Hindawi
Security and Communication Networks
Volume 2023, Article ID 9812080, 1 page
https://doi.org/10.1155/2023/9812080

Retraction

Retracted: Workload-Aware WiNoC Design with Intelligent Reconfigurable Wireless Interface

Security and Communication Networks

Received 5 December 2023; Accepted 5 December 2023; Published 6 December 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] Q. Wang, Y. Ouyang, Z. Huang, and H. Liang, "Workload-Aware WiNoC Design with Intelligent Reconfigurable Wireless Interface," *Security and Communication Networks*, vol. 2023, Article ID 9519044, 14 pages, 2023.

Hindawi
Security and Communication Networks
Volume 2023, Article ID 9519044, 14 pages
https://doi.org/10.1155/2023/9519044

Research Article

Workload-Aware WiNoC Design with Intelligent Reconfigurable Wireless Interface

Qi Wang,[1] **Yiming Ouyang**,[2] **Zhengfeng Huang**,[1] **and Huaguo Liang**[1]

[1]*School of Electronic Science and Applied Physics, Hefei University of Technology, Hefei 230009, China*
[2]*School of Computer and Information, Hefei University of Technology, Hefei 230009, China*

Correspondence should be addressed to Yiming Ouyang; 2018010128@mail.hfut.edu.cn

Received 13 June 2022; Revised 30 July 2022; Accepted 10 August 2022; Published 9 May 2023

Academic Editor: Mohammad Ayoub Khan

By introducing wireless interfaces in conventional wired routers or hubs, wireless network-on-chip (WiNoC) is proposed to relieve congestion pressure from high volume inter-subnet data transmission. Generally, processing elements on chip receive input data and return feedback through network interface, and data transmission function in Network-on-Chip (NoC) is completed by routers. Hubs equipped with wireless interface are fixed to certain wired routers. While wireless channels may not be fully utilized due to unbalanced workload and constant hub-router connection, e.g., certain nodes processing excess inter-subnet data traffic are far away from hubs. In this paper, we proposed a workload-aware WiNoC design with intelligent reconfigurable wireless interface to improve wireless resources utilization and mitigate congestion. Through multidimensional analysis of traffic flow, a 4-layer neural network is trained offline and applied to analyze workload in each tile, and return three most potential tiles for wireless interface reconfiguration to fully utilize wireless channel and lowing latency. We also implement a historical traffic information-based reconfigurable scheme for comparation. Evaluation results show that in an 8 × 8 hybrid mesh topology, the proposed scheme can achieve 10%–16% reduction in network latency and 5%–11% increment in network throughput compared with fixed-link hub-node connection scheme under several mixed traffic patterns.

1. Introduction

Network-on-chip (NoC) [1] has been widely used in multicore systems as paradigm communication architecture. The inherent characteristics of NoC such as high bandwidth scalability, high throughput, and low latency endow it with great advantages confronting common BUS structure. While increasing the number of integrated cores, network topology enlarges inevitably, which is an unfriendly application with a mass of multihop long-distance data transmission. By introducing wireless interface in conventional 5-stage routers or hubs, wireless network-on-chip (WiNoC) is proposed to relieve congestion pressure from high volume inter-subnet data transmission. With appropriate settings on wireless transmission frequency, the number of cycles required for a flit transmission through wireless channel can be the same or double to that through multihops wired transmission.

Wireless transmission on chip can be achieved by equipping a wired router with a wireless transceiver and a zig-zag antenna through millimeter wave channels, named wireless node.

Generally, in each subnet, several routers are selected to be attached with a hub using wired links [2]. Intra-subnet data is transmitted though hubs using wireless links so that the high latency caused by multihop long-range data transmission can be relieved. Restricted to resource limitations on chip, wireless interfaces (WI) cannot be placed indefinitely. Hence, in each subnet, data packets requiring wireless transmission must be routed to nodes attached to a wireless hub as intermediate transit points, then through crossbar arbitration for wireless access. Essentially this mechanism is similar to time division multiple access (TDMA), trading the latency for sequential wireless transmission.

