Received December 1, 2020, accepted January 8, 2021, date of publication February 9, 2021, date of current version March 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3058238

A Scalable Software Defined Network Orchestrator for Photonic Network on Chips

DOAA A. HAMDI[1], SAMY GHONIEMY[2], YASSER DAKROURY[1], AND MOHAMMED A. SOBH[1]

[1]Department of Computer and Systems, Faculty of Engineering, Ain Shams University, Cairo 11566, Egypt

[2]Department of Computer Networks, Faculty of Computer and Information Sciences, The British University in Egypt, Cairo 11837, Egypt

Corresponding authors: Doaa A. Hamdi (doaaahmed197@gmail.com), Samy Ghoniemy (samy.ghoniemy@bue.edu.eg), Yasser Dakroury (yasser.dakroury@eng.asu.edu.eg), and Mohammed A. Sobh (mohamed.sobh@eng.asu.edu.eg)

ABSTRACT Photonic networks and software-defined networks are two promising technologies to improve network-on-chips performance, scalability, and resource utilization. Several architectures suffer from scaling limitations, high-power consumption, and noise interference drawbacks. In this paper, an enhanced photonic network-on-chip architecture called (SD-PNoC) is presented. The proposed architecture based on a hybrid hardware-software approach, and a Software-Defined Management Orchestrator (SDMO) to separate the network control and data forwarding planes. This orchestrator has hosted on the upper hardware router as a virtual layer capable of dynamic management. It reconfigures data forwarding paths and allows dynamic execution of different algorithms in real-time, as it scales the proposed topology based on both applications and the network requirements. The proposed SD-PNoC architecture, hierarchical communication protocols, and orchestrator management policies were implemented, simulated, and tested using a customized Phoenix-SIM framework in the OMNIT++ simulation environment. Numerous simulation experiments under different conditions have been performed and have proven that the performance of the proposed architecture is better than that of the conventional electronic network on chip (ENoC). Furthermore, simulation results without using the management policies showed that the latency is reduced by 46% and 25.5% for the 4×4 and 8×8 network structures, respectively. While the power consumption is reduced by 76.5% and 78.5% for the 4×4 and 8×8 network structures, respectively. Besides, the chip area is reduced by 33.4%. Moreover, simulation results of SDMO with using the management policies for the 8×8 network structures increased the enhancement of latency from 25.5% to 37.3% and the power consumption from 78.5% to approximately 80%, which assure the ability of the proposed architecture to remarkably enhance the overall performance of complex network-on-chip structures.

INDEX TERMS Network-on-chip (NoC), photonic network-on-chip (PNoC), software-defined networking (SDN), reconfigurable NoC.

I. INTRODUCTION

The technological advances in IC fabrication techniques have allowed the integration of a large number of processing cores in a System-on-Chip (SoC), e.g. IBM's TrueNorth [1] neuromorphic transistor chip with 4,096 cores networked together, and Intel's Loihi [2] neuromorphic Many-core Processor with On-Chip Learning that supports scaling to 4096 on-chip cores and, hierarchical addressing, up to 16,384 chips. This fact allows the possibility of carrying out complex operations and algorithms but brings more challenges. The resulting growth of High-Performance Computing (HPC) systems, supercomputers and data centers are increasingly reshaping the network infrastructure and the interconnection network [3], which on the other hand increased the energy consumption and cost budgets [4]. In recent years, Networks-on-Chips (NoCs) have been proposed as a methodology for simplifying and improving the design of Multiprocessor Systems-on-Chips (MPSoC) [3], [5], [6] and [7], 3D architectures [8], routing and mapping algorithms have been proposed [9], [10], [11] and [12], Several reconfigurable ONoCs have been presented [13], [14], [15] and [16] so far. However, they either suffer from considerable area overhead and low efficiency. However, the objective of finding a general platform that suits any network for flexible chip control had a great deal in the research community. Researchers found that the concept of Software Defined Networks (SDN) can be applied to NoC [17], [18], [19], and [20]. The combination of these concepts

The associate editor coordinating the review of this manuscript and approving it for publication was Rongni Yang.

produced a novel NoC management technology, which was called Software Defined Network on Chip (SDNoC) [21] and [22]. Several papers have been devoted to the control of NoC using the scope of SDN, [23] and [24] proposed a Software-Defined Photonic Network-on-Chip (SD-PNoC) architectures based on the principle of a centralized control plane separated from the network forwarding or switching plane but they suffer from scaling limitations and high-noise interference problem. While [25] presented a hybrid hierarchical SD-PNoC that improved the scalability problem by introducing a hierarchical control plane with inter/intra communication protocols. On the other hand, the authors of [26] used the Bus-based NoC to introduce an SDN controller that permits a run-time reconfiguration of the data forwarding plane allowing the execution of different algorithms in run time. In [19], [27], [21], and [28], A SDNoC architecture and a performance evaluation using System C models are presented. The configuration time, delay, and throughput were evaluated for different routing algorithms (RA) e.g. XY, West First, North Last, etc. Results proved that the deterministic routing, such as XY routing, achieves the best performance among other adaptive routing algorithms. Since, adaptive RA is a better choice for low traffic load, while in high traffic loads, both routing algorithms approach the same results. However, the presented research showed a remarkable enhancement in the overall performance, the optimized performance against the energy consumption, and the area still research gaps.

Based on those works, this paper evaluates the communication latency, power consumption, and chip area of 2D-mesh NoC. The comparative analysis focuses on standard baseline conventional electronic NoC and the proposed Software-defined PNoC similar in architecture to the one presented in [21] and [29]. However, the main goal of this work is to identify an SD-Orchestrator including software management policies capable of passing requests upward/downward layers, reconfiguring data forwarding paths, and processing elements. Characterized among other models by a clear separation between the data forwarding plane and the control plane, this separation provides flexible management for the central controller that makes it possible to improve the utilization of all network-on-chip resources. This management layer added to the photonic layer to adjust the addressing schemes for routing levels used by the routing algorithm, based on deterministic RA with the SDNoC approach to produce the best communication performance for MPSoC's. Furthermore, hardware modification for data forwarding (Processing) plane made by summing up all Network Interface Function (NIF) to a concentrated gateway, leaving the network routers in the photonic layer, while using an arbiter/controller to connect local processors cores with a single shared NIF.

The rest of the paper is organized as follows; Section 2 discusses the proposed system architecture, including the topology, communication protocol, micro-architecture of the centralized controller. Then, Section 3 presents the network management plane, management policies, and required addressing scheme. Finally, the performance evaluation and conclusion are discussed in section 4 and section 5 respectively.

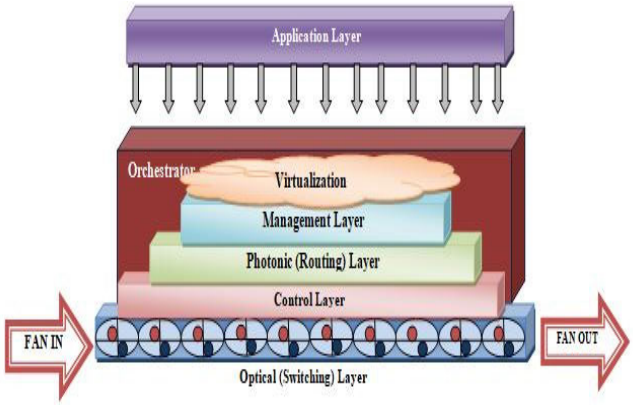

FIGURE 1. Proposed SDN-PNoC.

