Hybrid Silicon-Photonic Network-on-Chip for Future Generations of High-performance Many-core Systems

Achraf Ben Ahmed, Abderazek Ben Abdallah

Hybrid Silicon-Photonic Network-on-Chip for Future Generations of High-performance Many-core Systems

Achraf Ben Ahmed, Abderazek Ben Abdallah

The University of Aizu,

Graduate School of Computer Science and Engineering,

Adaptive Systems Laboratory,

Fukushima-ken, Aizu-Wakamatsu-shi 965-8580, Japan

E-mail: {d8151102, benab}@u-aizu.ac.jp

Abstract

Photonic Networks-on-Chip (PNoCs) promise significant advantages over their electronic counterparts. In particular, they offer a potentially disruptive technology solution with fundamentally low power dissipation that remains independent of capacity while providing ultra-high throughput and minimal access latency.

In conventional hybrid PNoC systems, several electrical control functions, such as path setup, acknowledgment and tear-down are necessary for the end-to-end optical transfer. However, the circuit-switched nature of photonic interconnect directly affects the performance and power characteristics of on-chip communication.

In this paper, we propose an energy-efficient and high-throughput hybrid Silicon-Photonic Network-on-Chip, named PHENIC[1], targeted for future generations of high-performance many-core systems. PHENIC is based on a smart contention-aware path configuration algorithm and an energy-efficient non-blocking optical switch to further exploit the low

---

1

energy proprieties of the PNoC systems. Through detailed simulation, we demonstrate that the proposed system has a better performance and low energy dissipation compared to conventional hybrid PNoCs.

1   Introduction

The continuous increasing demand for higher performance computing systems and aggressive technology scaling has driven the trend of integrating large number of cores in a single chip [3]. Such many-core processors are widely used across many application domains including general-purpose, embedded [18, 7], digital signal processing (DSP) [21, 31, 45], network [25, 49, 11], and graphics [14, 43, 30].

In future generations of high-performance many-core systems, the efficiency of the communication infrastructure is as important as the computation efficiency of individual cores. Conventional electrical networks-on-chip (NoCs) are expected to reach their limits with increasing core counts because of high power dissipation and reduced performance.

Photonic Network-on-Chip (PNoC) [4, 20, 5, 6, 8, 2] is a novel concept enabling ultra-high communication bandwidth in the terabits per second range, low power, and low communication latency. When combined with Wavelength Division Multiplexing (WDM), multiple parallel optical streams of data are concurrently transferred through a single waveguide. This contrasts with the Electronic Network-on-Chips (ENoCs) [12, 13, 15] that require a unique metal wire per bit stream.

The key to saving power in PNoC systems comes from the fact that once a photonic path is established, the optical data is transmitted in an end-to-end fashion without the need for buffering, repeating, or regenerating. This is different from electronic NoCs, where messages are buffered, regenerated and then transmitted on the inter-router links several times en route to their destination. In addition, photonic routers do not need to switch with every bit of the transmitted data like in

electronic routers; optical routers switch on and off once per message, and their energy dissipation does not depend on the bit rate. This feature allows for the transmission of ultra-high bandwidth messages while avoiding the power cost which found in traditional electronic networks.

In conventional hybrid PNoC systems [4, 10, 33, 42, 48, 36, 37, 38, 9], the source node first issues a configuration packet via a copper-based electrical link to the destination node. The configuration packet is routed via an Electric Control Network (ECN), reserving the photonic switches along the path for the photonic message which will follow it in the Photonic Communication Network (PCN). It includes a destination address information, and other additional control information. When the destination node receives the configuration packet, it will acknowledge that the optical path setup is done. When this small electric ACK is received and processed, the source node, then, starts transmission of optical data packets via the waveguides. When the transmission is finished, the reserved path would be released by a release packet. The circuit-switched nature of these hybrid PNoCs directly affects the performance and power characteristics of on-chip communication. As observed in our previously conducted study [20], hybrid PNoCs' energy overhead is mainly caused by the ECN which consumes more than 90% of the total power budget. Moreover, the latency required to execute the different steps involved in the path setting is found to be about three times longer than the photonic data transfer itself.

In this paper, we propose novel energy-efficient and high-throughput many-core hybrid Silicon-Photonic Network-on-Chip architecture (PHENIC). The proposed architecture efficiently reduces the blocking occurrence resulting in reducing the total energy and increasing the system's bandwidth. We demonstrate that the proposed system has a better performance and low energy dissipation compared to conventional hybrid PNoCs. The main contributions of this work are summarized as follows:

- A quantitative and performance study was performed on our earlier photonic NoC [4, 20] upon which the proposed system is based. In this study, we analyze in detail the behavior

of hybrid-PNoC systems and we derive the main limitations contributing to both energy overhead and bandwidth reduction.

- A contention-aware path configuration algorithm that aims to decouple the ECN from the PCN in a manner that they work independently from each other. The proposed algorithm orchestrates the different path setup packets processes; thus, significantly alleviating the contention in the ECN and its consequent energy overhead and further enhancing the bandwidth, contrary to the already proposed hybrid-PNoC architectures.

- A new non-blocking photonic switch capable of handling all acknowledgment signals required for the path setup process (i.e., ACK and Tear-down). Thus, we adopt a new hybrid switching policy in the PCN: *Spatial switching* for the data stream transfer which is mostly used in conventional hybrid-PNoC designs. This is done by manipulating the state of the broadband switching elements. The second switching used is a *Wavelength Selective Switching* for the acknowledgment and Tear-down signals by using passive filters placed at the input and output of each port.

- A detailed performance evaluation where we highlight the efficiency of the proposed system and the performance gain when compared to well known previously proposed hybrid-PNoC systems.

2 Related Work

Many works have been conducted so far to solve the various challenges in PNoC designs in general. *Vantrease et al.* proposed a 3D stacked 256-core fully optical architecture named *Corona* to completely remove all electrical interconnect replacing them by an optical crossbar and token [16]. In a later work [17], they presented channel-based and slot-based protocols for their arbitration mechanism in addition to a flow-control for fully optical interconnects. *Gu*

et al. proposed *FNOC* [24], a fat-tree based fully optical network. They omit the electronic control layer by using an optical turn-around router (OTAR) which carries both payload data and network control data on the same optical network. *Pasricha et al.* [26] proposed using an optical ring waveguide with bus protocol standards to replace global pipelined electrical interconnects. *Beausoleil et al.* [19] proposed a crossbar-based ONoC, where 64 wavelengths are multiplexed over 270 waveguides. 256 waveguides are allocated for control and data, and 14 waveguides are for broadcast and arbitration. *Zhang et al.* [22] introduced a multilayer Nanophtonic interconnection named *MPNOC* which uses multiple layers to create a crossbar with no optical waveguide crossover. A recent work proposed by *Randy et al.* [27] uses also a multi-layer photonic interconnect with a Micro-ring Resonator for the intra layer communication rather than TSVs, as used in [22]. *Kirman et al.* [23] proposed a fully optical ONoC using a wavelength-based oblivious routing, where each node has physical connectivity to all other nodes via static paths. For the wavelength allocation between the nodes, they use a wavelength-reuse algorithm proposed by *Aggrwal et al.* [1]. Some other works focused in how to reduce the crossbar complexity in fully optical architectures. *Pan et al.* proposed *Flexishare* [29] which is a flexible crossbar topology that allows channel provisioning according to the average traffic load and a distributed token stream arbitration which provides multiple tokens for a given channel.

Many research groups also proposed hybrid optical-electronic architectures. These works can be classified in two categories: the first one is circuit-switched based architecture where the electronic network is used for control and the data transmission is performed in the optical layer. The second category is cluster-based where the electronic and the optical networks are used for local and global communications, respectively. Previously in [20], we proposed a hybrid photonic NoC where the electronic layer is based on OASIS router [13] and the photonic switch is based on the work of *Wang et al.* [50]. In this work, a quantitative study was performed and we showed the importance of the optimization of the electronic layer. *Hendry et al.* [32] proposed

a circuit-switched memory access in photonic interconnection networks. This work represents a typical hybrid-PNoC, where all path setup steps are generated and executed in the ECN. *Chan et al.* [34] proposed a circuit switched Electro-Optical NoC for cores-to-memories connections with the addition of a wavelength-selective spatial routing to increase the path diversity and the bandwidth. *Chan et al.* [33] also proposed a circuit switched mesh using a 4x4 non blocking switch augmented with two gateways for ejection/injection from/to the network. An optical crossbar using 56 waveguides also used in this work. *Sacham et al.* [35] proposed a torus hybrid-PNoC based on a blocking 4x4 optical switch with an extra network for the ejection/injection from/to the torus. *Petracca et al.* [48] proposed a non-blocking torus hybrid-PNoC where the conventional path setup scheme is used. *Cisse et al.* [10] proposed a hybrid-PNoC torus named *HPNoC* which uses predictive switching [36] in the ECN to reduce the setup latency by reducing the pipeline stages of the electrical router. Although the latency is reduced by using such predictive switching, the path setup steps are all generated and transmitted in the ECN. *Ye et al.* [37] proposed a new protocol, called *Quickly Acknowledge and Simultaneously Tear-down* (QAST), to reduce the control delays during the path setup and tear-down processes. QAST uses an optical *ACK* signal and sends a *Tear-down* packet at the beginning of a transmission instead of sending it at the end of the transmission, as in conventional hybrid-PNoCs. Optimizing the *Tear-down* to be sent in parallel with the transmission does not really solve the problem of the path setup procedure. Because, the optical transmission of the data is very short and sending the *Tear-down* after, or at the same time, does not really reduce the latency overhead.

In a recent work proposed by *Wang et al.* [38], the typical ECN is reduced to one central controller to process all path setup requests packets and set the corresponding optical switch according to a Microring Resonators (MRs) state table. Although this solution reduces the hop count in the ECN, it suffers from a complex centralized router and the electronic layer cannot be used like a conventional one if we want to use it for small packets (e.g., cache block broadcasting).

Another interesting work to solve the path setup problem was proposed by *Hendry et al.* [40, 41], where they completely remove the ECN and they substitute it by a *Time Division Multiplexing* arbitration scheme which provides round-robin fairness to set up photonic circuit paths. In this work, instead of setting the path in the ECN, each communication between n any pair of nodes is only allowed to be active during a specific time slot. According to the obtained results, the electronic energy did not really decrease. This is because of the buffering required when there is a switching between the X and Y directions. Moreover, the path is fixed in the design level. For cluster-based architectures, *Pan et al.* proposed *Firefly* [28] which reduces the crossbar complexity by designing smaller optical crossbars connecting selected clusters and implementing electrical interconnect within the cluster. Another recent work was proposed by *Tan et al.* [39] where a butterfly fat-tree based hybrid optoelectronic NoC architecture is introduced using the generic wavelength-routed optical router. However, the wavelength assignment used in this approach for routing purposes leads to an inefficient use of the optical spectrum, as we previously explained. Figure 1 summarizes all the previously stated works categorized according to their types and main key architecture.

To the best of our knowledge, none of the existing solutions proposed so far take advantage of circuit-switching benefits by using the entire optical spectrum through WDM for end-to-end communication and combines them with a contention-free path setup algorithm which: first, it eliminates the association between the ECN and the PCN which is considered as a direct source of the latency overhead. Second, it provides a better use of the ECN resources by minimizing the number of *Path_blocked* packets generated in the network due to resources limitations. In the next section, we analyze the different limitations of conventional hybrid-PNoC systems.

3   Performance Study of Photonics NoC Systems

Previously in [20], we performed a preliminary evaluation where we analyzed the performance of our earlier proposed photonic NoC [4]. The evaluation is performed under different message sizes, network sizes, and with different synthetic traffic patterns. We also compared the results obtained with a conventional ENoC system [15, 46, 47, 3]. The power consumption near-saturation for different network sizes and different benchmarks, is evaluated. The first thing to notice from this figure is the efficiency of PHENIC system in terms of power consumption when compared to conventional ENoC. As we explained earlier, this efficiency is inherited from the low-power properties of hybrid-PNoC systems.

The second observation, is the large gap which manifests when we increase the network size from 64 to 256 cores. Despite the fact that this power overhead is still less than that of ENoC, it puts under question the scalability of hybrid-PNoC systems as we increase the network size. This is because these systems are targeted for Many-core systems which can reach the hundreds and thousands of cores [3].