Considering scarcity of wireless resources, excess data would gather around wireless nodes, generating congestion region or even hotspots. We can use intelligent algorithms and analyze traffic patterns in the network to determine which wired nodes to connect to. However, traffic is variable and if a fixed connection is used, if traffic changes, nodes with high demand for wireless transmission may not be able to use the wireless resources directly and will need to be wired over multiple hops to reach the attached node. This initial set of connections is likely to be a significant waste of wireless resources. An intelligent reconfigure wireless interface may help to solve this problem.

Traffic flow is the critical information needed to supervise system working conditions. It can be used to support congestion mitigation, faulty node avoidance, and reconfiguration mechanism [3–5]. The traffic information can be analyzed in several aspects, e.g., data volume between two communication pairs, wireless packets volume in a certain subnet, and buffer utilization rate in one node. Different metrics can reflect different characteristics of traffic flow. To find the fittest hub-node connection, one single metric for data flow is not enough. Buffer utilization or number of free virtual channels in router is the commonly used metric to reflect traffic flow. When buffer utilization keeps high in one region, we can estimate the workload of this region which exceeds the processing ability. While the traffic flow variation is complicated, the single dimension of buffer utilization information does not provide a complete picture of congestion in the network. The congestion may come from an excessive amount of data routing to the node as a source-destination node, the node carrying too much traffic as an intermediate node or packets that need to be transmitted across subnets using wireless channels are aggregated around the wireless node because of limited wireless resources. More metrics beside buffer utilization need to be considered to comprehensively analyze network conditions.

The rapid development of artificial neural network technology and the increase in computing power of end devices have led to the widespread implementation of related applications over the past few years. Among them, CNN [6] has a significant role in promoting popular fields such as face recognition, autonomous driving, and recommendation algorithms [7, 8]. Convolutional neural networks take a subject such as a picture as signal input, set different filters to multiply and add different combinations of the input signal, advance layer by layer, extract features, and then calculate the likelihood of the corresponding target through fully connected layers and activation functions. A small neural network module is built into the network-on-a-chip system, which is trained offline using the application traffic on which the network is likely to operate, and then deployed on-chip, so that the system operation can be analyzed online and reconfigured intelligently and autonomously to increase the network performance.

In this paper, we proposed a workload-aware WiNoc design with intelligent reconfigurable wireless interface. Through multidimensional analysis of traffic flow, a 4-layer neural network is trained offline and applied to analyze workload in each tile, return three most potential tile for wireless interface reconfiguration to fully utilize wireless channel and lowing latency. The contributions of this work can be summarized as follows:

(1) Proposes a CNN-based intelligent reconfigurable wireless interface with its hardware implementation and prioritize the wireless access according to the reconfigurable algorithm feedback.

(2) Through intelligent node placement and traffic flow arrangements, the additional latency overhead originated from the reconfigurable module can be minimized.

(3) Implements a reconfigurable mechanism based on historical traffic information for comparation, and evaluates the proposed scheme with several wireless/wired networks.

The rest of the paper is organized as follows: Section 2 summarizes some related work in adaptable wireless NoC design and intelligent NoC schemes in recent years. Section 3 describes preliminaries on network architecture and dataset preparation about the proposed scheme. Section 4 elaborates the adopted neural network topology and ANN process module placement with the convolution task allocation. Section 5 is the design of intelligent reconfigurable wireless interface with its hardware implementation and corresponding modification for wired routers. Evaluation environment and results analysis are discussed in Section 6. Finally, Section 7 concludes this work and points out our future work.

2. Related Works

As one promising solution to resource-limited application-driven on-chip communication, intelligent or reconfigurable wireless network on chip has been proposed in many studies. While research combining both reconfigurability and autonomous control ability remain insufficient [9]. Systematically one illustrates that establishing wireless connections between remote nodes or subnets can improve network performance and increase system energy efficiency without introducing significant hardware overhead. However, if hub-node connections are fixed, this topology is hardly an optimal solution once task immigrates and traffic pattern changes. This has led to many studies on hub connection point selection, mac mechanisms, and routing algorithms. Researchers generally adjust the traffic locally or globally, using adaptive traffic aware methods to reconfigure routers or topologies to alleviate port-level congestion, to avoid congestion, and improve the reconfigurable capability of chip communication. While the value of multidimension analysis of data flow is not well explored. As shown in Figure 1, a hybrid Cmesh wireless network-on-chip architecture is proposed, in which four PE cores are connected to one wireless router, routers transmit messages using a wired line within each subnet, using three fixed wireless channels, and one adaptive wireless channel cross-subnet, so that data packet can be routed to destination within one hop. The