II. THE PROPOSED SOFTWARE DEFINED CONTROLLER FOR PHOTONIC NETWORK ON CHIP

Software-Defined Networking (SDN) refers to a new approach for network programmability, that is, the capacity to initialize, control, change, and manage network behavior dynamically via open interfaces. SDN emphasizes the role of software in running networks through the introduction of abstraction for the data forwarding plane and separating it from the control plane. This separation allows faster innovation cycles at both planes. However, there was an increasing confusion as to what exactly SDN is, what is the layer structure in an SDN architecture, and how layers interface with each other.

In this section, the proposed Software-Defined Controller for Photonic Network on Chip (SD-PNoC) architecture is presented. Then, the proposed network controller and Network Interface Function (NIF) micro-architecture is produced. After that, communication protocols, and path setup discussed. Finally, Software Management is applied to the proposed hierarchy and plentifully expressed.

A. SYSTEM ARCHITECTURE

Achieving high performance from a many-core, the multiprocessor system depends on the best utilization of both the computational and the communicational resources. The proposed architecture is simplified in a sketch presented in Fig.1 that is based on a merger of two concepts, software-defined networks (SDN) and a Photonic network on chips (PNoC). The architecture presented at first look seems like a hierarchy, but this is for a clear system description. The presented system consists of four sub-layers; these sub-layers can work separately or simultaneously.

1) PROCESSING PLANE

This plane consists of the two bottom layers, the Optical (switching) layer with Fan in/out process and the Control layer. Based on the architecture described in [30],

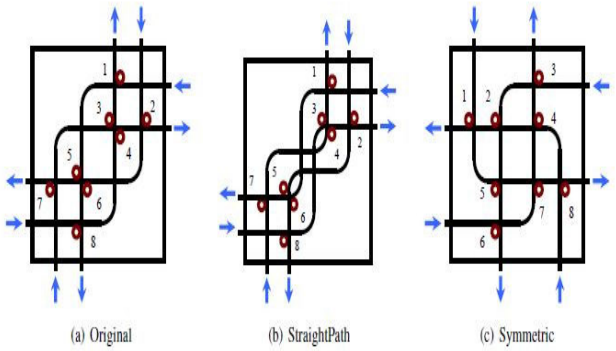

FIGURE 2. Various 4 × 4 non-blocking switch designs, (a) the original 4 × 4 non-blocking switch, (b) a 4 × 4 non-blocking switch that reduces the number of crossings, and (c) a 4 × 4 non-blocking switch that minimizes insertion loss for the straight path cases (North-South, and West-East).

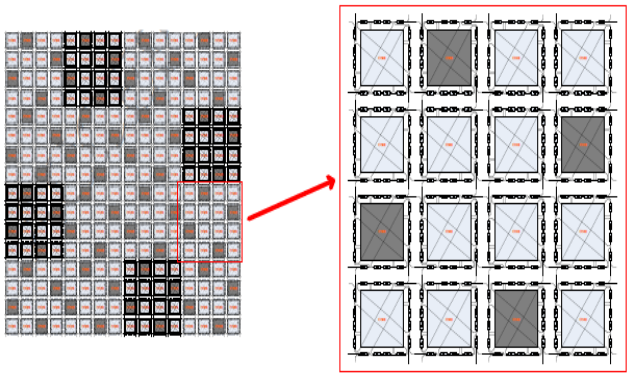

FIGURE 3. Proposed data plane.

A 16 × 16 mesh is constructed from 4 × 4 photonic switches connected to form a mesh topology with 4 × 4 clusters, The clusters are generally small in sizes, such as 2 × 2, 3 × 3, 4 × 4, or 5 × 5, that what makes up the data plane of the software-defined network. In our architecture, we used a 4 × 4 non-blocking Straight-Path switch from switch designs described in Fig. 2 (b), which reduces the number of crossings to reduce power losses to form a 256-core system with a 16-core cluster combination described in Fig. 3. The proposed architecture also tested for a more complex network structure of an 8 × 8-mesh network structure using an asymmetric photonic switch since it can save even more loss by eliminating a number of the crossings.

The power budget is one of the major design constraints of any photonic network, which is determined by the nonlinear threshold (Ψ) of devices that will start to induce nonlinear effects. The first, a non-linear effect to be induced in the waveguide as follows

$$\sum_{\lambda}^{N} P\boldsymbol{\lambda} \leq \Psi_{wg} \tag{1}$$

The sum of the optical power of all the wavelengths at a point in a waveguide must be less than the nonlinear threshold power, where N is the number of wavelengths. The second nonlinear critical point is in a modulator, which says:

$$P\boldsymbol{\lambda} \leq \Psi_{mod} \tag{2}$$

Since a modulator is resonant with only one wavelength, we are only concerned with the optical power of a single wavelength. While the real value of P_λ should be P_λ -ζ_{delivery}, where ζ_{delivery} is the optical loss from the laser source to delivery point to the first modulator, ζ_{delivery}, is usually negligible, and therefore we only consider the injected power P_λ. Finally, for circuit-switched networks, the third potential source of nonlinear effects is in a broadband switch, which says:

$$\sum_{\lambda}^{N} P_\lambda \leq \Psi_{swich} \tag{3}$$

The sum of the optical power of all the wavelengths going through the switch must be less than the nonlinear threshold power Ψ_{switch}. Since the switch that will experience the most amount of power is the one immediately following the modulator bank, where $\zeta_{\text{modulation}}$ is the loss experienced passing through the modulator bank and modulated. The power of each wavelength will be:

$$P\boldsymbol{\lambda} = P_{lamda} - \zeta_{delivery} - \zeta_{modulation} \tag{4}$$

In [30], The control plane is distributed to the processing nodes; each node controls and configures a photonic switch using a dominating node in each row in each cluster. Core concentration is accomplished by using ordinary network-side concentration, shown in Fig. 4 (a). The network-side concentration involves modifying the network switches and routers to accept multiple injection/ejection points, such as increasing the number of ports (radix) in an electronic router. Network-side concentration requires no changes in the processing plane because each Processor still has one NIF, which it uses to interface to the increased-radix routers. In this paper, the core concentration accomplished using an alternate approach using a concentrated gateway (Controller) which leaves the network routers in the photonic layer, while using a switch to connect local processors multi-cores with a single shared NIF. This switch, shown in Fig. 4 (b), represents the proposed control layer of the proposed Software-Defined photonic NoC orchestrator. Processor router, routes between processing elements (PE) and collectively orchestrate the data plane (i.e. The photonic switches) to implement data paths required for the application by providing a framework for processing elements to request and receive services (i.e., Setting-up routing paths) from the network via a centralized SDN controller. SDN-based control makes the procedure of path set-up simpler. By sending requests to the central controller, the source PE avoids the traditional hop-by-hop consequences (i.e., Delay). Therefore, it is better because it saves on the number of modulators and detectors needed, and thus power and area.

2) NETWORK PLANE

To illustrate the idea of the proposed photonic network plane, a 4 × 4 switch shown in the upper part of Fig. 5 is used as a base. In the proposed solution, a modulator and detector banks have been added to the East and South ports, respectively. The proposed components are designed

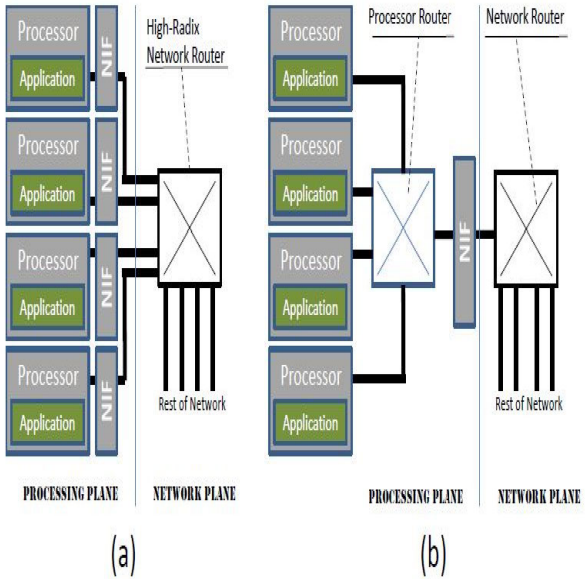

FIGURE 4. Core concentration, (a) Network-side, (b) Concentrated gateway (Controller).