To understand the reasons for this increase in power, and which of the ECN and PCN is mainly responsible, we analyzed the power and latency overheads of 256 cores PHENIC system under different message sizes. We find out that the ECN consumes the largest portion of the system power budget. This portion varies between 78% and 93% of the system total power, depending on the message size. At the same time, the setup latency is more than 3x greater than the transmission latency.

From these results, it becomes obvious that the ECN should be given more attention and further optimized in order to reduce the power overhead, and also to increase the throughput in a given hybrid-PNoC system. For this purpose, we evaluated the average dynamic energy in the input-buffer which is considered as the most congested and power-hungry component in the ECN. With

this analysis, we can have a clear understanding about the effects of the different packets traveling the ECN since the input-buffer activity reflects the other components behavior, such as the routing computation module, arbiter, crossbar, and inter-router links.

When evaluating the input-buffer dynamic energy, we observed that two portions dominate the total dynamic energy in the input-buffer which are the *Path-setup-Control-Packet* (PSCP) and *Path_blocked* packets (85%), while the *ACK* and *Tear-down* packets consume much smaller ration (15%). As we previously mentioned, for every communication between a (source, destination) pair, a *PSCP* packet is injected and travels the ECN in order to request the necessary resources in the PCN. When these resources are utilized by another communication, a *Path_blocked* packet is generated and travels back to the source node while releasing the already reserved resources by the *PSCP* packet. The alternation between the PSCP and *Path_blocked* packets continues several times until the requested resources are released and become available. In this fashion, a significant amount of energy is wasted on generating, processing, and storing ineffective packets that do not reach their destinations after all. The energy burden of these two packets is quite high, and we can also observe that they are quite equivalent. This is logic since as the number of *Path_blocked* packets increases, the number of *PSCP* packets necessary to establish again the path increases as well. In an ideal situation, the *Path_blocked* energy overhead should be removed and the PSCP one should be a little bit higher than those of the ACK and *Tear-down* overheads. In practice, it is very difficult; thus, the most effective approach is to reduce the blocking occurrence as much as possible in order to reduce the *PSCP* packets generation and alleviate the consequent congestion.

In conventional hybrid-PNoC systems, blocking is mainly caused by two major factors: first, the use of a blocking optical switch where some input- or output-ports share the same resources (MRs and waveguides). This kind of switch has been used in many prior works in order to decrease the energy (both static and dynamic) in the PCN by reducing the number of MRs and waveguides. Figure 2 shows four kinds of switches where Figs. 2 (a) and (c) represent two blocking switches

[20, 33, 50] while Figs. 2 (b) and (d) depict two non-blocking ones [33, 48]. From these figures, we can see the difference between the two kinds of switches in terms of complexity. As a matter of fact, blocking switches are simple with limited number of MRs and waveguides, while non-blocking switches are much more complex. Nevertheless,the ECN energy and latency significantly increase when using a blocking type, as we previously proved. This is due to the big difference between the energy properties of each of the photonic and electrical paradigms. As a consequence, hybrid-PNoCs should be equipped with non-blocking optical switches that allow the elimination of the dependency caused by the resources' sharing between communications.

The second factor for blocking is the high congestion frequently found in the ECN. As we previously mentioned, the ECN in most hybrid-PNoCs host different kinds of packets (e.g., *PSCP, Path_blocked, ACK*, and *Tear-down*). These packets share all the resources of the ECN, creating a congestion that has a huge impact on the energy, as well as on the system bandwidth. To relieve this congestion, the naive approach is to increase the buffer size; however, this solution increases the static power consumption and also the ECN area. Allowing part of these packets to be transferred in the PCN, provides a better traffic balance and fair resource utilization. In particular, by sending the *ACK* and *Tear-down* packets in the PCN as optical signals, three main advantages can be achieved: (1) the congestion in the ECN is significantly relieved and as a result the blocking probability decreases as well. (2) When transferring these two types of packets in the PCN, we can exploit the benefits of the latter's low-energy properties. Moreover, the ECN buffer size can be reduced without affecting the performance. (3) Break the dependency between the different path setup steps which can increase the blocking probability.

To understand this latter point, Fig. 3 illustrates a simplified example of three cases of dependency frequently observed during our evaluation study between the *PSCP* and *Tear-down* packets. In the first case, a *PSCP* of a given communication (C3) is stored in the west input-port and requesting the east output-port; however, the requested resources for both west input-port

and east output-port are utilized by a former communication (C1). Despite the fact that the *Tear-down* packet which will release these resources in the next cycle, the *PSCP* is dropped and a *Path_blocked* packet is generated to travel back to the source node (represented by a green dashed line in Fig. 3) where a new *PSCP* is generated. Similarly in the second case, a PSCP in the south input-port (C6) is requesting the north output-port. In this case, the PSCP does not share the same output-port with the previous communication (C4). Nevertheless, it is blocked since the input-port resources are already reserved and will be released in the next cycle by the *Tear-down* packet located in the same input-port. In the third case, the *PSCP* (C5) and the *Tear-down* (C2) packets are located in different input-ports and requesting the same local output-port. We assume that for arbitration reasons the *PSCP* is served first; therefore, it is blocked despite the fact that the local output-port resources will be released in the next cycle. This case is considered to be the worst. This is because the *PSCP* is already in the destination node. Nevertheless, it is blocked and has to travel all the way back to the source node due to its dependency with the *Tear-down* packet.

As a conclusion for this study, blocking constitutes the major source of energy and latency overhead in conventional hybrid-PNoC systems. It is mainly caused by congestion, which mostly occurs due to the different types of packets sharing the ECN resources. In order to solve the problems elaborated in this study, we explain the details of the complete architecture of the proposed PHENIC-II system in the next section. We first highlight the key functions and components of the energy-efficient non-blocking switch, and then the adopted path setup algorithm targeted to alleviate the contention commonly found in conventional hybrid-PNoC systems.

4   Proposed System Architecture

The simplified block diagram of the PHENIC system is shown in Fig.4. The system consists of two networks: the first one is the PCN, and is based on silicon broadband photonic switches interconnected by waveguides; the second one is the ECN and is used for path reservation

and configuration of the optical switches at the PCN by mainly powering *ON/OFF* the MRs. Each Processing Element (PE) is connected to a local electrical router and also connected to the corresponding gateway (modulator/detector) in the PCN. Messages generated by the PEs are separated into control signals and payload signals. Control signals are routed in the ECN and used for path setting (routing). The payload signals are converted to optical data and transmitted on the PCN. In the next subsections, we explain the photonic switch, electronic router, and the adopted path setup algorithm.

4.1 Non-blocking Photonic Switch

As we mentioned earlier, our proposed switch should be able to handle the data stream like any other conventional photonic switch, as well as the *ACK* signals and the resulting regeneration process of the *Tear-down* signal at each hope. Thus, we adopt a hybrid switching policy: *Spacial-switching* for the data signals by manipulating the state of the broadband switching elements (green MRs in Fig. 5) and a *Wavelength-selective switching* for the *Tear-down* signals by using detectors and modulators. Moreover, since the *Tear-down* signals should be checked and regenerated at each hop, it is crucial that their manipulation should be done automatically and without interfering with data signals nor causing a blockage inside the switch.

It is important to mention that we did not add a dedicated gateway including detector and modulator banks for the *Tear-down* signal at the local port. Instead, when the *Tear-down* is generated at the source Network Interface, it is first sent to the electronic router. There, the *Photonic Switch Controller*, explained later in Fig.6, will release the corresponding MRs and generate another *Tear-down* which is sent to the output-port modulator in the PCN where it continues its path in a hop-by-hop basis until it reaches its destination. At the destination node, the *Tear-down* is detected in the input-port and sent to the *Photonic Switch Controller* in the corresponding electronic router. In this fashion, we can omit the overhead of an additional gateway

which becomes significant when we increase the number of cores.

Table 1 shows the MRs configuration for data transmission, where 18 MRs are used in a non-blocking fashion. We use the first six wavelengths in the optical spectrum starting from 1550 nm, with a wavelength spacing equal to 0.8 nm to maintain a low cross-talk as reported in [55]. For the acknowledgment signals, we use the first five wavelengths in the optical spectrum starting from 1550 nm: four wavelengths for the *Tear-down* signal where each one is dedicated for each port except the local one. In addition, a single wavelength for the *ACK*. The remaining available wavelengths are used for data transmission. Moreover, the five wavelengths used to control the *ACK* and *Tear-down* signals are constant regardless of the network size, in contrast with the fully optical where the number of wavelength used for control and arbitration grows with the network size. Thus, cutting these wavelengths from the available spectrum to be used for control, would not degrade the system bandwidth. These five wavelengths will be negligible especially when Dense Wavelength Division Multiplexing (DWDM) is used providing up to 128 wavelengths per waveguide [52]. The wavelength assignment for each port is shown in Table 2.

In case where the *Tear-down* signals enter the switch, they need to be redirected to the corresponding electronic router. Since these signals are coming from different ports, and they are modulated with different wavelengths, detectors capable of switching all the four wavelengths are placed in front of the input-ports to intercept them. The converted optical signal will be redirected to the electronic router to be processed. According to the information included, the corresponding MRs will be released. For the *ACK*, when the PSCP reaches the destination, 1-bit optical signal is modulated starting from the output port (i.e., opposite direction) and travels back to the source.

With this smart hybrid switching mechanism, we take advantage of the low-power consumption of the optical link by using optical pulses modulated with the adequate wavelength instead of propagating the acknowledgment signals in the ECN. Second, we take advantage of the WDM proprieties by separating the acknowledgment packets and the data signals and let them coexist

in the same medium without interfering with each other. This is in contrast with the electronic domain where these acknowledgment packets travel for a several hops consequently blocking (preventing) the waiting cores from sending their PSCP packets.

Although the insertion loss and cross-talk performance study are out of the scope of this work, it is important to mention that by using such scheme to handle *ACK* and *Tear-down* signals no additional insertion loss will be added. In fact, since the *Tear-down* signal will be generated at each hop, the incurred insertion loss will be much lower than the worst case insertion loss. For the *ACK* signal, the insertion loss will be the same as the corresponding data signal loss. Equation 1 [51] shows how the laser power is calculated in the PCN. Where $P_{sense,i}$ is the laser power required at a given photo-detector i and $loss_i$ is the loss of that photo-detector, given in dB.

$$P_{Laser} \quad = \quad \sum P_{sense,i} . 10^{loss_i/10} \tag{1}$$

The energy power overhead is caused by the modulators and detectors placed in the front of each port, which as we show later in the evaluation section, is much lower than the one caused if acknowledgment signals were transferred in conventional electrical links. In addition, since modulators and detectors are energy bit-dependent, we only use 1-bit to modulate the *ACK* signal and 8-bits for the *Tear-down* to modulate any destination address in a 256 cores system.

4.2 Light-weight electronic router

In the proposed PHENIC system, the ECN is based on Mesh topology. The packets are forwarded along the network using *Wormhole-like* switching policy and then routed according to *Dimension-Ordered-Routing* (DOR-XY). As a flow control the ECN adopts Stall-Go mechanism, and Matrix-Arbiter as a scheduling technique. The router is considered as the backbone element in the whole ECN. The ECN router architecture is based upon OASIS-NoC router (ONoC Router) [15, 46, 47, 3]. Figure 6 illustrates the ECN router architecture where the routing process at each router can

be defined by three main pipeline stages: *Buffer writing* (BW), *Routing Calculation and Switch Allocation* (RC/SA), and finally the *Crossbar Traversal* (CT).

As shown in the Fig. 6, the electronic arbiter receives the detected *Tear-down* from the above switch (colored arrows). According to the information encoded in this signal, the corresponding MRs are released and a new *Tear-down* is generated for the next hop until it reaches its final destination and all MRs involved in this communication are released. The figure shows also the connection between the network interface (NI) and the local port, where a configuration packet (CP) is sent from the NI to the local port. The CP could be a setup packet or a path blocked packet. The NI is connected also to the data switch (i.e., PCN). When the source node receives the ACK, the payload is processed by a serializer bank (if needed), a high speed driver, and a modulator to convert the electrical signal to an optical one. At the source node, the optical data leaves the data switch and go through a detection step, a high speed Trans-Impedance-Amplification step, and a deserilization step. At the end the NI's receiver, receives the payload data with its original clock speed.