adaptive wireless channels can be assigned to any other subnet according to workload, and significantly improve system performance. While to achieve this design, each router must be equipped with at least 6 ports and 4 wireless transceivers, and through several arbitration processes can data be routed to destination, leading to high area overhead and low buffer utilization. Reference [11] proposed a flexible NoC architecture-AdaptNoC, in which different size and topology of subnets can be assigned to parallel applications, to improve the energy efficiency. Reference [12] proposed a single-cycle asynchronous multihop design to reconfigure data path from the source node to the destination node in one cycle. Reference [13] focuses on the fault-tolerant communication, in which data packet can be routed through bypass to avoid faulty nodes. Reference [14] proposed a low power neural network task scheduler, by predicting video content and return workload analysis, tasks can be remapped in NoC. Due to training process and dataset, this scheduler can only deal with video tasks. An online reinforcement learning-based adaptive bandwidth reallocation and cores remapping method is proposed in Reference [15] to address on-chip resource management problem. Reference [16] designs an intelligent NoC to regulate bandwidth according to workload in four Feature Extraction Clusters which is predicted by fuzzy rules applied on the variance of pixel intensities, the peak counts from the vertical and horizontal projections of a tile. These works are application-specific, focusing on certain video or figure processing tasks. We are pursuing a holistic solution for the system to automatically analyze workload and reallocate hub-cores connection when dealing with common tasks.

3. Preliminaries

3.1. Wireless NoC Architecture. Millimeter-wave can be adopted in NoC as a wireless medium to mitigate latency and power overhead from long-range multihop on-chip communications [2]. Wireless communication can be achieved by introducing wireless interface into wired routers or hubs. The ideal implementation of wireless communication is to endow every node with wireless access so that each communication pair can transmit data within one wireless hop distance. While considering the limitation of wireless channels and restriction of area and power overhead, equipping wireless interfaces for all nodes is not practical. Meanwhile, except for high injection load application, buffer utilization rate maintains a low level in pure wireless network, leading to a waste of wireless resources in respect of leakage power consumption. Hence, we proposed a reconfigurable wireless hub, connected to three nodes inside subnet periodically. The traditional wired router does not need to be redesigned, but rather a selector is used to determine which nodes are given the access to wireless transmission. Typically, the WI placement is chosen to be fixed to certain nodes inside the subnet so that other nodes in the subnet have the shortest distance to reach the wireless node. However, this choice is not optimal when considering different traffic patterns, especially for complicated traffic flow during the task immigration process.

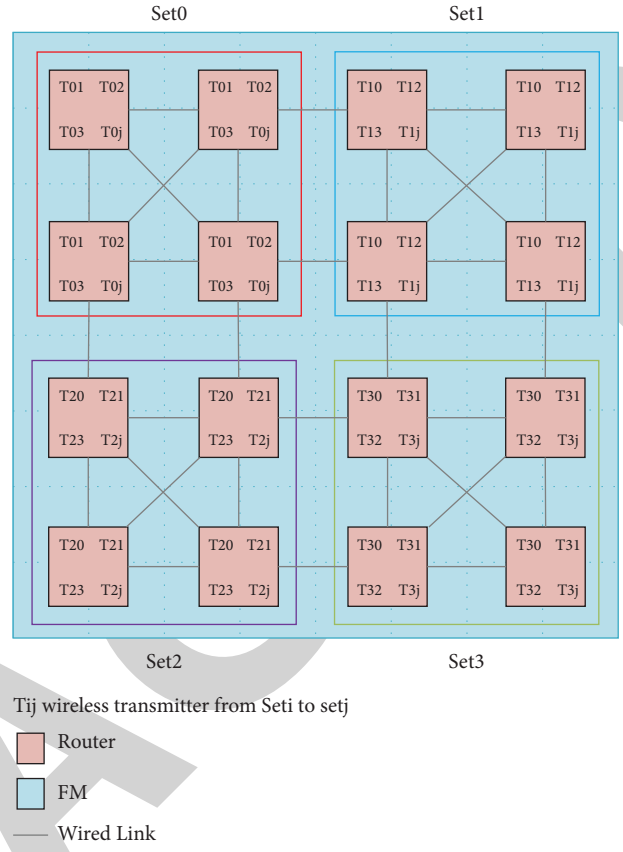

Figure 1: A-WiNoc adaptable wireless architecture [10].