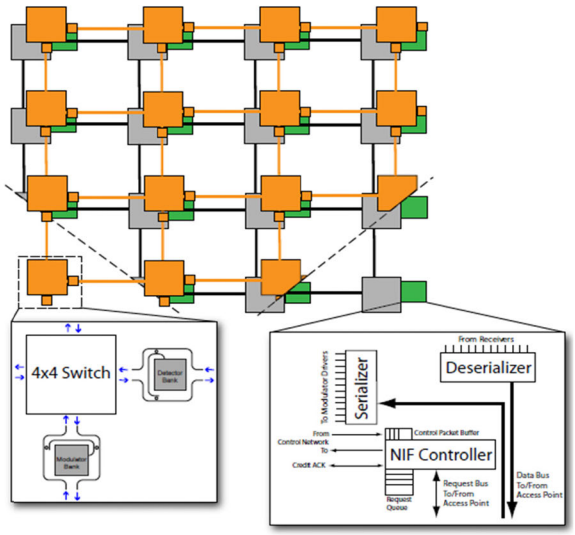

FIGURE 5. A 4 × 4 Mesh topology, showing 5-port photonic switch and NIF block diagram.

based on the optimized switch illustrated in Fig.3 to highly reduce the insertion loss. To analyze and evaluate the proposed multi-layer network, a high crossing StraightPath switch illustrated in Fig. 2 (b) is used as a test case. Considering the mesh network design shown in Fig.5, the closed-form approximation for worst-case loss in the network is given by Eq. (5);

$$\zeta_{network} = \zeta_{mod-det} + \zeta_{inj-ej} + (2N-1) \times \left(\zeta_{switch-thru} + \zeta_{Prop}\right) + \zeta_{switch-drop} \quad (5)$$

where $\zeta_{networkmod-det}$ is insertion loss from passing through the modulator and the filter bank, ζ_{inj-ej} Is from dropping through rings to be injected into the network, $\zeta_{switch-thru}$ Is

TABLE 1. Insertion loss optimization parameters.

Parameter	Symbol	Single-layer (dB)	Multi-layer (dB)
Modulation	ζ_{mod}	1.2	1.2
Filter-drop	$\zeta_{filter-drop}$	0.5	-
Filter-thru	$\zeta_{filter-thru}$	0.05	0.05
Crossings	ζ_{cross}	0.05	0
Waveguide	ζ_{wg}	0.5	0.5
Bend	ζ_{bend}	0.005	0.005

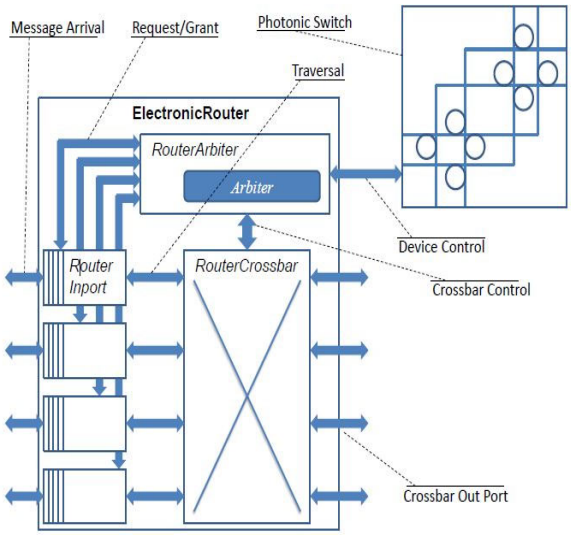

FIGURE 6. Proposed network plane.

passing straight through a switch (without dropping through a ring), ζ_{drop} Is making a turn at a switch (dropping through a ring), and ζ_{Prop} Is the propagation loss (as a function of distance) between the switches. These terms defined as follows:

$$\zeta_{mod-det} = \zeta_{mod} + 2\sigma_{filter} + \zeta_{filter-drop} \quad (6)$$

$$\zeta_{inj-ej} = 2\zeta_{ring-drop} \quad (7)$$

$$\zeta_{switch-thru} = 6\zeta_{cross} + 4\zeta_{ring-thru} + 2\zeta_{bend} + \zeta_{wg} \times S_{switch} \times 1.5 \quad (8)$$

$$\zeta_{switch-drop} = 5\zeta_{cross} + \zeta_{ring-thru} + 2\zeta_{bend} + \zeta_{wg} \times S_{switch} \times 1.5 \quad (9)$$

$$\zeta_{prop} = (S_{chip}/N - S_{switch})\zeta_{wg} \quad (10)$$

where (S_{chip} and S_{switch} Are the sizes (of one side) of the chip and switch, respectively. Table 1 shows the remarkable enhancement in the insertion loss using the proposed multi-layer network compared with the conventional single-layer network.

As shown in Fig.6, the network plane proposed as a photonic switch controlled by an electronic packet-switched router. The photonic switch consists of a 4 × 4 non-blocking Photonics switch, which is carefully composed of the photonic basic building blocks, including 1 × 2 and 2 × 2 photonic switching elements (PSE), micro-ring resonators (MRR), and

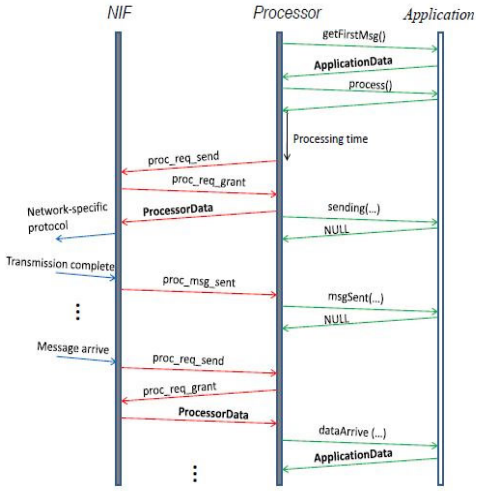

FIGURE 7. Communication between application, Processor, and NIF.

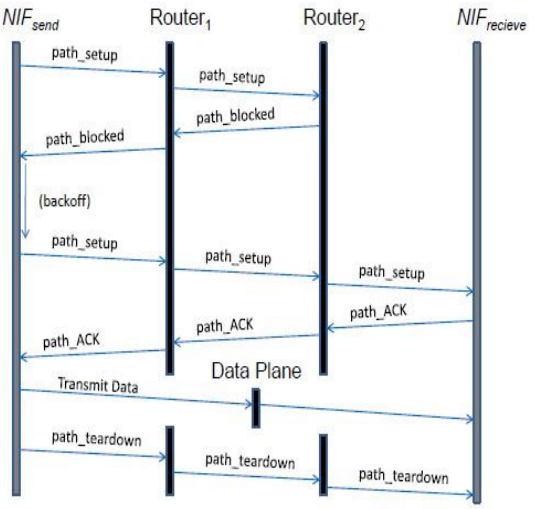

FIGURE 8. Path setup protocol.

waveguides [31]. An electronic router in Fig.6, is a basic store-and-forward packet-switched router [32]. It consists of three sub-modules working together, including router arbiter, router crossbar, and router in port. The router is modeled to have a 3-stage pipeline: message arrival and request, arbitration, and switch traversal. Once the electronic router and the photonic switch connected, it creates a circuit-switched network where the router arbiter performs routing calculations and switch allocations according to addressing schemes identified in the management plane. To manage the photonic layer, control messages travel through the centralized controller to enable the photonic switches to trace out and complete path setup from source to destination. The management plane and addressing schemes are presented in the next subsections.