4.3 Contention-free Path Configuration Algorithm

After introducing the photonic switch and electronic router, we dedicate this subsection to explain the proposed path-setup algorithm and explain its ability to remove the dependency between the ECN and PCN which is causing a significant latency overhead in conventional hybrid-PNoC systems. In addition, we considerably decrease the latency caused by the path blocking that requires several cycles for the path dropping and the new PSCP generation. Another key feature of the proposed path setup algorithm is the efficiency of the ECN resources' utilization. By moving the acknowledgment signals to the upper layer, we can reduce the buffer depth to only 2 slots, since half of the network traffic is eliminated. This reduction is a key factor to design a lightweight router, highly optimized for latency and energy.

Figure 8 (a) shows an example of a successful path-setup process where all the necessary resources between a given source-destination pair are reserved. Before optical data transmission, the source node issues a $Path-setup-Control-Packet$ ($PSCP$) which is routed in the ECN and includes information about the destination and source addresses. In addition to the source and destination addresses, other information are included. For example, 1-bit is used for the Packet-type field. This field can be "0" for a PSCP (Path-Setup-Control-Packet) and "1" when this configuration packet is a Path-blocked. Other information to ensuring Quality-of-Service and fault-tolerance, such as Message-ID, Fault-status, Error-Detection-Code, can be also included. For each electrical router, the output-port is calculated according to Dimension-Order routing [15]. Every time the PSCP progresses to the next router, the optical waveguides between the previous and current routers are reserved. Depending on the output port of the electrical router, the corresponding photonic router is configured by switching ON/OFF one or more MRs using the MRs configuration table shown in Table 1. In the example shown in Fig. 8 (a), the packet is entering the local input-port attached to the Network Interface (NI) and requesting the east output-port. According to Table 1, MRs 12 and 17 are required and their availability is checked in the (Micro Ring State Table) MRST. In this table, both MRs' states are "0" (free). Therefore, the switch controller reserves these two MRs and changes their states from "0" (free) to "1" (not free). After this successful reservation (hop based), the PSCP continues its path to the next hop and the same procedure is repeated until all necessary MRs are reserved for the complete path. This process is illustrated in $lines$ $1-10$ of Algorithm 1. In case where the requested MRs at a given optical switch along the path are not available, blocking occurs. This can be seen in Fig. 8 (b) where MR 16, which is necessary for the ejection to the local output-port from the west input-port, is used by another communication. In this case, the $PSCP$ is converted into a $Path\_blocked$ packet (PB). The PB, then, travels back to the source node and releases the already reserved resources. The release is done by re-updating the corresponding entries in the MRST to "0" and by sending

an electrical "OFF" signal to the corresponding MRs in the PCN. This process is illustrated in *lines* $11 - 15$ of Algorithm 1. When the *PSCP* arrives successfully at the destination node, the NI modulates one-bit acknowledgment (ACK) signal to travel back to the source via the PCN. This can be seen in Fig. 8 (c) and in lines $1 - 5$ of Algorithm 2. Upon the arrival of this *ACK* signal, the source node modulates the payload through the data modulators and sends it to the destination node via the PCN. Lines $6 - 10$ of Algorithm 2 depicts this data/payload transfer phase. The last process of the proposed path-configuration algorithm is the $Tear - down$ step as shown in lines $11 - 16$ of Algorithm 2. When the entire payload is transmitted, it is necessary to release the reserved optical resources. This is handled by the source node which sends a $Tear - down$ packet to the destination after predetermined number of cycles depending on the source-destination addresses, transmission bandwidth and message size. As shown in Fig. 8 (d), the source's NI sends the electronic $Tear - down$ packet (TD) to the first electronic router ER_1. The Electronic Controller (EC) in this router indexes the MRCT with input-output ports information and determines the MRs that need to be released. As we can see in this figure, the states of MRs 12 and 17, previously reserved in the path-setup process, are reset to *Free* (state="0") and electrical "OFF" signals are sent to these two MRs.

After the MRs are deactivated, a new optical Tear-down signal is generated according to the used wavelength. It is sent through the PCN to the next hop where it is converted back to electrical and redirected to the EC in the corresponding electronic router to be processed. After this process, the MRs are released and a new optical Tear-down signal is generated. This process is repeated until the *Tear-down* reaches the destination and all optical resources are released. It is important to mention that the path-setup and path-blocked processes of the proposed algorithm are very similar to the conventional ones [4, 20, 10, 32, 33, 35]. The main difference is that the MRST in our proposal contains only two states: *Free* and *Active*. The MRs are set "ON" as soon as the PSCP succeeds to reserve them. In the conventional mechanisms, three states are necessary:

Free, *Reserved*, and *Active*. When the PSCP finds the requested MRs *Free*, it updates their states in the MSCT to *Reserved* without turning them "ON". When the complete path-setup process is completed, the ACK signal travels back to the source node and sets the corresponding MRs "ON" by updating their states in the MSCT to *Active*. With the proposed algorithm, some portions of the reserved path might be set "ON" and then "OFF" due to the unavailability of the resources. However, it enables the fast ACK transmission in the PCN.

In conventional path-setup algorithms, the ACK and Tear-down packets are transmitted in the ECN and have to go through all the buffering, routing computation, and arbitration stages. With the proposed algorithm, they are carried via the PCN. As a consequence, the ETE latency can be significantly reduced in addition to the dynamic energy saving that can be achieved.

5 Evaluation

5.1 Evaluation Methodology

We simulate our proposed PHENIC system using a modified version of PhoenixSim which is a physical-layer simulator developed in the OMNeT++ simulation environment [44]. The used simulator incorporates detailed physical models of basic photonic building blocks such as waveguides, modulators, photodetectors, and switches. Electronic energy performance is based on the ORION simulator [53]. We evaluate the bandwidth performance and energy consumption for 64 and 256 cores systems. We compare the obtained results with the previous blocking mesh-based PHENIC system (PHENIC_BL) [20] and three conventional hybrid-PNoC architectures [33, 44, 42]. We chose these three networks for their different behaviors. In fact, the first one has a blocking switch (Chan_Mesh) and was proposed by Chan et al. [33]. The second one is considered as non-blocking (Chan_Xb), since it uses a crossbar [44]. The third system is a torus-based system (Shacham) [42] having the capability of setting the path with less hop count

by taking advantage of the connections between the edges. For benchmarks, we used *Random Uniform* and *Bitreverse* traffic patterns. *Random Uniform* traffic is a communication pattern where the destinations are randomly and uniformly selected each time a new communication occurs. In *Bitreverse*, each node sends messages to the complement node of its ID; thus, resulting in very long communications to observe the scalability of the proposed system. Tables 3 and 4 show the system and energy configuration parameters, respectively.

5.2 PHENIC System Performance Evaluation

5.2.1 Complexity

In this section we evaluate the complexity of the proposed system against the four other architectures. The evaluation considers the number of used rings and the resulting static thermal tuning. The number of used MR is given by equation 2, where $Mod/Detc_{(ring)}$ is the number of rings required to modulate/detect the payload signal. $Switch_{(ring)}$ is the number of ring required for the photonic switch to route the optical data. Finally, the $ACKs(ring)$ is the number required to handle the acknowledgment signal.

$$Total_{(ring)} = Mod/Detc_{(ring)} + Switch_{(ring)} + ACKs(ring) \tag{2}$$

Tables 5 and 6 show the comparison results for 64 and 256 cores system, respectively. We can see that the two blocking networks (PHENIC_BL and Chan_Mesh) have the lowest number of rings. In fact, this kind of network is used for light-traffic load, where the injection rate is low and the use of blocking switch does not degrade the performance. In addition, with minimal number of rings, the resulting insertion loss is lower than the non-blocking one. But, when it comes to the system performance, this kind of network shows higher energy and the number of blocked requests increases considerably, as shown in the next section. For the proposed PHENIC system,

it has an additional rings used for acknowledgment signal, compared to the other networks. This increase can reach 100%, 50% and 12% when compared to the blocking networks, crossbar and torus systems, receptively. We also observe the same behavior when evaluating the required static thermal tuning, which is required to maintain the functionality of the ring, under 20K temperature with 1μW for each ring.

5.2.2 Latency and Bandwidth Evaluation

Figures 9 (a) and (b) show the overall average latency and the average latency near the saturation region, respectively. We can see that for zero-load latency, all networks behave in the same way. Near saturation, PHENIC shows more flexibility and scalability in 256 cores when compared to the other networks. For the 64 cores configuration, the crossbar-based system slightly outperforms PHENIC system in terms of latency. This can be explained by the use of Optical-to-Electronic conversion of the *Teardown* which affects the overall latency for small networks.

For the achieved bandwidth, Fig. 10 shows that the bandwidth is increased by 24% and 51% when compared to PHENIC_BL and Chan_Mesh, respectively, for both 64 and 256 cores configurations. When compared to the crossbar and the torus systems, we can see that the three systems behave in the same way. While the torus system has the capability of setting the path with less hop count, we can see that PHENIC system can achieve the same performance without the need for an extra accessing network which is required for the torus. This behavior is observed for both 64 and 256 core systems.

From Fig. 11 we can see the resulting blocking latency comparison results for all studied networks in 64 and 256 cores systems under random uniform traffic. The blocking latency can be defined as the average time added to the overall latency when a path setup packet is being blocked and needs to go back to the source node. The first thing to notice from Fig. 11 is the resulting overhead of using a blocking switch to save on the number of rings. We can

see how the previous blocking PHENIC_BL and the Chan_Mesh networks have a considerable blocking latency, reaching the 200% when compared to the proposed PHENIC, crossbar, and the torus systems, in both 64 and 256 cores systems. When comparing the proposed PHENIC to the crossbar-based and the torus-based systems, we can see that the proposed PHENIC slightly outperforms the two networks in 64 cores system. When it comes to larger networks, we can clearly see the benefits of the proposed PHENIC system. For example, when compared with the crossbar-based system, which is considered as non blocking and also having the same number of rings (except for those used for the acknowledgment signals), we can see an improvement of 60% just before the saturation. We can conclude that by breaking the dependency of the different configuration packets, many requests can be saved from being blocked. This improvement is less when compared to the torus-based system with just 37%. This is because it has the capability of using the edges, so a path blocked packet spends less time to reach the source node. Another interesting behavior is the one of the curve in the proposed system is less aggressive then the other networks. We can see, for instance, that between 0.06 ms and 0.04 ms injection rates (near-saturation region), the blocking latency for the crossbar-based system increased by 300%, while it is just 63% for the PHENIC system. We can say that the proposed system is less sensitive to the blocking when compared to a blocking (PHENIC_BL, Chan_Mesh) and non-blocking networks (Chan_Xb, Shacham).

Our final evaluation in this subsection is shown in Fig. 12, which shows the number of blocked requests that reached more than half of the network diameter. In other words, the number of PSCPs that failed to reach their destinations after traveling more than half of their path. We can see that for low injection rates, all networks behave similarly. When the injection rate increases and the system reaches the near-saturation region (between the two vertical dashed lines) we can see that in the proposed PHENIC system, the number of blocked requests decreases by 31% and 36% when compared to the crossbar and the torus based systems for 256 cores, respectively. Compared to the

blocking networks, the number of blocked requests for PHENIC 256 cores decreases by 42%, and by 35% for 64 cores system. Moreover, the curves have the same behavior as the blocking latency in Fig. 11. We can notice that the curve for the other networks is more aggressive, in contrast with the proposed system. In this figure, we are only showing the most energy-costly portion (i.e., packet blocked). Since a PSCP traveling more than half the network and after that it is canceled, this incurs high wasted energy dissipation (i.e., buffering, switching, crossbar traversal).