As shown in Figure 2, in the proposed topology graph, 64 PE cores are organized in mesh architecture and divided into 4 rectangular clusters. Reference [17] demonstrates that for a 64 cores NoC, the optimum arrangement for WI number is 12. Hence, we assign one hub with three wireless transceivers and receivers in each subnet, and those transceivers work in exclusive frequencies. Each hub connects all routers inside one subnet by a wired line, while only three connections are selected to be active. For data transmission integrity, we use the token-packet medium access mechanism. Until one data packet is fully transmitted, the token can be released and transferred.

3.2. Dataset Building. The main factor that affects network performance when the network workload does not reach its designed saturation capacity is congestion. Congestion usually arises in areas where data flows are slow at certain nodes. For a particular node, congestion arises for three main reasons. An excessive amount of data routing to the node as a source-destination node; the node is carrying too much traffic as an intermediate node; and packets that need to be transmitted across subnets using wireless channels are aggregated around the wireless node because of limited wireless resources. In this paper, we focus on the third aspect to alleviate congestion and improve network performance. Under each traffic pattern, after warming up the network for 1000 cycles, we sample the three metrics for each 100 cycles. Then increase the data injection rate to

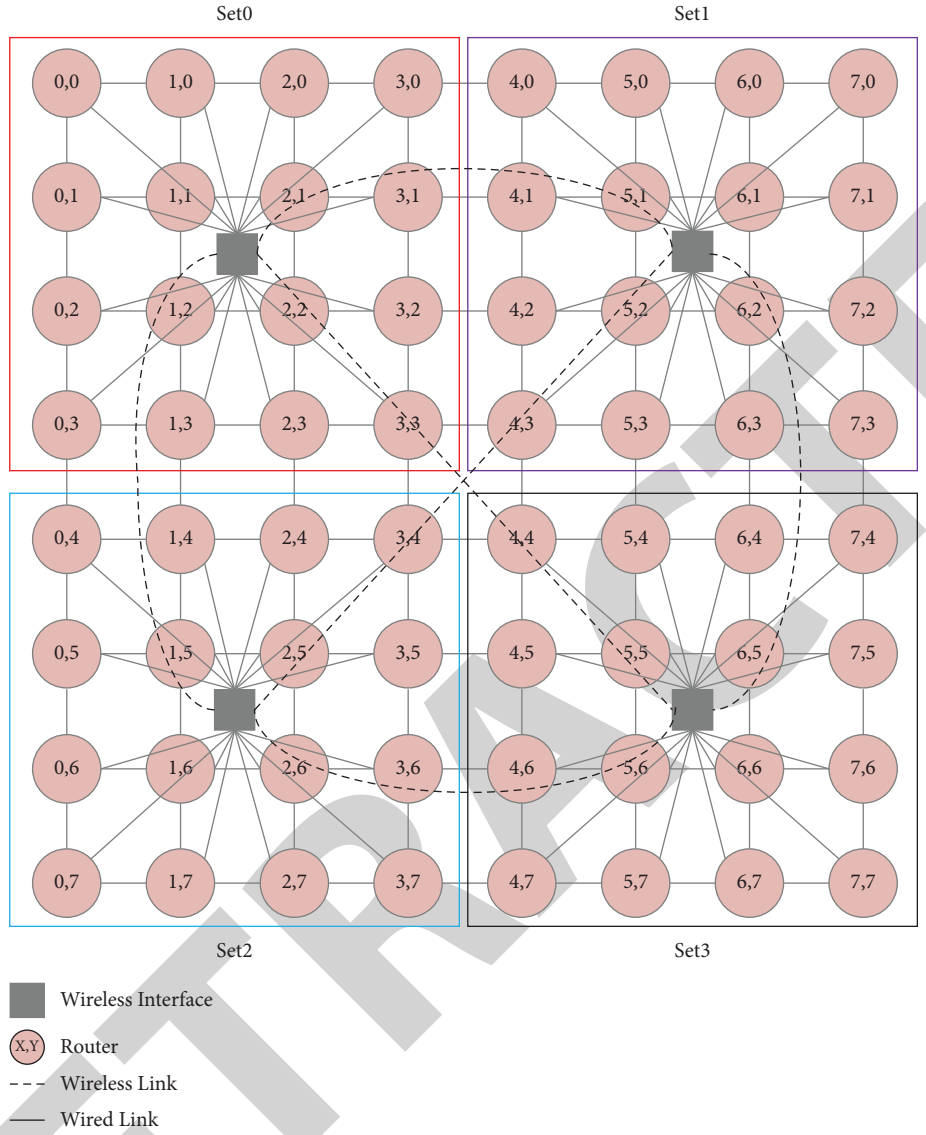

Figure 2: Wireless NoC overall architecture.

boost the network load and resample. Afterward, we change the traffic pattern and repeat the process. The simulated annealing algorithm is adopted to find the best hub connection for each traffic pattern, which is used as the label for the training data, so the input part of the final training data set raw data is an 8×8 matrix, each point in the matrix is a vector of depth 3, and the label is 4 vectors of depth 16, representing the location of the hub-node connection. The goal of the hub connection point selection is to reduce the overall transmission distance and make the wireless transmission efficient considering the communication frequency. We use the simulated annealing WI placement algorithm in Reference [18] to maximize the metric by considering the communication frequencies for different traffic patterns, as three WI's are arranged in each subnet, which in most cases also satisfies the requirement of distance (7 nm) between WI's [9].

4. Neural Network Architecture and ANN Module Placement

As claimed in preliminary, to reflect the nature of the data stream, three metrics, packet duration, throughput, and cross-subnet communication frequency, are chosen as training data. The dataset is arranged in an $8 \times 8 \times 3$ matrix structure. If a fully connected network is adopted, the input layers would have to be tiled, which would make the input layers too large. While considering the limitation of on-chip resources, the hardware overhead and computing complexity should be minimized as far as possible on the premise of effect guarantee. A $2 \times 2 \times 3$ convolutional kernel with a step size of 1 is designed to convolve the input matrix to obtain a 7×7 output matrix, which is then fully connected to the 64 hidden layer neurons, and the output layer uses a sigmoid activation function to output 64 probabilities.