B. COMMUNICATION PROTOCOLS

A typical communication in the processing plane between the NIF, Processor, and Application can be described as communication events shown in Fig.7 for an Application generates the first message, then waited until it received a message from another core. The three main protocols are color-coded by the interacting pairs as application event timing (green), NIF-Processor communication (red), and NIF-Network communication (blue).

Circuit-switched networks require a control mechanism to set up end-to-end circuit paths. This mechanism is done on a separate physical router, which is the proposed SD-PNoC controller. In Fig.8 an example of a path setup was described, the resources required at Router2 conflicted, and a path-blocked message returned to the NIF that requested the path setup. A linear back-off period was implemented for congestion control. At the receiving end, when the NIF receives a path setup, it returns a path ACK, where the sending NIF can transmit the data on the data plane, and finally release the reserved network resources with a path teardown message.

III. PROPOSED SOFTWARE MANAGEMENT PLANE

The Management Plane of the proposed SD-Orchestrator is a software management algorithm applied to the router's arbiter of the network plane. The Router Arbiter contains the basic algorithm for ensuring that a message has correctly routed through implementing three main functions: routing, contention, and device setup based on the addressing that is clearly described in the next subsections.

A. NETWORK MANAGER

The NM allocates routes based on the global information that is stored in the NM itself. This global information describes the addressing configuration of the data network. The network manager NM has two functions. The first function is collecting messages from the NIs and the second is distributing configuration instructions to the switches and NIs. The topology to perform these functions can be a tree or a fat tree. A reasonable control network is characterized by short messages assumed. Thus, (1) The Arbiter has a Round-robin scheduling [33] approach to provide fairness in delivering requests. (2) Photonic Arbiter implements the logic for handling various path-setup messages depending on the addressing schema. (3) The proposed reconfigurable optical software manager is a separate layer in the orchestrator as illustrated in Fig. 9 by the dashed lines and is designed using wavelength selective optical switches that utilize one or multiple wavelength channels carrying management optical data signals and it configures wavelength channels and routes dynamically. The proposed design facilitates and provides an identical number of hops in both the forward and backward paths, from the NM to each NI and each switch and vice versa to ensure symmetrical synchronization of configuration messages.

B. THE ADDRESSING

How cores in multi-processor system architecture addressed is a very critical issue. Considering the 256-core system

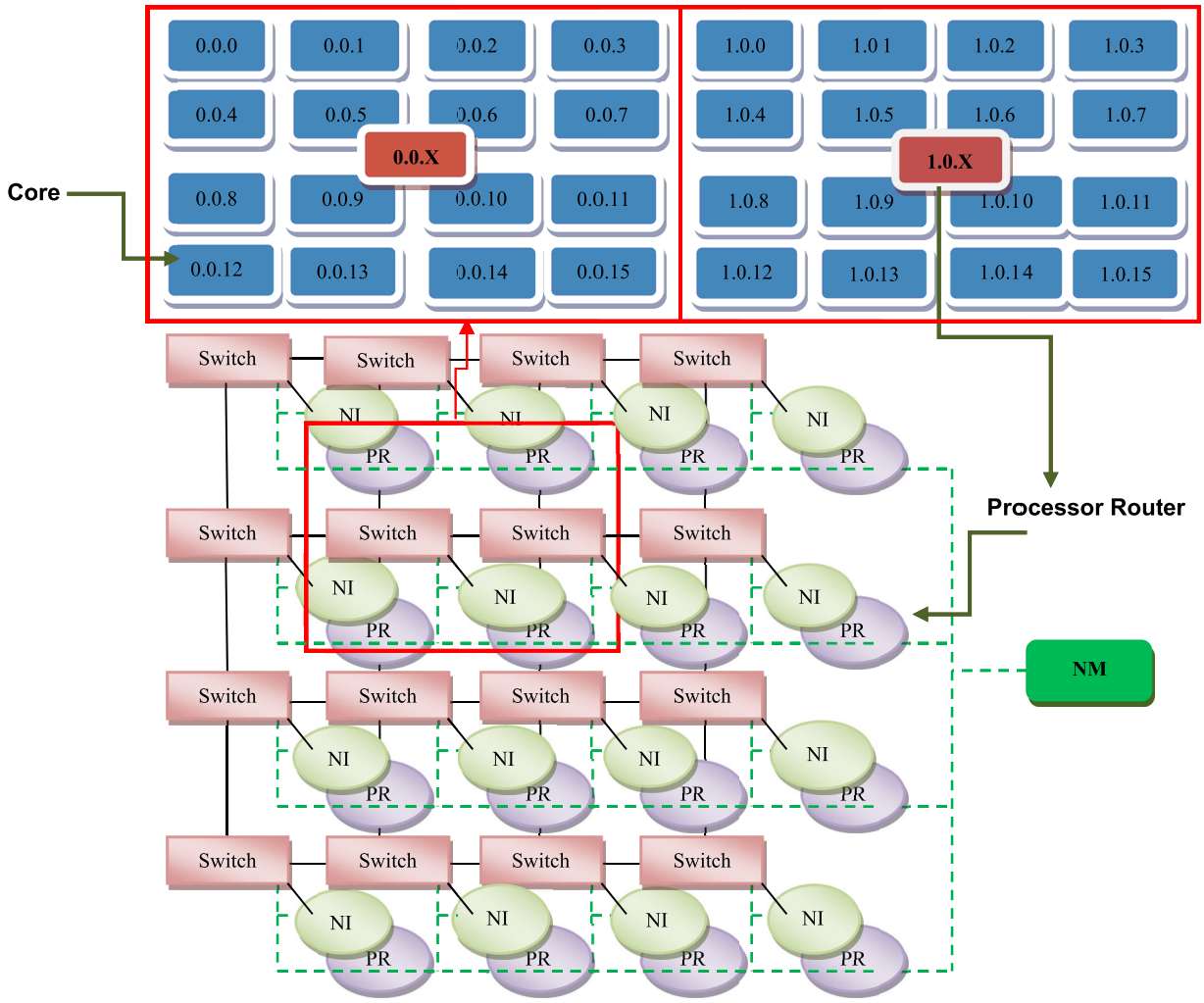

FIGURE 9. The proposed software-defined network manager with addressing schema.

structure made of a 4 × 4-grid topology and 4 × 4 cluster, a hierarchal addressing scheme is proposed, the leftmost address considered as the top, and the rightmost is the bottom. The format of addresses is:

$$\mathbf{NET} : \mathbf{MEM} : \mathbf{PROC}$$

where NET is the network domain or the routers, MEM is the memory gateway domain and PROC is the processor domain. We can see how addresses are assigned in Fig.8. Since the routers are'' top'' level, the Arbiter has only the NET domain defined which specifies which domain it is. The process for routing up and down the address domains defined in arbiters as a management algorithm for parsing the addresses. The algorithm implements three functions: (1) Route function to decide where to forward the message when in the same address domain. (2) Getup Port function decides which port to go to when we need to go up a domain. (3) Get-down function to decide which port to go to when we need to go down a domain.

When a message reaches an arbiter, it looks at the destination address, starting with the top level. Depending on which address does not match the arbiter's address, and which level the arbiter is will dictate which function is called. The algorithm used to select the routing level and the routing algorithm within the same level described in Algorithm 1.