5.2.3 Energy Evaluation

We evaluate the energy overhead for the PSCP which is given by Equation 3, where PS_{Succ} is the dynamic energy in the ECN dissipated by the successful *PSCPs* reaching their destinations, and PS_{Failed} is the dynamic energy consumed by the *PSCPs* which resulted in *Path_blocked* packets. We also evaluate the *ACK* energy overhead which is defined as: (1) the energy dissipated by the *ACK* and *Tear-down* packets for the PHENIC_BL system, and (2) the sum of the dynamic energy of the modulators and detectors used for the optical *ACK* and *Tear-down* signals in PHENIC system. These two definitions are represented by equations 4 and 5, respectively.

$$PSCP_{Energy} = PSCP_{Succ} + PSCP_{Failed} \tag{3}$$

$$E - ACKs_{Energy} \quad = \quad Ack_{Packet} + Teardown_{Packet} \tag{4}$$

$$O - ACKs_{Energy} \quad = \quad ACKs_{Modulators} + ACKs_{Detectors} \tag{5}$$

Figures 16 (a) and (c) show the PSCP and ACKs dynamic energy overhead for half-load traffic under random uniform and bitreverse benchmarks. As can be seen in these two figures, the energy overhead of the PSCP considerably decreases by almost 66% for both 256 and 64 cores systems, when compared to the blocking networks. The same enhancement can also be seen for the ACKs

energy which is also considerably reduced by 36% in 256 cores and 64% in 64 cores systems. When compared to the crossbar-based system, this latter outperforms the proposed PHENIC in both 64 and 256 cores systems. Nevertheless, PHENIC system is still showing better performance when compared to the torus-based system.

Figures 16 (b) and (d) represent the energy overhead when the system is fully loaded (i.e., near the saturation region) for random uniform and bitreverse traffic, respectively. We can notice that the decrease in the PSCP and ACK energy is considerable when compared to the other architectures for both small and large networks, especially when compared to the blocking ones. Moreover, the torus based system is largely penalized due to the additional ports for the connection between the edges.

We can also see, that for all network the PSCP energy dominates the overall energy, this is because the blocking can be avoided to a certain limit; but, due to the photonic resources limitation, some of the requests become blocked. This problem is mostly related to the structure of the switch and can be avoided by using high-radix switches in addition to be related to the used routing algorithm. For the acknowledgment's energy, it is clear that the optical handling of the *Tear-down* and *ACK* adopted in PHENIC is more energy efficient for the two benchmarks and for the two network sizes.

This can be clearly observed when we compare the total energy and the energy efficiency. While the proposed PHENIC, crossbar-based, torus-based systems behave in the same way in terms of bandwidth, they have different energy profiles.

Figure 15 shows the total energy and the energy efficiency comparison results for 64 and 256 cores systems. For the 256 cores configuration, the proposed system outperforms all other networks. This is illustrated by an improvement in terms of energy efficiency reaching 26% and 48% when compared the crossbar-based (non blocking) and the mesh-based (blocking), respectively. When compared to the torus-based architecture, PHENIC improves the energy

efficiency by up 70%. The torus-based architecture offers high bandwidth thanks to the connection between edges leading to short communications. On the other hand, it comes at high energy cost. This can be explained by the fact that the additional input-ports, required for the edge connections established in the torus-based system, incur increased area and consequently an energy overhead.

In Fig. 16 (a) and (b), the energy breakdown is shown for 64 and 256 cores systems, respectively. Compared to other networks where the electronic energy is reaching 90% of the total energy, PHENIC shows more balanced energy distribution between the photonic and electronic networks. This is despite the fact that the electronic power is still high with 70% of the total system energy. When we dig more in the energy evaluation, we find the explanation of this energy efficient scheme. Figures 17 (a) and (b) show the buffering dynamic energy comparison results. From this figure, we can see first how the dynamic energy of the *PSCP* is decreased considerably when compared to all other networks. We can also observe the significant decrease in the *Path_blocked* dynamic energy which is a direct consequence of the considerable decrease of the *PSCP* dynamic energy. it can be seen that the torus-based system can achieve the same performance of the proposed PHENIC system in terms of bandwidth, at the cost of higher electronic energy.

From these results, we can see that PHENIC outperforms systems whether having non blocking or blocking switches. In addition, it provides much better energy efficiency than the torus-based which can offer the same bandwidth as the proposed system. We can conclude that the obtained improvement by PHENIC is the result of the association of three main factors together: (1) the non blocking switch supporting optical acknowledgment signals, (2) the light-weight router with reduced buffer size, (3) and the path setup algorithm to adopt hybrid switching inside the photonic switch.

6 Conclusion and Future Work

Future generations of many-core system applications with hundreds of cores will require a scalable communication fabric that can enable high performance. To address this challenge, we proposed an energy-efficient and high-performance hybrid Silicon-Photonic Network-on-Chip architecture (PHENIC). The proposed system is based on a novel contention-aware path configuration algorithm and an energy-efficient non-blocking photonic switch. Simulation results show that PHENIC enjoys 50% increase in bandwidth and about 60% decrease in energy related to the control unit, versus other reported architectures. This performance comes from the decrease in the blocking latency and the number of blocked requests. This encouraging results highlight the potential of using photonics on chip and the PHENIC hybrid photonic NoC architecture to meet the design and performance challenges of future generations of many-core systems.

As a future work, we plan to investigate the reliability issue of such complex systems. We also plan to investigate the routing algorithm by providing more path diversity during the path configuration process in the ECN. This aims to further reduce the number of blocked requests and the consequent energy and latency overhead.

References

[1] A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, B. Schieber, M. Sudan, "Efficient routing in optical networks", Journal of ACM, Vol. 43, no 6, 1994, pp. 973-1001.

[2] Z. Chen, H. Gu, Y. Chen, Y. Chen, H. Zhang, "Source-and Destination-based Wavelength Assignment in Optical Network-on-Chip: Design and Performance", Proceeding of the IEEE region 10 (TENCON 2013), 2013, pp. 1-4.

[3] A. Ben Abdallah, "Multicore Systems-on-Chip: Practical Hardware/Software Design, 2nd Edition", Publisher: Atlantis, 2013, ISBN-13: 978-9491216916.

[4] A. Ben Ahmed, A. Ben Abdallah, "PHENIC: Towards Photonic 3D-Network-on-Chip Architecture for High-throughput Many-core Systems-on-Chip", Proceedings of the 14th International conference on Sciences and Techniques of Automatic control and computer engineering, 2013, pp. 1-9.

[5] A. Ben Ahmed, Y. Okuyama, A. Ben Abdallah, "Contention-free Routing for Hybrid Photonic Mesh-based Network-on-Chip Systems", Proceedings of the IEEE 9th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC-15), 2015.

[6] A. Ben Ahmed, M. Meyer, Y. Okuyama, A. Ben Abdallah, "Hybrid Photonic NoC based on Non-blocking Photonic Switch and Light-weight Electronic Router". To appear in the IEEE Int. Conf. on systems, man, and cybernetics (SMC-2015), 2015.

[7] S. M. Bhandarkar, H. R. Arabnia, "The REFINE Multiprocessor: Theoretical Properties and Algorithms", Parallel Computing (journal), Elsevier, Vol. 21, No. 11, 1995, pp. 1783-1806.

[8] A. Ben Ahmed, Y. Okuyama, A. Ben Abdallah, "Non-blocking Electro-optic Network-on-Chip Router for High-throughput and Low-power Many-core Systems". Proceedings

of the IEEE World Congress on Information Technology and Computer Applications (WCITCA'2015), 2015.

[9] L. Zhang, X. Tan, M. Yang, J. Jiang, P. Liu, J. Yang, "Circuit-switched on-chip photonic interconnection network", Proceedings of the 9th International Conference on Group IV Photonics, 2012, pp. 282-284.

[10] C.A.D Adi, H. Mtasutani, M. Koibuchi, H. Irie, T. Miyoshi and T. Yoshinaga, "An Efficient Path Setup for a Photonic Network-on-Chip", Proceeding of the First International Conference on Networking and Computing, 2010, pp. 156-161.

[11] H. R. Arabnia and S. M. Bhandarkar, "Parallel Stereocorrelation on a Reconfigurable Multi-Ring Network", The Journal of Supercomputing (Springer Publishers), Vol. 10, No. 3, 1996, pp. 243-270.

[12] L. Benini, G. De Micheli "Networks on chips: technology and tools", Publisher:Morgan Kauffmann, San Mateo, 2006, ISBN-13: 978-0123705211

[13] K. Mori, A. Esch, A. Ben Abdallah, K. Kuroda, "Advanced design issues for OASIS network-on-chip architecture", Proceedings of the IEEE 5th international conference on broadband, wireless computing, communication and applications, 2010, pp. 74-79.

[14] H. R. Arabnia and J. W. Smith, "A Reconfigurable Interconnection Network For Imaging Operations And Its Implementation Using A Multi-Stage Switching Box", Proceedings of the 7th Annual International High Performance Computing Conference. The 1993 High Performance Computing: New Horizons Supercomputing Symposium, Calgary, Alberta, Canada, June, 1993, pp. 349-357.

[15] A. Ben Abdallah, M. Sowa, "Basic Network-on-Chip Interconnection for Future Gigascale MCSoCs Applications", Communication and Computation Orthogonalization, Proceeding of the Symposium on Science, Society, and Technology, 2006, pp. 4-6.

[16] D. Vantrease, et al. "System implications of emerging nanophotonic technology", Proceeding of the 35th International Symposium on Computer Architecture I(SCA 08), 2008, pp. 153-164.

[17] D. Vantrease, N.L. Binkert, R. Schreiber, M. H. Lipasti, "Light speed arbitration and flow control for nanophotonic interconnects", Proceeding of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-42), 2009, pp. 304-315.

[18] H. R. Arabnia, "A Distributed Stereocorrelation Algorithm", IEEE Proceedings of Computer Communications and Networks (ICCCN'95), 1995, pp. 479-482.

[19] R. Beausoleil et al., "A nanophotonic interconnect for high-performance many-core computation", Proceeding of the 16th IEEE Symposium in High Performance Interconnects, 2008, pp 182-189.

[20] A. Ben Ahmed, M. Meyer, Y. Okuyama, A. Ben Abdallah, "Efficient Router Architecture, Design and Performance Exploration for Many-core Hybrid Photonic Network-on-Chip (2D-PHENIC)", Proceedings of the Int. Conf. on Information Science and Control Engineering (ICISCE), 2015, pp.202-206.

[21] H. R. Arabnia and M. A. Hough , "A Transputer Network for Fast Operations on Digitised Images", International Journal of Eurographics Association (Computer Graphics Forum), Vol. 8 No. 1, 1989, pp. 3-12.

[22] X. Zhang, A. Louri , "A Multilayer Nanophotonic Interconnection Network for On-Chip Many-Core Communications", Proceeding of the 47th ACM/IEEE Design and Automation Conference (DAC), 2010, pp. 156-161.

[23] K. Kirman, J. Martinez, "A Power-efficient All-optical On-chip Interconnect Using Wavelength-based Oblivious Routing", Proceeding of the 15th edition of ASPLOS on Architectural support for programming languages and operating systems, 2010, pp.15-28.

[24] H. Gu, J. Xu, W. Zhang, "A low-power fat tree-based optical network on-chip for multiprocessor system-on-chip", Design, Automation and Test in Europe (DATE), 2009, pp. 3-8.

[25] S. M. Bhandarkar and H. R. Arabnia, "The Hough Transform on a Reconfigurable Multi-Ring Network", Journal of Parallel and Distributed Computing, Vol. 24, No. 1, January, 1995, pp. 107-114.

[26] S. Pasricha, N. Dutt, "ORB: An on-chip optical ring bus communication architecture for multi-processor systems-on-chip", Proceeding of Asia South Pacific Conference Design Automation, 2008, pp. 789-794.

[27] R.W. Morris, A.K. Kodi, A. Louri, R.D. Whaley, "Three-Dimensional Stacked Nanophotonic Network-on-Chip Architecture with Minimal Reconfiguration", IEEE Transactions on Computers, Vol. 63, no 1, 2014, pp. 243-255.

[28] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, A. Choudhary, "Firefly: Illuminating future network-on-chip with nanophotonics", International Symposium on Computer Architecture (ISCA), 2009, pp. 429-440.

[29] Y. Pan, J. Kim, G. Memik, "Flexishare: Channel sharing for an energy-efficient nanophotonic crossbar", International Symposium High-Performance Computer Architecture (HPCA), 2010, pp. 1-12.

[30] H. R. Arabnia, "A Parallel Algorithm for the Arbitrary of Digitized Images using Process-and-Data-Decomposition Approach", Journal of Parallel and Distributed Computing, Vol. 10, No. 2, 1990, pp. 188-193.