These 64 bits of probability information are divided into four 16 bit vectors and sent to the reconfigurable control module of the corresponding subnetwork, and the larger three bits of each vector are selected as the connection points. The following diagram shows the neural network framework we designed, as shown in Figure 3.

The reconfiguration mechanism is usually activated when the network congestion deteriorates. Data packets carrying network condition information required by the reconfiguration algorithm need to be transmitted to the intelligent nodes in the form of data packets, and these additional data packets can also increase the congestion of the network. The main part of the data transmission for the convolutional neural network determination mechanism in this paper comes from the three-dimensional traffic information collection required for the convolutional process, which accounts for more than 70% of the data collection and transmission for the reconfiguration algorithm. The input data of the fully connected layer of inference process has already been much compressed through three-dimensional convolution process, and the weights can be stored in ANN nodes. We calculate the forward propagated data in parallel in a distributed manner through a rational arrangement of intelligent nodes, with each intelligent node processing information from the local subnetwork, reducing the total amount of information needed to be transmitted. This operation benefits the network latency by reducing the sequential time for packet generation and improving the efficiency of the intelligent reconstruction algorithm.

The four nodes at the center of the entire network act as intelligent nodes, completing the aggregation of input data, calculating, and transmitting the resultant functions. The nodes complete the convolution calculation of the local subs before interacting with the information, reducing the amount of information interacted with, and reducing the communication pressure. It is also important to note that cross-position convolution at the center of the network requires information across subnets and requires special arrangements. As shown in Figure 4(a), the four intelligent nodes ABCD collect traffic information of all nodes in the belonging subnet for convolutional calculation. The amount of input data after the $2 \times 2 \times 3$ filter convolution has been greatly reduced, relieving the pressure for the subsequent interactive transmission. The red boxes mark the 13 convolutional objects that are not included. Figure 4(b) shows the abstracted convolutional operation objects. The blue boxes mark the allocated objects according to the intelligent nodes, and the left ones are unassigned. In Figure 4(c), the unassigned convolution objects in Figure 4(b) are arranged. The convolution at the network center is performed by the intelligent node A. The additional assigned convolutional tasks in the subnetwork and outwards require the relevant nodes to send traffic information to the corresponding intelligent nodes. The allocated raw data was convolved to produce a vector of depth 49, which was shared and stored in the four intelligent nodes. The fully connected and output layers of the network have 64 convolutional kernels, and each intelligent node is responsible for 16 convolutional kernels. Each time a 64 bit output is generated by multiplication and addition, it is shared among the four intelligent nodes. The final output is one-hot encoded and sent by each intelligent node to the control component of the reconfigurable module for the next reconfiguration step. The neural network we use consists of four layers: an input layer, a convolutional layer, a fully connected layer, and an output layer.

