thermal constraints," in *Proc. Design Automation Conference, San Francisco, CA, USA*, 2012, pp. 648–655.

[43] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC benchmark suite: Characterization and architectural implications," in *Proc. Parallel architectures and compilation techniques, Toronto, Canada*, 2008.

[44] D. Campbell *et al.*, "Ubiquitous high performance computing: Challenge problems specification," *Georgia Tech. Res. Inst., Atlanta, GA, USA, Tech. Rep. HR0011-10-C-0145*, 2012.

[45] M. Heroux, "HPCCG microapp," 2007.

[46] D. H. Bailey *et al.*, "The NAS parallel benchmarks," *International Journal of Supercomputing Applications*, vol. 5, no. 3, pp. 63–73, 1991.

[47] H. Wong. A comparison of Intel's 32nm and 22nm core i5 CPUs: Power, voltage, temperature, and frequency. [Online]. Available: http://blog.stuffedcow.net/2012/10/intel32nm-22nm-core-i5-comparison/

[48] H. Su, F. Liu, A. Devgan, E. Acar, and S. Nassif, "Full chip leakage estimation considering power supply and temperature variations," in *Proc. International Symposium on Low power electronics and design, Seoul, South Korea*, 2003, pp. 78–83.

[49] J. Parry and L. Wang. A complete guide to 3D chip-package thermal co-design… 10 key considerations. [Online]. Available: http://www.mentor.com

[50] H. Akaike, "Fitting autoregressive models for prediction," *Annals of the institute of Statistical Mathematics*, vol. 21, no. 1, pp. 243–247, 1969.

[51] F. J. Massey Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[52] G. Simon *et al.*, "Experimental demonstration of low cost wavelength drift mitigation for TWDM systems," in *Proc. European Conference on Optical Communication, Dusseldorf, Germany*, 2016, pp. 1–3.

Aditya Narayan received his B.E. in electrical and electronics engineering from BITS-Pilani, India in 2012 and M.S. degree in electrical engineering from University of Pennsylvania, PA, USA in 2014. He is currently pursuing his Ph.D. degree with the Electrical and Computer Engineering Department, Boston University, MA, USA. His research interests include silicon photonic network-on-chips, memory management for heterogeneous memory systems and 2.5D architectures. He is a student member of IEEE.

Yvain Thonnart has been a researcher at CEA-Leti, since 2005. He is now senior expert on communication and synchronization in Systems on Chip, and scientific advisor for the Lab for Silicon Integration of Digital Architectures. His main research interests include asynchronous logic, networks on chip, physical implementation, interposers for large scale systems-on-chip. He is currently leading a project on silicon photonic interposers for optical communication in massively parallel SoCs. He is a member of IEEE and ACM.

Pascal Vivet is Scientific Director of the Architecture, IC Design and Embedded Software Division in CEA-LETI, Grenoble, France. He received his PhD from Grenoble Polytechnical Institute in 2001, designing an asynchronous microprocessor. After 4 years within STMicroelectronics, he joined CEA-Leti in 2003 in digital design lab. His research interests include multicore architecture, Network-on-Chip, energy efficient design, CAD design, and in strong links with 3D integration, Non-Volatile-Memories, photonics. He is a member of IEEE.

Ayse K. Coskun received her M.S. and Ph.D. degrees in Computer Science and Engineering from the University of California, San Diego. She is currently an Associate Professor at the Department of Electrical and Computer Engineering, Boston University, Boston, MA. She was with Sun Microsystems (now Oracle), San Diego, prior to her current position at BU. Her research interests include energy-efficient computing, 3-D stack architectures, and embedded systems and software. She is a senior member of the IEEE and ACM.