IV. SIMULATION EXPERIMENTS AND PERFORMANCE EVALUATION

A. SIMULATION SETUP

In this section, a conventional electronic network-on-chip and the proposed SD-PNoC system architectures have been simulated using OMNET++ platform [34], with a modified version of the PhoenixSim framework [35], This simulation environment is suitable to investigate both electronic and photonic NoC, it provides a detailed physical model for basic photonic building blocks, such as waveguides, modulators, photo-detectors, and switches. Electronic performance based on the ORION simulator. We simulated a 4 × 4.2 D-mesh topology with 16 core clusters where the processor frequency is 1.6 GHz to form 256-core system architecture. This structure tested for conventional ENoC and the proposed SD-PNoC in terms of communication latency, power consumption. Besides, the proposed architecture tested for a

Algorithm 1 The Algorithm of Selecting Routing Level and x-y Routing Within the Same Level

Requirements: The allowed longest distance D, Network Address, Destination ID (des X, desY), the source routers' location (myX, myY), and routing level (numX, numY).
Ensure: The output port or routing level is OutLev and output port of the message packet is Out PA

1. **if** abs(desX - myX) + abs(desY – myY) ≤ D **Then**
2. **for** i ∈ [myY, numY], j ∈ [myX, numX] **do**
3. **if** (myY =0) **then**
4. OutLev ← Node_N
5. **else-if** (myY = numY − 1) **then**
6. OutLev ← Node_S
7. **else-if** (myX =0) **then**
8. OutLev ← Node_W
9. **else-if** (myX = numX − 1) **then**
10. OutLev ← Node_E
11. **Else**
12. OutLev ← Node_Out
13. **end if**
14. **end for**
15. **for** i ∈ [myY, desY], j ∈ [myX, desX] **&&**
16. [myY = Network Address] **do**
17. **if** (desY > myY)**then**
18. OutPA ← Node_S
19. **else-if** (desY < myY) **then**
20. OutPA ← Node_N
21. **else-if** (desX > myX) **then**
22. OutPA ← Node_E
23. **else-if** (desX < myX) **then**
24. OutPA ← Node_W
25. **else**
26. OutPA ← Node_Out
27. **end if**
28. **end for**

more complex structure of an 8 × 8.2D mesh topology with the same clusters of 16 cores producing 1024-core system architecture. The proposed architecture produces a 33.4 % reduction in chip area compared to the conventional ENoC structure. As the electronic buses and connections require more chip area compared to optical ones. Fig. 10 shows the proposed system architecture in the simulation environment.

Table 2 and Table 3 show the simulation parameters for 4 × 4 and 8 × 8 system architectures in the hardware and the photonic communication networks.

B. SIMULATION RESULTS AND DISCUSSIONS

In this section, two experiment scenarios have been simulated and tested individually. The first scenario discusses the performance of our proposed SD-PNoC architecture without any management policies compared with a conventional electronic network on chip (ENoC). The second scenario discusses the performance of our proposed architecture

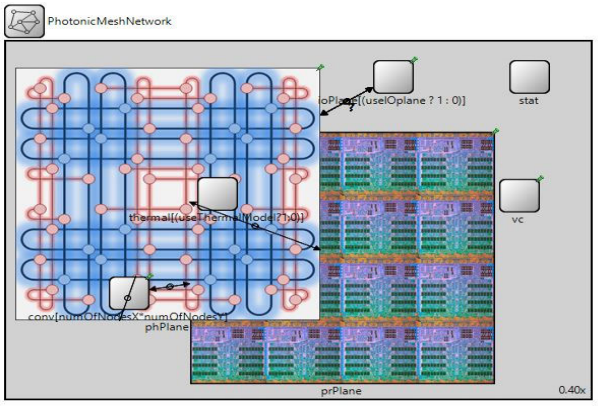

FIGURE 10. Proposed system architecture on PhoenixSim.

TABLE 2. Hardware configuration parameters.

Property	Value	Comment
Processor Technology	32 nm	
Processor concentration	16	
Chip area	400 mm2	8x8 structure in conventional chip needs to increase to 600 mm2
Core frequency	1.6 GHz	
Electronic control frequency	2.5 GHz	

TABLE 3. Photonic communication network parameters.

Network Configuration	Value
Data rate (per wavelength)	2.0 GB/s
Intra Packet time	50e-9 s
Latency Rate Line	1.14e-14 s/um
Propagation Delay	1.14e-14 s/um
Thermal Ring's tuning power	1e-6 J/s/degree/ring
Modulator Dynamic energy	85 fJ/bit
Modulator Static energy	30 uW
Photo-detector energy	50 fJ/bit
MRs Dynamic energy	375 fJ/bit
MRs Static energy	400 uW

with SDMO and management selective policies compared to the proposed architecture without policies and conventional network on chips.

1) WITHOUT MANAGEMENT POLICIES

Figure 11 (a) and (b) compare the average latency of both conventional electronic network (ENoC) and the proposed software-defined photonic network (SD-PNoC) using 4 × 4 and 8 × 8 network structures, respectively. Both Figures 11(a) and 11(b) illustrate that the latency of the conventional network is better than the proposed photonic network for message size up to 500 bytes this because the transmitted messages should be processed and modulated using a photonic network before sending which increases the overall latency. Compared to the conventional network, these messages are

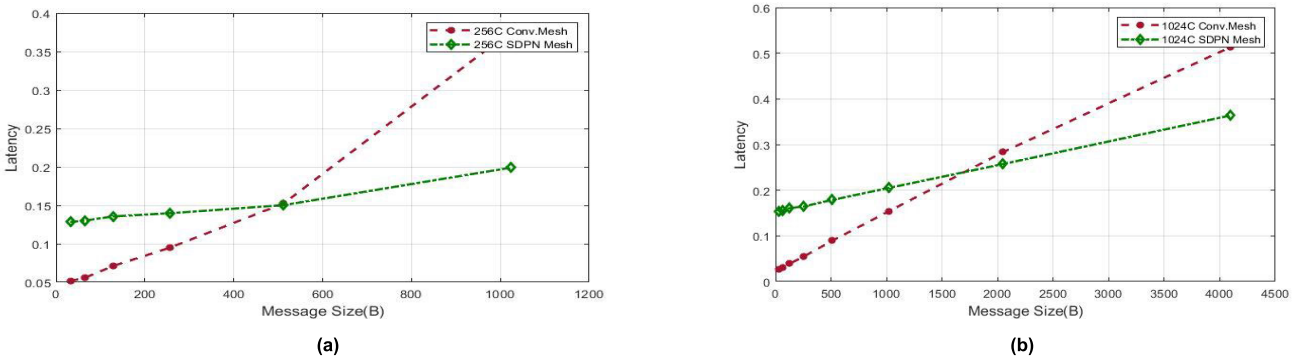

FIGURE 11. Latency performance, (a) a 4 × 4-network structure, (b) an 8 × 8-network structure.