[31] H. R. Arabnia and M. A. Oliver, "Arbitrary Rotation of Raster Images with SIMD Machine Architectures", International Journal of Eurographics Association (Computer Graphics Forum), Vol. 6 No. 1, 1987, pp. 3-12.

[32] G. Hendry, E. Robinson, V. Gleyzer, J. Chan, L. Carloni, N. Bliss, and K. Bergman, Circuit-switched memory access in photonic interconnection networks for high-performance embedded computing, Proceeding of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC), 2010, pp. 112.

[33] J. Chan, G. Hendry, K. Bergman, and L. Carloni, Physical-layer modeling and system-level design of chip-scale photonic interconnection networks, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 30, no. 10, pp. 15071520, Oct 2011.

[34] J. Chan and K. Bergman, "Photonic interconnection network architectures using wavelength-selective spatial routing for chip-scale communications", IEEE/OSA Journal of Optical Communications and Networking, vol. 4, no. 3, 2012, pp. 189-201.

[35] A. Shacham, K. Bergman, and L. Carloni, "On the design of a photonic network-on-chip", Proceeding of the First International Symposium on Networks-on-Chip (NoCs), 2007, pp. 53-64.

[36] H. Matsutani, M. Koibuchi, H. Amano, T. Yoshinaga, "Prediction Router: Yet Another Low Latency On-Chip Router Architecture", Proceedings of the 15th IEEE International Symposium on High-Performance Computer Architecture (HPCA 2009), pp. 367-378.

[37] Y. Ye, J. Xu, B. Huang, X. Wu, W. Zhang, X. Wang, M. Nikdast, Z. Wang, W. Liu, and Z. Wang, 3-D mesh-based optical network-on-chip for multiprocessor system-on-chip, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 32, no. 4, 2013, pp. 584-596.

[38] J. Wang et al., "CPNoC: An Energy-Efficient Photonic Network-on-Chip", Proceeding of the 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2013, pp. 1571-1576.

[39] X. Tan, M. Yang, L. Zhang, X. Wang, Y. Jiang, "A Hybrid Optoelectronic Networks-on-Chip Architecture", Journal of Lightwave Technology, Vol.32, no.5, 2014, pp. 991-998.

[40] G. Hendry et al., "Time-Division-Multiplexed Arbitration in Silicon Nanophotonic Networks-On-Chip for High- Performance Chip Multiprocessors", Journal of Parallel and Distributed Computing, vol. 71, 2011, pp. 641-650.

[41] G. Hendry et al., "Silicon nanophotonic network-on-chip using TDM arbitration", Proceeding of IEEE Symposium on High-Performance Interconnects, 2010, pp. 88-95.

[42] A. Shacham et al., "Photonic Networks-on-Chip for Future Generations of Chip Multiprocessors", IEEE Transactions on Computers, vo1.57, no.9, 2008, pp. 1246-1260.

[43] M. Arif Wani and H. R. Arabnia, "Parallel Edge-Region-Based Segmentation Algorithm Targeted at Reconfigurable Multi-Ring Network", The Journal of Supercomputing, Vol. 25, No. 1, 2003, pp. 43-63.

[44] J. Chan, G. Hendry, A. Biberman, K. Bergman, and L. Carloni, "PhoenixSim: A Simulator for Physical-Layer Analysis of Chip-Scale Photonic Interconnection Networks", Design, Automation and Test in Europe (DATE), 2010, pp. 691-696.

[45] S. M. Bhandarkar, H. R. Arabnia, and J. W. Smith, "A Reconfigurable Architecture For Image Processing And Computer Vision", International Journal of Pattern Recognition And Artificial Intelligence (IJPRAI) (special issue on VLSI Algorithms and Architectures for Computer Vision, Image Processing, Pattern Recognition And AI), Vol. 9, no. 2, 1995, pp. 201-229.

[46] A. Ben Ahmed and A. Ben Abdallah, "Graceful Deadlock-Free Fault-Tolerant Routing Algorithm for 3D Network-on-Chip Architectures", Journal of Parallel and Distributed Computing, Vol. 74-4, 2014, pp. 2229-2240.

[47] A. Ben Ahmed and A. Ben Abdallah, "Architecture and Design of High-throughput, Low-latency, and Fault-Tolerant Routing Algorithm for 3D-Network-on-Chip (3D-NoC)", The Journal of Supercomputing, Vol. 66-3, 2013, pp. 1507-1532.

[48] M. Petracca, B. Lee, K. Bergman, L. Carloni, "Design exploration of optical interconnection networks for chip multiprocessors", Proceeding of the 16th IEEE Symposium High Performance Interconnects, 2008, pp. 31-40.

[49] H. R. Arabnia and M. A. Oliver, "A Transputer Network for the Arbitrary Rotation of Digitised Images", The Computer Journal, Vol. 30 No. 5, 1987, pp. 425-433.

[50] H. Wang et al., "On the Design of a 4x4 Nonblocking Nanophotonic Switch for Photonic Networks on Chip", Proceeding of Frontiers in Nanophotonics and Plasmonics, 2007.

[51] C. Sun et al., "DSENT-A Tool Connecting Emerging Photonics with Electronics for Opto-Electronic Networks-on-Chip Modeling", Proceeding of the Sixth IEEE/ACM International Symposium on Networks-on-Chip (NoCs), 2012, pp. 201-210.

[52] L. Brusberg et al., "Single-mode Glass Waveguide Platform for DWDM Chip-to-Chip Interconnects", Proceeding of the 62nd Conference on Electronic Components and Technology, 2012, pp. 1532-1539.

[53] A. Kahng, B. Li, L.-S. Peh, K. Samadi, "Orion 2.0: A power-area simulator for interconnection networks", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol.20-1, 2012, pp. 191-196.

[54] Y. Ye et al., "System-level analysis of mesh-based hybrid optical-electronic network-on-chip", IEEE International Symposium on Circuits and Systems, 2013, pp. 321-324.

[55] K. Preston K et al., "Performance guidelines for WDM interconnects based on silicon microring resonators", Conference on Lasers and Electro-Optics (CLEO), 2011, pp. 1-2.
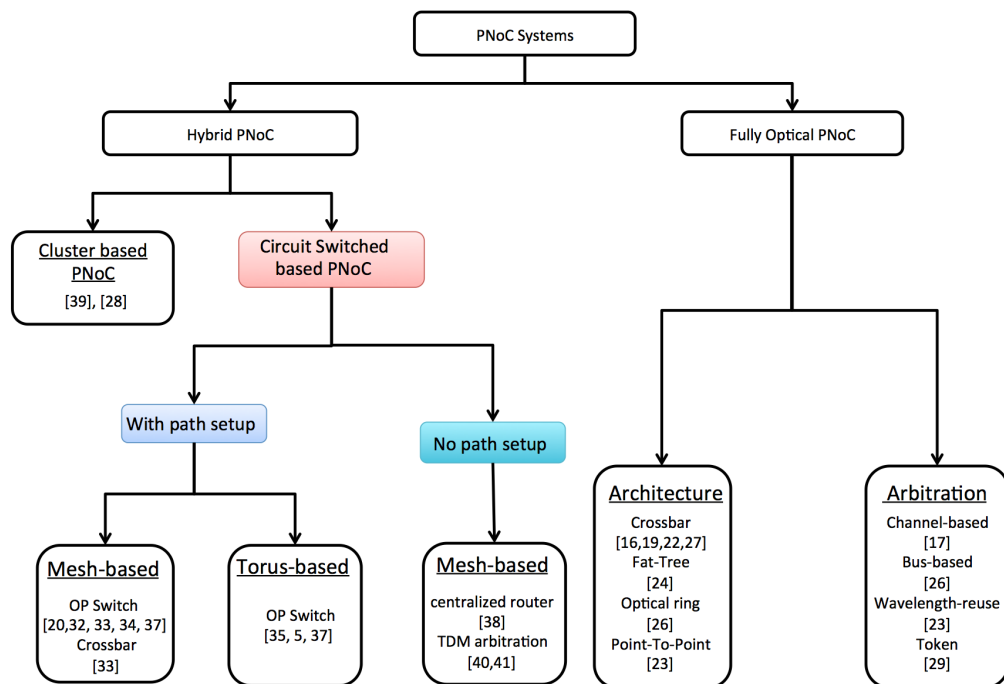
Figure 1: Hybrid and fully PNoC systems taxonomy.

(a)

(b)

(c)

(d)
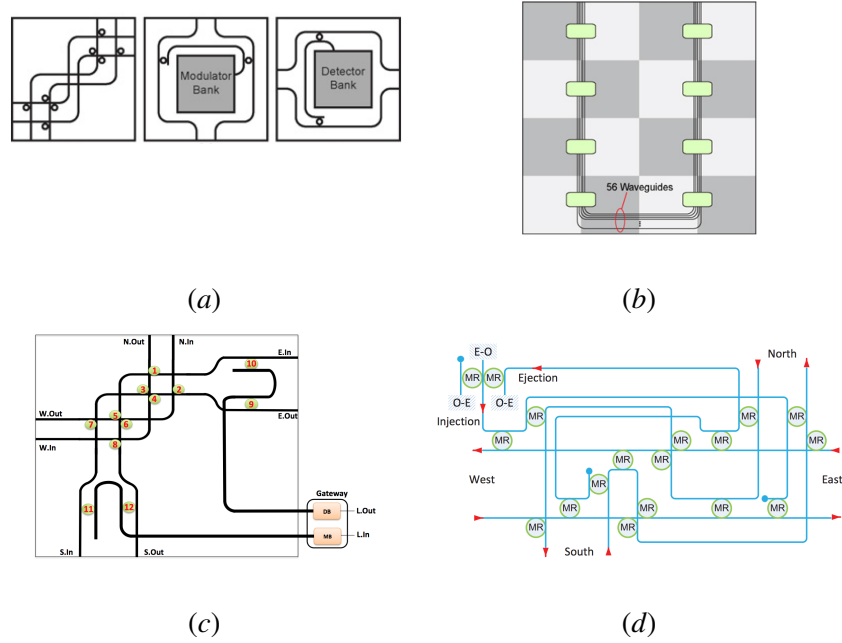
Figure 2: Optical switch and crossbar typologies: (a) 4x4 non blocking switch augmented with two gateways for injection and ejection [33], (b) optical crossbar [33], (c) 5x5 blocking switch [20] [50], (d) 5x5 non blocking switch [54].

Table 1: Micro-ring Configuration For Data Transmission.

output/Input	Local	North	East	South	West
Local	-	9,18	11,18	14	16
North	17,10	-	1	3	None
East	17,12	2	-	None	4
South	13	6	None	-	8
West	15	None	5	7	-
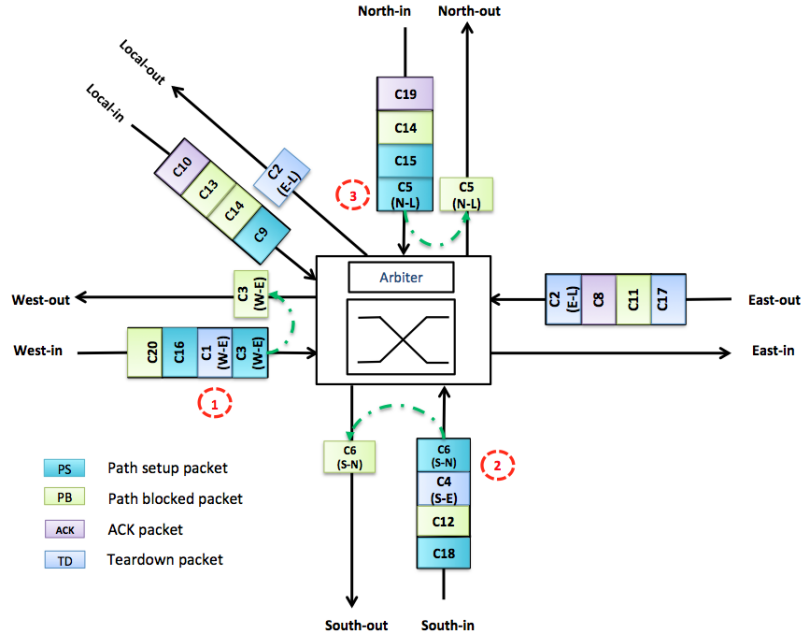
Figure 3: Simplified example illustrating the dependency between the PSCP and Tear-down packets. In cases one and two, the path setup packets for C3 and C6 have been blocked because the needed ports will be released by the following Tear-down Packet in the next cycle. In case three, the path setup packet for C5 has been blocked because for arbitration purpose the Tear-down for C2 is served first.