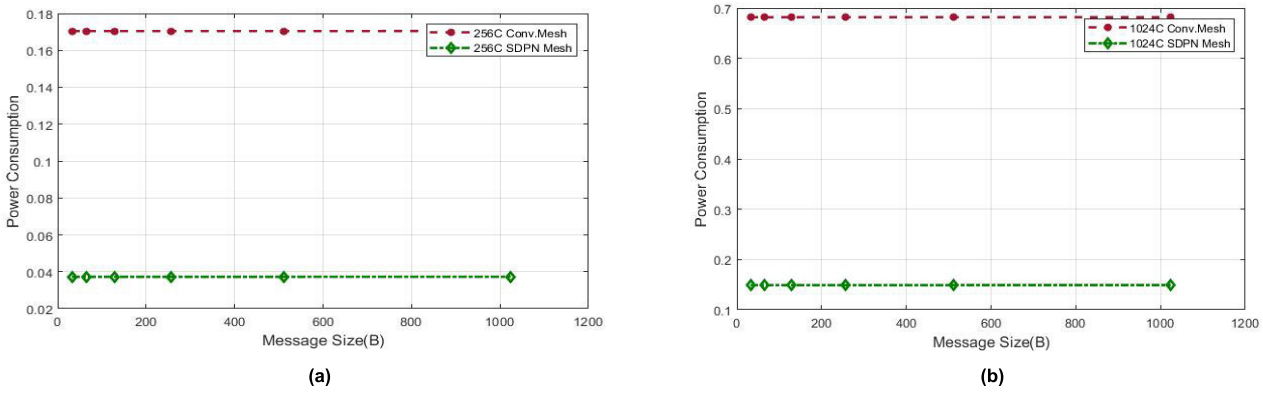

FIGURE 12. Power consumption performance, (a) a 4 × 4-network structure, (b) an 8 × 8-network structure.

sent directly and their size is much less than the data rate. However, for the larger messages, the photonic network performs faster and the overall latency remarkably decreases compared with that using the conventional network because the overall processing and modulation time is much less than the time used in the conventional network to divide, process, and transmit large messages. Moreover, for larger messages, the switching time (speed) of modulation and the bandwidth are much larger than that of the conventional network. Thus, as the figures illustrate, there is a 46% enhancement in latency of a 4 × 4-network structure and 25.5% in latency performance of an 8 × 8-network structure. Furthermore, we evaluated the energy consumption performance for the two network architectures as shown in Fig.12 (a) and (b). The SD-PNoC improved the average energy consumption of the order of 76.5% compared to the conventional ENoC energy consumption in the case of a 4 × 4 network structure whereas 78.5% in an 8 × 8 network structure. Most of the power consumption in our architecture comes from modulation and detection of optical signals, in addition to the crossbars and buffers of the electronic control network. On other hand, ENoC uses pure electronic paths and switching elements that consume much power. This demonstrates the power of our proposed system architecture interconnects over the conventional architecture interconnect. However, compared to the network structure in [20], their architecture enhanced the power consumption by 72.4%, while our proposed architecture produces an extra 6% enhancement to power consumption.

2) WITH A SOFTWARE DEFINED MANAGEMENT ORCHESTRATOR (SDMO) AND POLICIES

In this section, various simulations are presented. Both the latency and power consumption of the system have been evaluated after overlaying software SDMO with policies on an 8 × 8-mesh SD-PNoC architecture, in which the buffer size is 2176 bit and the channel width is 136 bit. The first policy concerns the message size to control the processing network scalability. This policy also chooses to send the small control messages to the electronic network while sending larger message sizes optically. However, we found that the performance of this policy is limited by how large do the electronic buffer and channel widths need to be to provide good performance at low power. As shown in Fig.13 (a), a little improvement in average latency when a size policy was added to our proposed architecture. It enhanced the performance by 3% over pure SD-PNoC with no policies and a total of 28.5% compared to conventional ENoCs. While in Fig.13 (b), illustrates the energy consumption performance. The proposed policy performs well with smaller messages but degrades the overall performance with a larger message size. The second policy is another possible selection policy based on traveling distance. This policy aims on sending messages, which are destined for nearby access points to the control network,

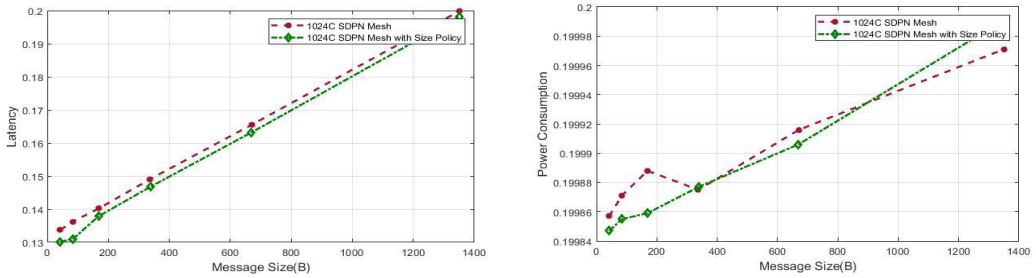

FIGURE 13. An 8 × 8-network structure with size policy performance, (a) Latency, (b) Power consumption.

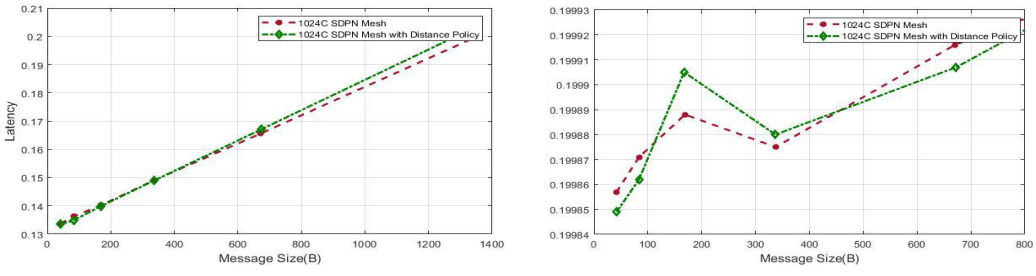

FIGURE 14. An 8 × 8-network structure with distance policy performance, (a) Latency, (b) Power consumption.

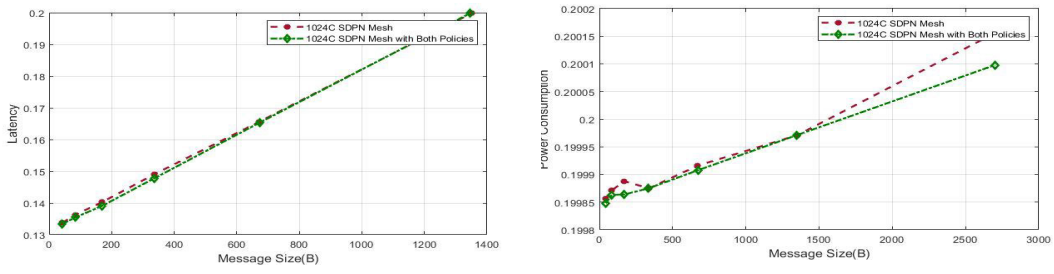

FIGURE 15. An 8 × 8-network structure with both policies performance, (a) Latency, (b) Power consumption.