Table 2: Wavelength Assignment For Acknowledgment Signal Handling.

	Local	North	East	South	West
Input	Mod_{λ_0}	Det_{λ_3}	Det_{λ_2}	Det_{λ_1}	Det_{λ_4}
Output	Det_{λ_0}	Mod_{λ_1}	Mod_{λ_4}	Mod_{λ_3}	Mod_{λ_2}
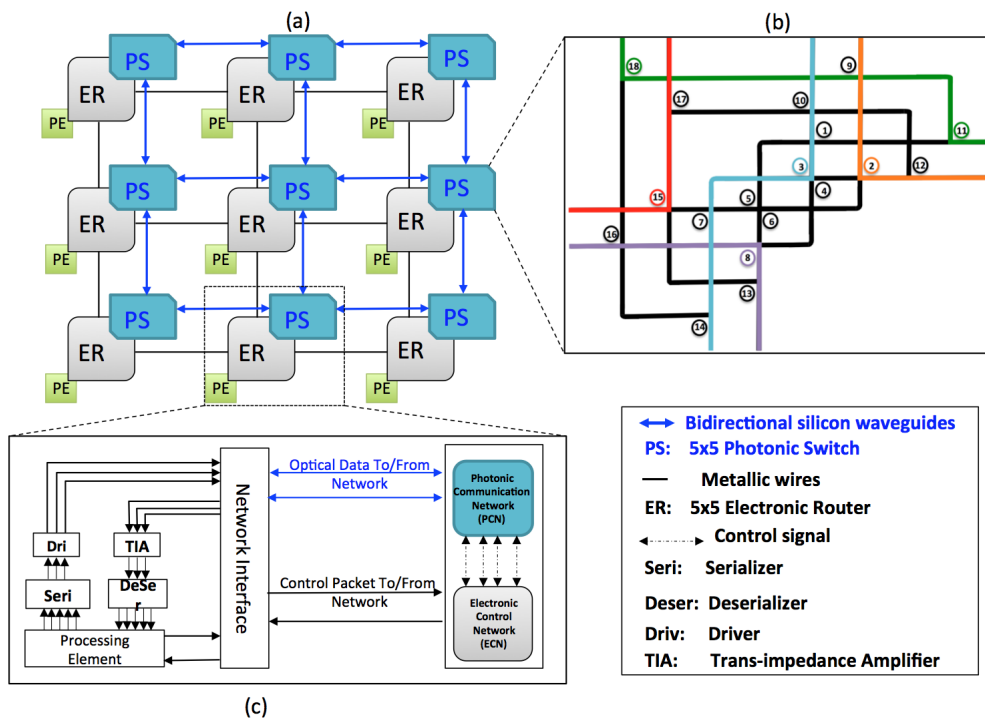
Figure 4: PHENIC system architecture. (a) Hybrid electro-optical router interconnected for a 3x3

mesh-based, (b) 5x5 non-blocking photonic switch, (c) Unified tile including PE, NI and control

modules.
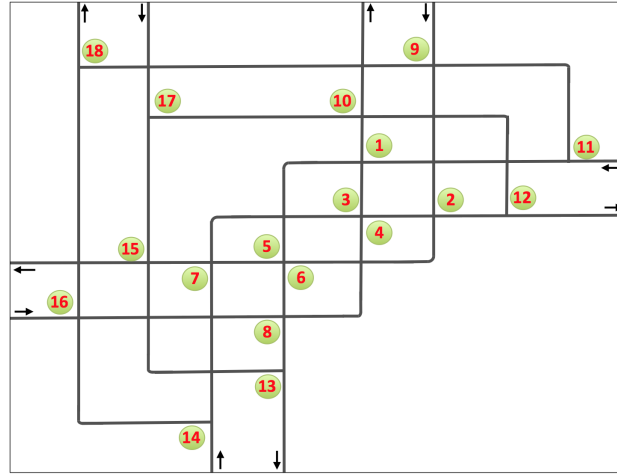
(a)
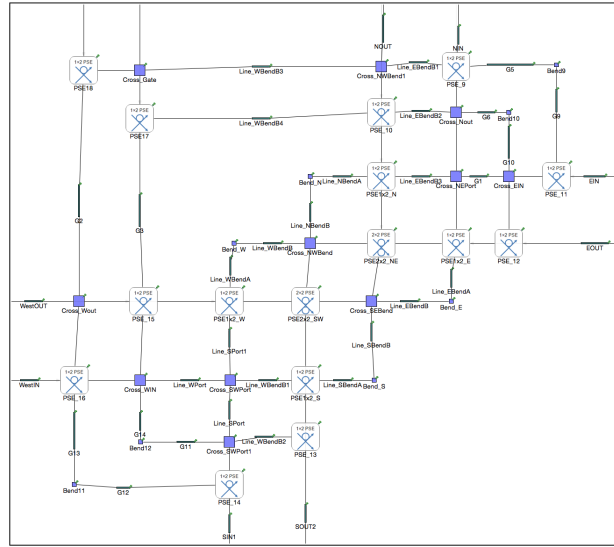
(b)

Figure 5: PHENIC non-blocking photonic switch.(a) Microring assignment, (b) Photonic components instantiation.

Algorithm 1: Path-configuration Algorithm: Path setup and Path blocked.

```
// Path Setup Control Packet for communication i, PSCPi

// Path Blocked Packet for communication i, PBi
```

Input: S_i, D_i

```
// From ACK detector
```

Input: $Detc_{ACKs}$

```
// To ACK modulator
```

Output: Mod_{ACKs}

```
// From Teardown detector
```

Input: $TeardMod_i$

```
// To Teardown modulator
```

Output: $TeardMod_i$

```
// To Microring resonator
```

Output: $MRs_{j=0...n}$

```
// Buffer writing and routing computation stages
```

1   initialization;

2   **while** *(Path-Setup-Control-Packet (PSCP) !=0)* **do**

3   DestAdd ← PSCP*i*;

4   PortIn ← PSCP*i*;

5   **if** *(resource are available)* **then**   /* check MRs state */

6           $Grant_i$ ← Arbiter;

7   **else**   /* generate path blocked */

8           $Blocked_i$ ← Arbiter;

9   **end**

10 **end**

```
// Path blocked
```

11 initialization;

12 **while** *(PB !=0)* **do**   /* Path blocked arrives */

13   **if** *(MRsi state is **reserved**)* **then**   /* release reserved MRs */

14   release ← MRs*i*;

15 **end**

Algorithm 2: Path-configuration Algorithm (Cont.): ACK, Payload Transmission and Teardown.

   // Generate ACK

1   initialization;

2   **while** *(NI receiver ← PSCPi)* **do**   /* PSCP arrives to Dest */

3   **if** *(PSCP arrives to NI)* **then**   /* generate ACK to Src */

4   ACK_i ← To modulator ACK ($\lambda 0$);

5   **end**

   // Receives ACK and Payload Transmission

6   initialization;

7   **while** *(NI receiver ← ACK_i ($\lambda 0$))* **do**   /* ACK arrives to Src $\lambda 0$ */

8   **if** *(ACK arrives to the NIsender)* **then**   /* modulate the data */

9            $Data_i$ ← To Data's Modulator;

10   **end**

   // Identify and Generate *Teardown$_i$*

11   initialization;

12   **while** *(From detector signal =Teardown$_i$ with λi)* **do**

13   *findInport* ← λi;   /* find In-port according to the wavelength */

14   free ← MRs*i*;   /* Free involved MRs */

15   *Teardwon$_i$* ← To modulator λi;   /* generate new Tear-down according to λi */

16   **end**
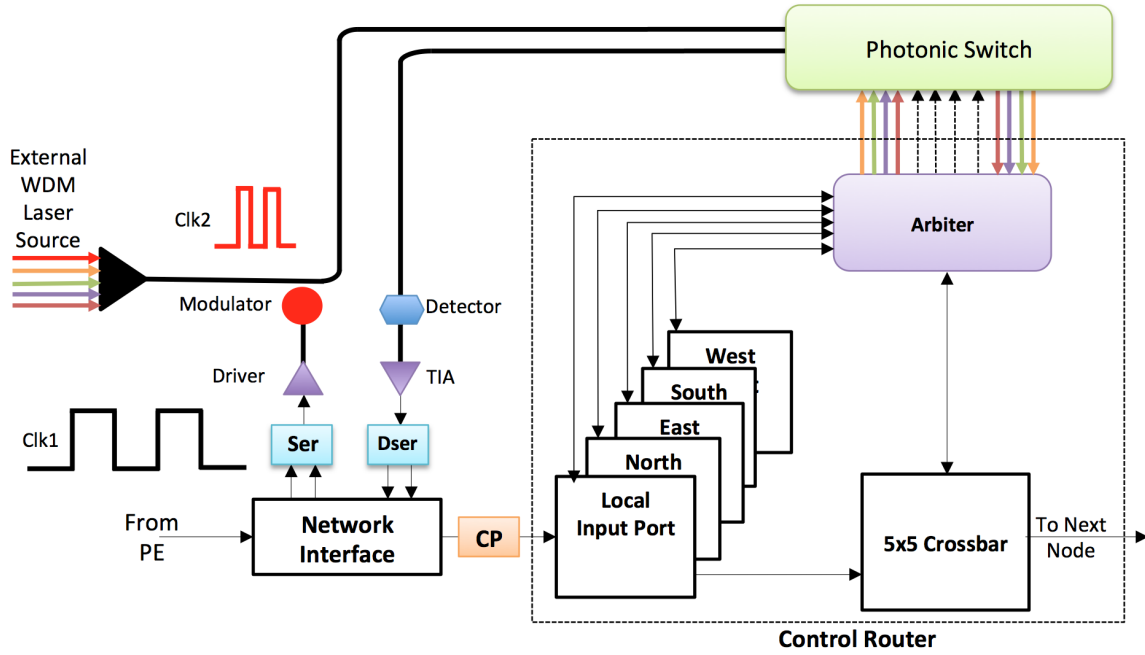
Figure 6: PHENIC Light-weight electronic router.

Table 3: Configuration Parameters.

Network Configuration	Value
Process technology	32nm
Number of tiles	256,64
Chip area (equally divided amongst tiles)	400 mm^2
Core frequency	2.5GHz
Electronic Control frequency	1GHz
Power Model	Orion 2.0
Buffer Depth	2
Message size	2 kilobytes
Simulation time	10ms (25 10^8 cycles)
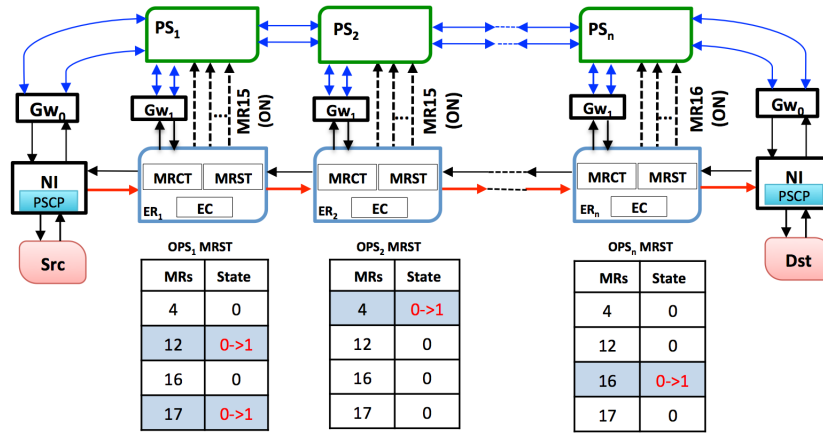
(a)
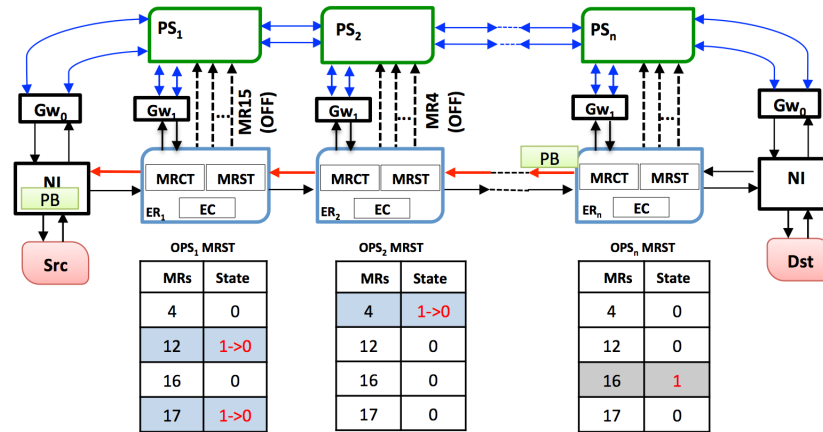
(b)