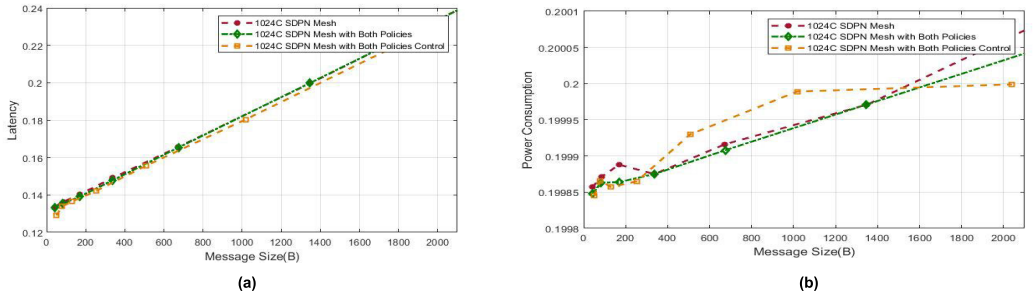

FIGURE 16. An 8 × 8-network structure with both policies and control performance, (a) Latency, (b) Power consumption.

and leaving long-distance communication up to the photonic network. Fig.14 (a), illustrates the latency performance of the system after adding the distance policy. It is noted that adding this policy did not remarkably enhance the overall performance; it even becomes worst when larger size messages are used. However, this result is explainable since this policy concerns only with the availability, distance, and ignores other factors such as multiplicity of wavelengths, utilization of resources …etc. This is why we looked at enhancing this policy by combining other factors/policies such as size,

distance, and controllability, which ended up taking about a complete orchestrator instead of separate management policies. Furthermore, as shown in Fig.14 (b), it begins with a low energy consumption compared to the original architecture. It increases and eventually, after the system saturates, it produces better performances by 1% and 79.5% enhancement compared to conventional ENoCs even with bigger messages. This probably due to the possibility of large messages being transmitted on the electronic network, though not going very far end up blocking circuit-path setup messages. The third policy we consider is the combination of the two previous policies, where the messages selected by-product of their size and distance traveled. As illustrated in Fig.15 (a), the latency performance has no improvement compared to the original structure as the message size increases. However, this performance is reasonable since the adding size policy slightly enhances the latency while adding distance policy slightly degrades the latency performance. As a result, combining these two policies returns the original system performance. However, as shown in Fig.15 (b), no enhancement was noted at the beginning, but it had a 1% reduction in energy consumption and 79.5% enhancement compared to conventional ENoCs as far as the increase in message size was concerned. Finally, given that merging two earlier policies does not significantly improve the system performance, we have considered adding additional control over these policies. The control was applied to limit the number of messages and inter-arrival time. As shown in Fig.16 (a), latency has an improvement over the previous performance of 11.8%. This reflects an approximate 37.3% enhancement compared to conventional ENoCs. While in Fig.16 (b), it begins with a small amount of energy consumption compared to the original architecture and architecture with two policies, then it grows and finally, after the system saturates, it performs better through the improvement of an 80% compared to conventional ENoCs.

V. CONCLUSION

Photonic networks and software-defined networks are two promising technologies to improve the performance and scalability of on-chip networks. However, these technologies do not solve the problem on their own. In this paper, an efficient software-defined photonic network on chip SD-PNoC is proposed. The proposed architecture and hierarchal communication protocols among the proposed layers have been designed to reduce the complexity growth of the SDN control plane and utilize the advantage of electronic short-distance communication and photonic long-distance communication. In addition, A Software-defined management orchestrator SDMO with different management policies for the processing plane has been applied considering the proposed addressing schemes. Nevertheless, these software management policies have been used to apply more control over path selectivity, and produce better on-chip resource utilization. The simulations for proposed SD-PNoC architecture and SDMO with policies tested in different conditions. The results showed high stability with a remarkable enhancement in the processing speed, latency, and power consumption.

REFERENCES

[1] A. S. Cassidy, J. Sawada, P. Merolla, J. Arthur, R. Alvarez-lcaze, F. Akopyan, B. Jackson, and D. Modha, "TrueNorth: A high-performance, low-power neurosynaptic processor for multi-sensory perception, action, and cognition," in *Proc. IBM Res.*, Mar. 2016, pp. 341–344. [Online]. Available: https://pdfs.semanticscholar.org/721b/270b7a92b60ee346a494b79dc28b419bac73.pdf

[2] M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018, doi: 10.1109/MM.2018.112130359.

[3] Q. Cheng, M. Glick, and K. Bergman, *Optical Interconnection Networks for High-Performance Systems*. Amsterdam, The Netherlands: Elsevier, 2019.

[4] C. Huang, A. Jha, T. F. de Lima, A. N. Tait, B. J. Shastri, and P. R. Prucnal, "On-chip programmable nonlinear optical signal processor and its applications," *IEEE J. Sel. Topics Quantum Electron.*, vol. 27, no. 2, pp. 1–11, Mar. 2021, doi: 10.1109/JSTQE.2020.2998073.

[5] K. Feng, Y. Ye, and J. Xu, "A formal study on topology and floorplan characteristics of mesh and torus-based optical networks-on-chip," *Microprocessors Microsyst.*, vol. 37, no. 8, pp. 941–952, Nov. 2013, doi: 10.1016/j.micpro.2012.06.010.

[6] T. Zhou and H. Jia, "Method to optimize optical switch topology for photonic network-on-chip," *Opt. Commun.*, vol. 413, pp. 230–235, Apr. 2018, doi: 10.1016/j.optcom.2017.12.062.

[7] S. Werner, J. Navaridas, and M. Luján, "Efficient sharing of optical resources in low-power optical networks-on-chip," *J. Opt. Commun. Netw.*, vol. 9, no. 5, pp. 364–374, 2017, doi: 10.1364/JOCN.9.000364.

[8] U. Iru, K. Edvhg, Q. Vwhpv, F. F. E. Rswlfdo, R. Q. Vlolfrq, and S. Lqwhusrvhuv, "323675*d*5*rexvw*0*rgxodu*2*swlfdo*1*r*& ufklwhfwxuh iru &klsohw edvhg,' qwhjudwhg 6 vwhpv," Tech. Rep., 2020, pp. 1456–1461.

[9] L. Guo, W. Hou, and P. Guo, "Designs of 3D mesh and torus optical network-on-chips: Topology, optical router and routing module," *China Commun.*, vol. 14, no. 5, pp. 17–29, May 2017, doi: 10.1109/CC.2017.7942191.

[10] L. Guo, Y. Ge, W. Hou, P. Guo, Q. Cai, and J. Wu, "A novel IP-core mapping algorithm in reliable 3D optical network-on-chips," *Opt. Switching Netw.*, vol. 27, pp. 50–57, Jan. 2018, doi: 10.1016/j.osn.2017.08.001.

[11] A. Ben Ahmed and A. Ben Abdallah, "Graceful deadlock-free fault-tolerant routing algorithm for 3D network-on-chip architectures," *J. Parallel Distrib. Comput.*, vol. 74, no. 4, pp. 2229–2240, Apr. 2014, doi: 10.1016/j.jpdc.2014.01.002.

[12] S. J. B. Yoo, "2D and 3D heterogeneous photonic integrated circuits," in *Proc. Smart Photonic Optoelectron. Integr. Circuits XVI*, Mar. 2014, pp. 16–18, doi: 10.1117/12.2047502.

[13] M. Kennedy and A. K. Kodi, "CLAP-NET: Bandwidth adaptive optical crossbar architecture," *J. Parallel Distrib. Comput.*, vol. 100, pp. 130–139, Feb. 2017, doi: 10.1016/j.jpdc.2016.05.004.

[14] H. Falahati, S. Koohi, and S. Hessabi, "Application-based dynamic reconfiguration in optical network-on-chip," *Comput. Electr. Eng.*, vol. 45, pp. 417–429, Jul. 2015, doi: 10.1016/j.compeleceng.2015.04.019.

[15] J. H. Lee, M. S. Kim, and T. H. Han, "Insertion loss-aware routing analysis and optimization for a fat-tree-based optical network-on-chip," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 3, pp. 559–572, Mar. 2018, doi: 10.1109/TCAD.2017.2712670.

[16] X. Xue, F. Yan, K. Prifti, F. Wang, B. Pan, X. Guo, S. Zhang, and N. Calabretta, "ROTOS: A reconfigurable and cost-effective architecture for high-performance optical data center networks," *J. Lightw. Technol.*, vol. 38, no. 13, pp. 3485–3494, Jul. 1, 2020, doi: 10.1109/JLT.2020.3002735.