Figure 7: Path-configuration Algorithm: a) Path-setup, (b) Path-blocked. GW_0: Gateway for data, GW_1: Gateway for acknowledgment signals, PS: Photonic Switch, MRCT: Micro Ring Configuration Table, MRST: Micro Ring State Table.
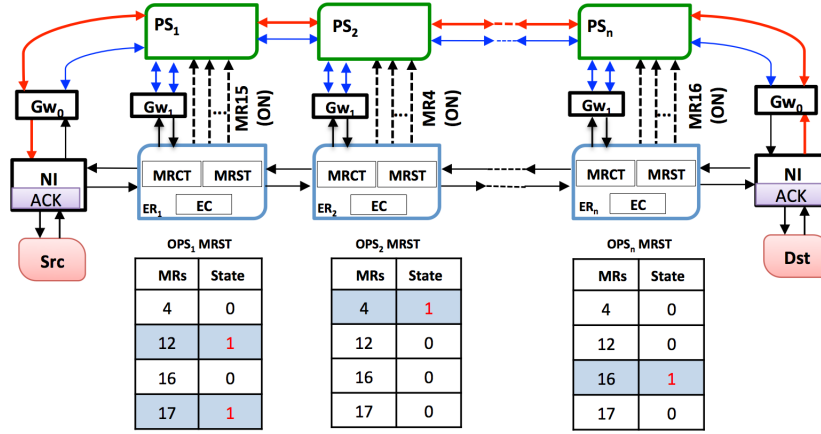
(c)
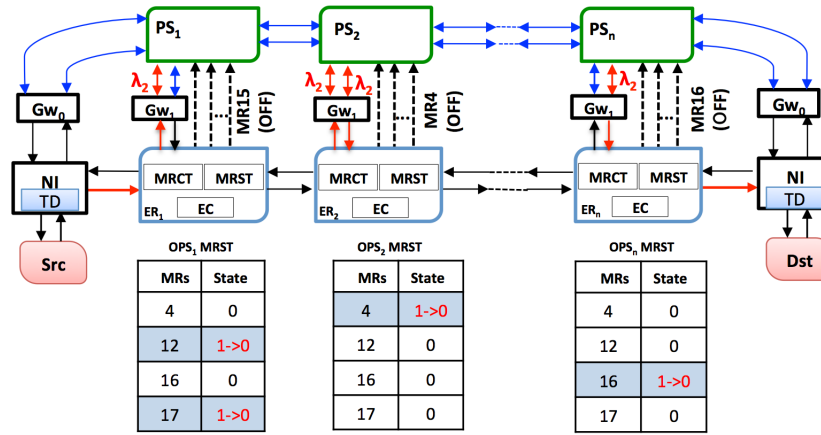
(d)

Figure 8: Path-configuration Algorithm (Cont.): (c) Acknowledgment, (d) Tear-down. GW_0: Gateway for data, GW_1: Gateway for acknowledgment signals, PS: Photonic Switch, MRCT: Micro Ring Configuration Table, MRST: Micro Ring State Table.

Table 4: Photonic Communication Network Energy Parameters.

Network Configuration	Value
Datarate (per wavelength)	2.5GB/s
MRs dynamic energy	375fJ/bit
MRs static energy	400 μ W
Modulators dynamic energy	25fJ/bit
Modulators static energy	30 μ W
Photodetector energy	50fJ/bit
MRs static thermal tuning	1μW/ring

Table 5: Ring Requirement Comparison Results For 64 Cores Systems.

	PHENIC	PHENIC_BL	Chan_Mesh	Chan_Xb	Shacham
Mod/Detc	64	64	64	64	64
Switch	1152	852	768	1152	1620
ACKs	640	-	-	-	-
Total	1856	916	832	1216	1684
Static Thermal Tuning (mW)	37	18	16	24	33

Table 6: Ring Requirement Comparison Results For 256 Cores Systems.

	PHENIC	PHENIC_BL	Chan_Mesh	Chan_Xb	Shacham
Mod/Detc	256	256	256	256	256
Switch	4608	3252	3072	4608	6324
ACKs	2560	-	-	-	-
Total	7424	3508	3328	4864	6580
Static Thermal Tuning (mW)	149	71	67	98	131

(a) (b)
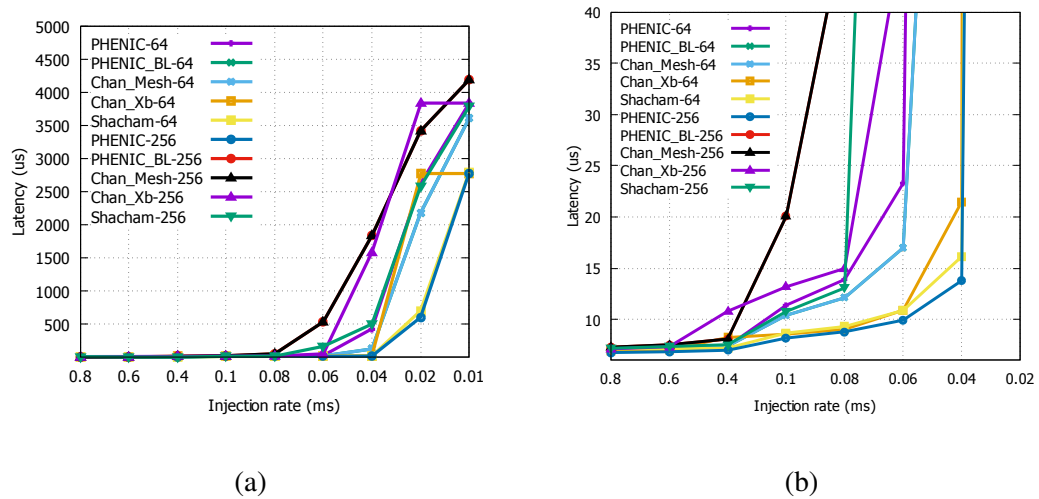
Figure 9: Latency comparison results under random uniform traffic: (a) Overall Latency, (b) Latency near-saturation.
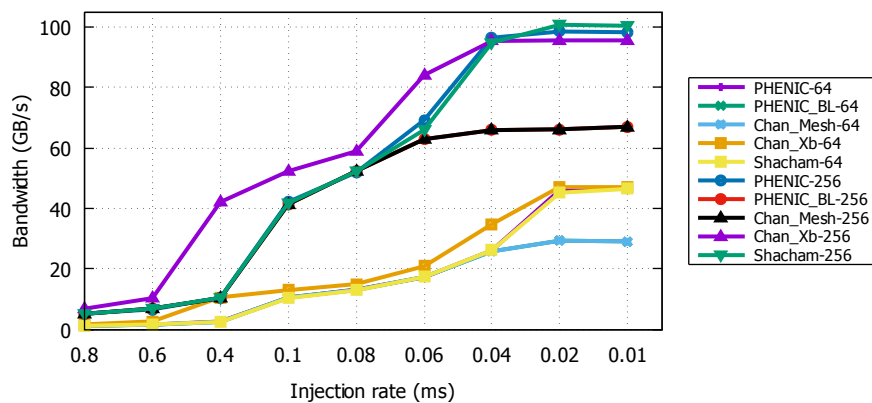
Figure 10: Bandwidth comparison results under random uniform traffic.
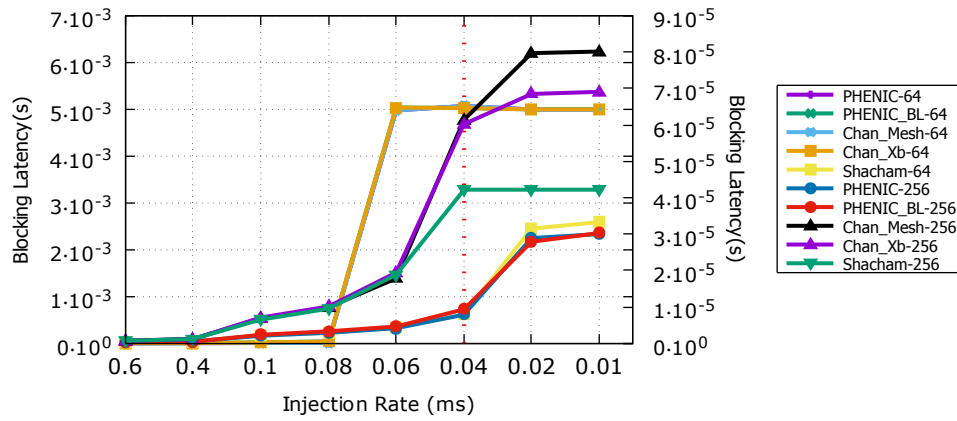
Figure 11: Average blocking latency comparison under random uniform traffic. The left Y-axis shows the blocking latency for PHENIC_BL and Chan_Mesh networks and the right Y-axis for PHENIC, Chan_Xb and Shacham networks.
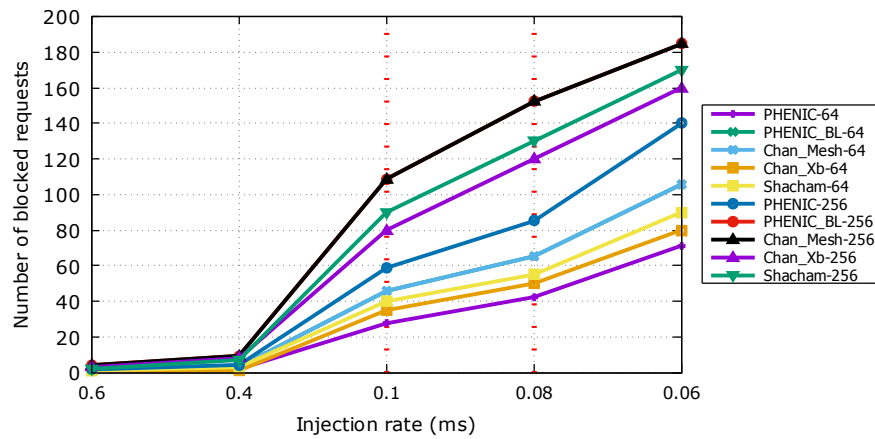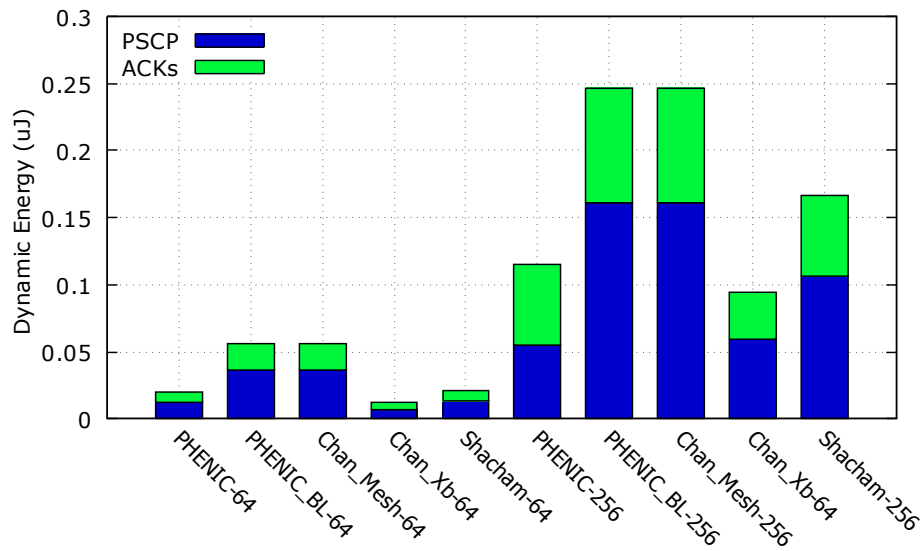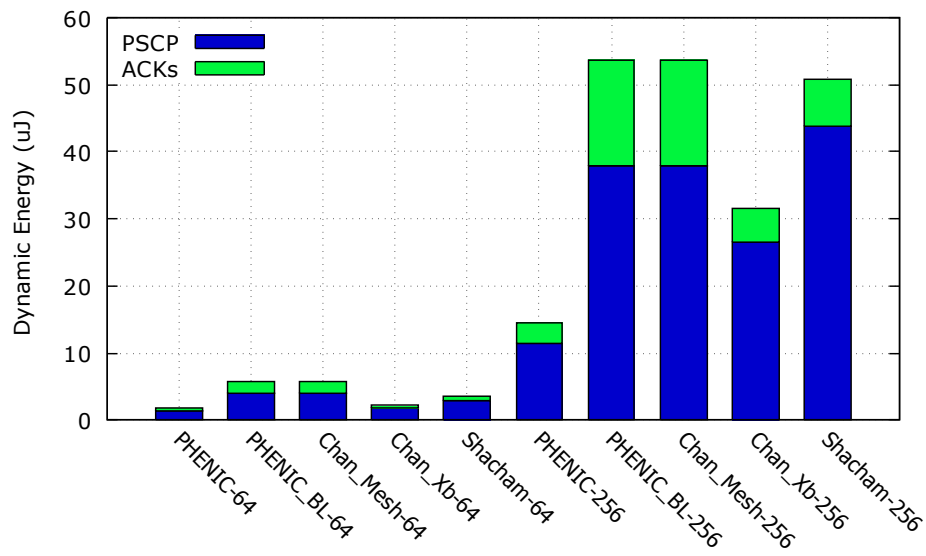
Figure 12: Comparison under uniform traffic Number of blocked requests having reaching more than the half network diameter. The vertical dashed line represent the near-saturation point.

(*a*)

(*b*)

Figure 13: Path setup and acknowledgments energy: (a) half-load under random traffic, (b) near-saturation under random traffic.
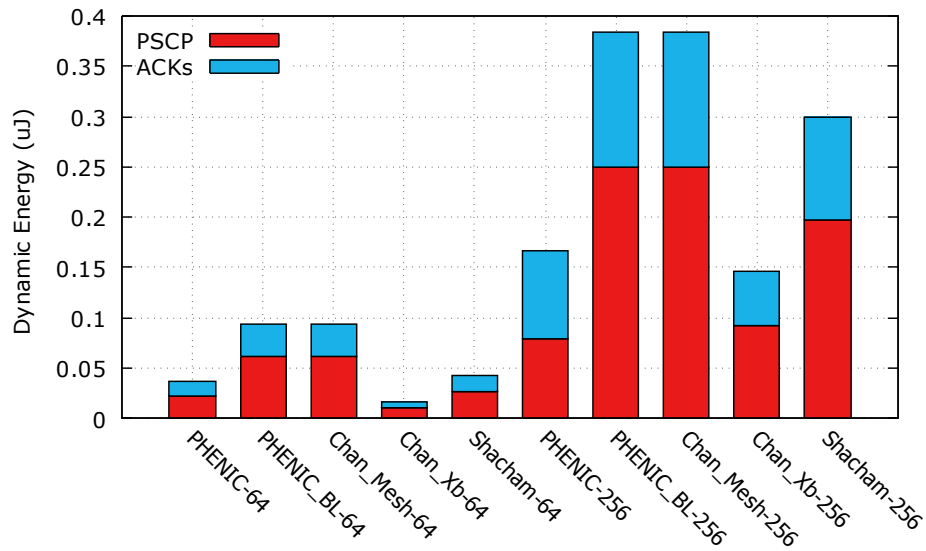
(c)
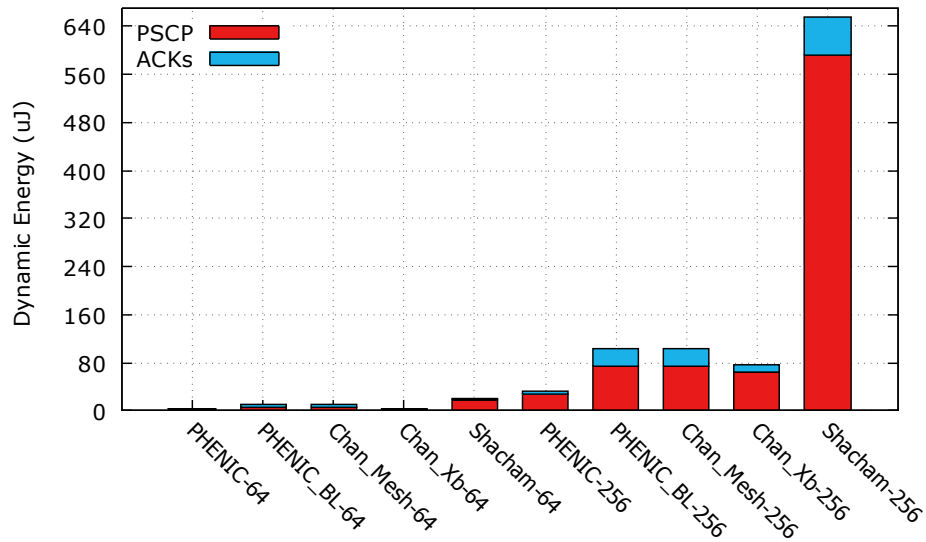
(d)

Figure 14: Path setup and acknowledgments energy (Cont): (c) half-load under bitreverse traffic, (d) near-saturation under bitreverse traffic.
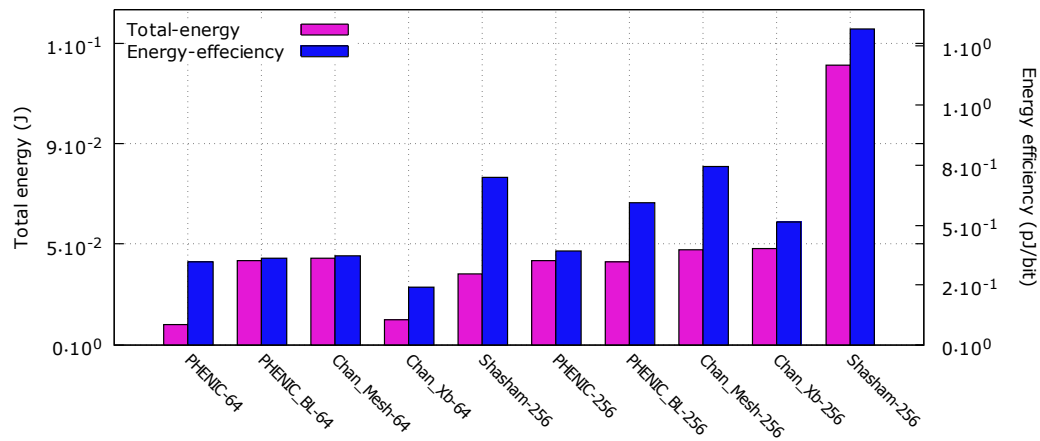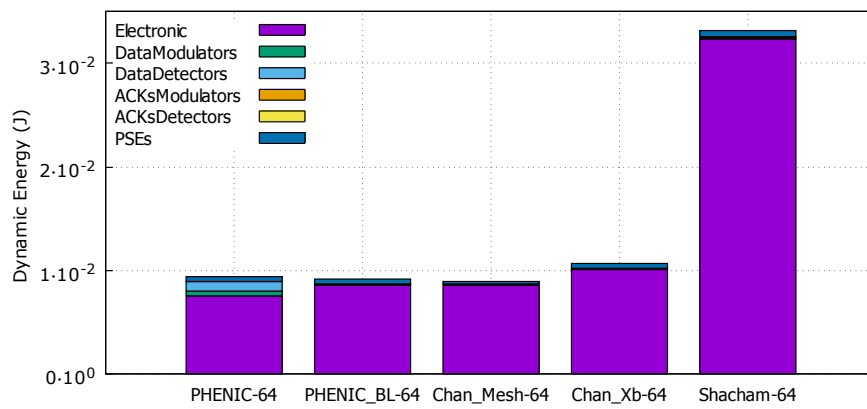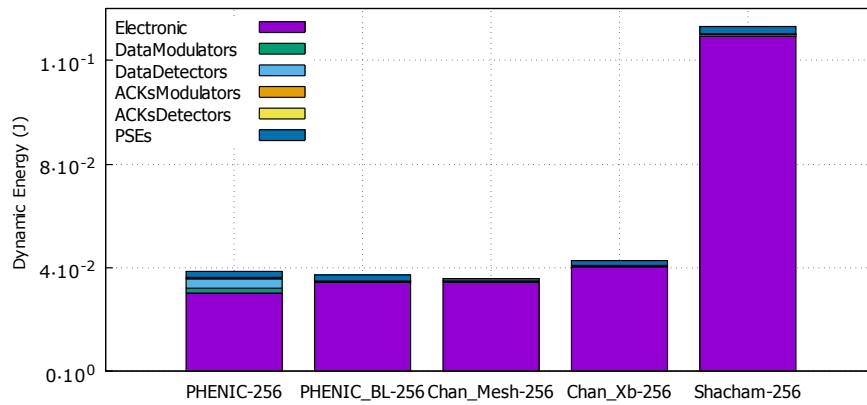
Figure 15: Total energy and energy efficiency comparison results under random uniform traffic near-saturation.
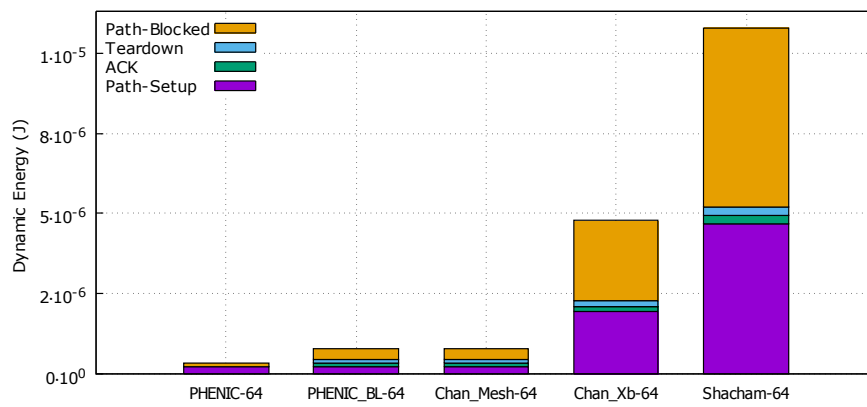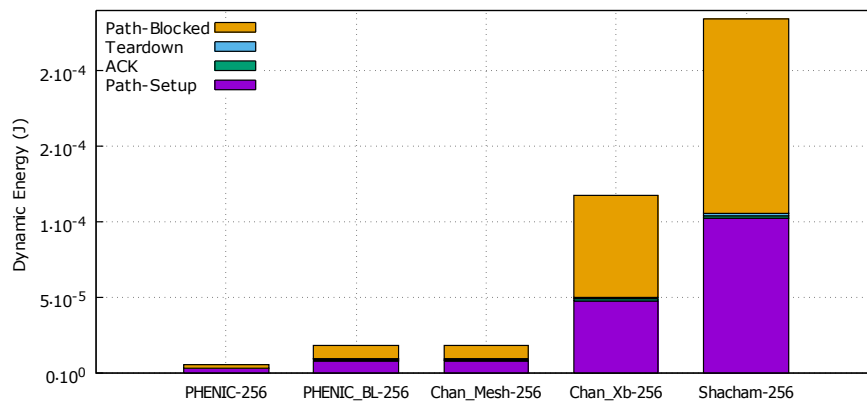
(a)

(b)

Figure 16: Total Energy breakdown comparison under random uniform traffic near-saturation. (a) 64 cores systems, (b) 256 cores systems.

(a)

(b)

Figure 17: Input-buffer dynamic energy breakdown near-saturation. (a) 64 cores systems, (b) 256 cores systems.