[17] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking?: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[18] M. Ruaro, H. M. Medina, A. M. Amory, and F. G. Moraes, "Software-defined networking architecture for NoC-based many-cores," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5, doi: 10.1109/ISCAS.2018.8351830.

[19] M. Ruaro and F. G. Moraes, "Multiple-objective management based on a distributed SDN architecture for many-cores," in *Proc. 33rd Symp. Integr. Circuits Syst. Design (SBCCI)*, Aug. 2020, pp. 1–6, doi: 10.1109/SBCCI50935.2020.9189905.

[20] M. Ruaro, L. L. Caimi, and F. G. Moraes, "A systemic and secure SDN framework for NoC-based many-cores," *IEEE Access*, vol. 8, pp. 105997–106008, 2020, doi: 10.1109/ACCESS.2020.3000457.

[21] R. Sandoval-Arechiga, J. L. Vazquez-Avila, R. Parra-Michel, J. Flores-Troncoso, and S. Ibarra-Delgado, "Shifting the network-on-chip paradigm towards a software defined network architecture," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Dec. 2015, pp. 869–870, doi: 10.1109/CSCI.2015.45.

[22] F. L. D. Nunes and M. E. Kreutz, "Using SDN strategies to improve resource management on a NoC," in *Proc. IFIP/IEEE 27th Int. Conf. Very Large Scale Integr. (VLSI-SoC)*, Oct. 2019, pp. 224–225, doi: 10.1109/VLSI-SoC.2019.8920321.

[23] L. Cong, W. Wen, and W. Zhiying, "A configurable, programmable and software-defined network on chip," in *Proc. IEEE Workshop Adv. Res. Technol. Ind. Appl. (WARTIA)*, Sep. 2014, pp. 813–816, doi: 10.1109/WARTIA.2014.6976396.

[24] J. Wang, M. Zhu, C. Peng, L. Zhou, Y. Qian, and W. Dou, "Software-defined photonic network-on-chip," in *Proc. 3rd Int. Conf. e-Technol. Netw. Develop. (ICeND)*, Apr. 2014, pp. 127–130, doi: 10.1109/ICeND.2014.6991365.

[25] J. Wang, Q. Feng, Y. Wang, Q. Dou, and W. Dou, "A hybrid hierarchical software-defined photonic on-chip network," in *Proc. Int. Conf. Netw. Inf. Syst. Comput., ICNISC*, Apr. 2016, pp. 133–137, doi: 10.1109/ICNISC.2016.36.

[26] I.-D. Salvador, S.-A. Remberto, M. Brox, and M. A. Ortiz, "Software defined network controller: A neat solution administration for reconfigurable multi-core NoC," in *Proc. Int. Conf. ReConFigurable Comput. FPGAs (ReConFig)*, Dec. 2017, pp. 1–4, doi: 10.1109/RECONFIG.2017.8279821.

[27] R. Sandoval-Arechiga, R. Parra-Michel, J. L. Vazquez-Avila, J. Flores-Troncoso, and S. Ibarra-Delgado, "Software defined networks-on-chip for multi/many-core systems: A performance evaluation," in *Proc. Symp. Archit. Netw. Commun. Syst.*, Mar. 2016, pp. 129–130.

[28] R. S. Silva, E. Kreutz, and M. Magalh, "Communication latency evaluation on a software-defined network-on-chip," in *Proc. 9th Brazilian Symp. Comput. Syst. Eng. (SBESC)*, Nov. 2019, pp. 1–7.

[29] A. Scionti, S. Mazumdar, and A. Portero, "Software defined network-on-chip for scalable CMPs," in *Proc. Int. Conf. High Perform. Comput. Simul. (HPCS)*, Jul. 2016, pp. 112–115.

[30] H. Temuçin and K. M. Imre, "Scheduling computation and communication on a software-defined photonic network-on-chip architecture for high-performance real-time systems," *J. Syst. Archit.*, vol. 90, pp. 54–71, Oct. 2018, doi: 10.1016/j.sysarc.2018.07.007.

[31] G. Hendry. (2011). *Architectures and Design Automation for Photonic Networks On-Chip*. [Online]. Available: http://academiccommons.columbia.edu/catalog/ac:131950

[32] H. Li, H. Gu, Y. Yang, and X. Yu, "A hybrid packet-circuit switched router for optical network on chip," *Comput. Electr. Eng.*, vol. 39, no. 7, pp. 2197–2206, Oct. 2013, doi: 10.1016/j.compeleceng.2013.08.006.

[33] K. A. Helal, S. Attia, T. Ismail, and H. Mostafa, "Priority-select arbiter: An efficient round-robin arbiter," in *Proc. IEEE 13th Int. New Circuits Syst. Conf. (NEWCAS)*, Jun. 2015, pp. 3–6.

[34] S. Manual, "Simulation manual," in *Proc. OMNeT++ Simulation Man.*, 2016, pp. 3–6.

[35] P. U. Manual, G. Hendry, and J. Chan, "PhoenixSim 1.0 user manual," Tech. Rep., 2011, pp. 1–60.

DOAA A. HAMDI received the B.Sc. degree from the Department of Electrical and Computer Engineering, Menoufia University, Egypt, in 2014. She is currently pursuing the M.Sc. degree in computer and systems engineering from the Department of Electrical Engineering, Faculty of Engineering, Ain Shams University, Cairo, Egypt. She was a Teaching Assistant at the Faculty of Computers and Information Technology, National Egyptian E-Learning University (EELU), Egypt, for four years. She has a well technical experience and academic background in the field of computer networks. She is currently a Teaching Assistant in computer and systems engineering with the Faculty of Engineering, Badr University in Cairo (BUC), Egypt.

SAMY GHONIEMY received the B.Sc. degree in communications and computer engineering and the M.Sc. and Ph.D. degrees in systems and computer engineering from Carleton University, Canada. He is currently a Professor of computer systems and the Vice Dean for postgraduate studies and research, Faculty of Informatics and Computer Science, The British University in Egypt (BUE). He also has extensive professional and consultancy experience in ICT. He worked and now a Consultant for national and multinational ICT private companies. He authored or coauthored more than 100 research works, which has been published in well-known international refereed journals such as IEEE, Springer, and Elsevier, which are indexed by ISI, SCOPUS, Science Direct, Google Scholar, and so on. He received local and international grants for various research and projects in artificial intelligence areas, such as cognitive informatics, health surveillance, and health-care services.

YASSER DAKROURY received the B.Sc. degree from the Faculty of Engineering, Ain Shams University, in 1982, and the M.Sc. and Ph.D. degrees in computer and systems engineering from the École Centrale de Nantes, France, in 1987 and 1991, respectively. He was a former President of National Egyptian E-Learning University (EELU), Egypt. He is currently a Professor of computer and systems engineering, Ain Shams University, Cairo, Egypt, and the Vice President for International Relations, Sinai University, Egypt. He has authored or coauthored more than 100 research works, which has been published in well-known international refereed journals such as IEEE, Springer, and Elsevier, that are indexed by ISI, SCOPUS, Science Direct, Google Scholar, and so on.

MOHAMMED A. SOBH received the B.Sc. and M.Sc. degrees and the Ph.D. degree in computer and systems engineering from the Faculty of Engineering, Ain Shams University, Cairo, Egypt, in 1996, 2001, and 2007, respectively. He is currently an Associate Professor of computer and systems engineering with Ain Shams University. He has authored or coauthored many research works, which has been published in well-known international refereed journals and conferences such as IEEE, Springer, and Elsevier, that are indexed by ISI, SCOPUS, and Science Direct.