The five-layer network has limited accuracy gains over the four-layer network, less than 4.7% increment of the total correct output, but the network transmission, data processing tasks, and parameter storage requirements are greatly increased, while deepening the network requires methods such as dropout to avoid overfitting, so the four-layer network is a better trade-off.

5. Regonfigurable Wireless Interface Design and Arbitration Logic

5.1. Wireless Interface Design. Implementing reconfigurable configurations of the wireless interface allows for each node to be equipped with a wireless transceiver that generates the wireless topology by controlling the power supply and adapting the channel. However, this solution obviously adds significant hardware overhead, with insufficient utilization wireless resources and high proportion of leakage power.

We propose the following wireless hub architecture, where three wireless transceivers and their corresponding buffer, including buffer_to_tile and buffer_from_tile, are integrated on the hub of each subnet and can be selectively linked to the three routers in the interior of the subnet. Figure 5 shows the allocation of the wireless transceivers (the red boxes are marked for the hub, *T* for the transceiver, and *R* for the receiver), each transmitter in the hub corresponds to a receiver in the other subnet, each receiver corresponds to a transmitter in the other subnet, and the number behind the transceiver corresponds to each group of transceivers in the pair. In order to avoid adjacent interference, we need to set up channels for each fixed link pair, requiring a total of 12 nonoverlapped wireless channels. Bandwidth of 42 GHz, corresponding to a 42 Gbps data rate for our binary modulation. Each hub assigns transceivers to the three routers of the link via a set of 3×3 crossover switches. Packets competing for the same output port are given transmission access based on priority through an arbiter.

Figure 6 shows the architecture of the wireless HUB. The on-chip routers are coded as Ri ($i = 0, 1, \ldots, 14, 15$), in which R marks the belonging subset, and i marks the exact location of router. The control module receives a 16 bits selection signal from the intelligent nodes and selects three routers to connect to the buffer interface via three 16-select 1 MUXs. For instance, if the control module receives reconfigure instruction 0010_1001_1110 (from high to low bit), link $R2$ to the buffer corresponding to the upper MUX, link R9 to the buffer corresponding to the middle MUX, and link $R14$ to the buffer corresponding to the lower MUX. After the reconnection is completed, the priority is set in Switch Arbitration according to reconfigure instruction. The feedback from the intelligent algorithm is put in order before one-hot coding, the node with higher connection

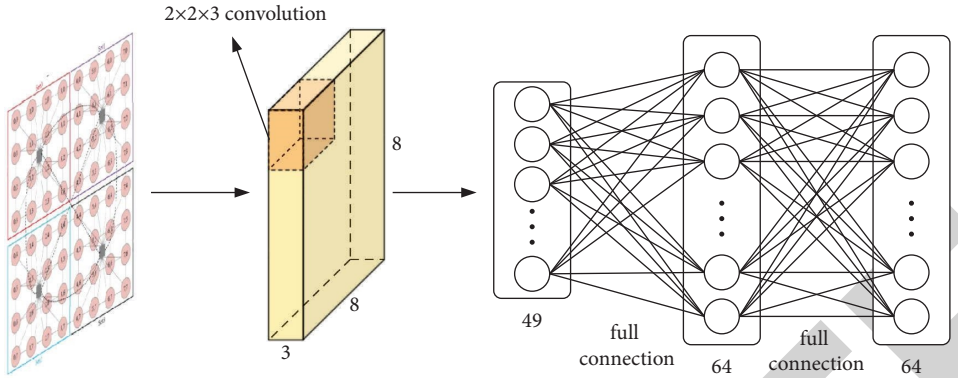

Figure 3: CNN architecture.

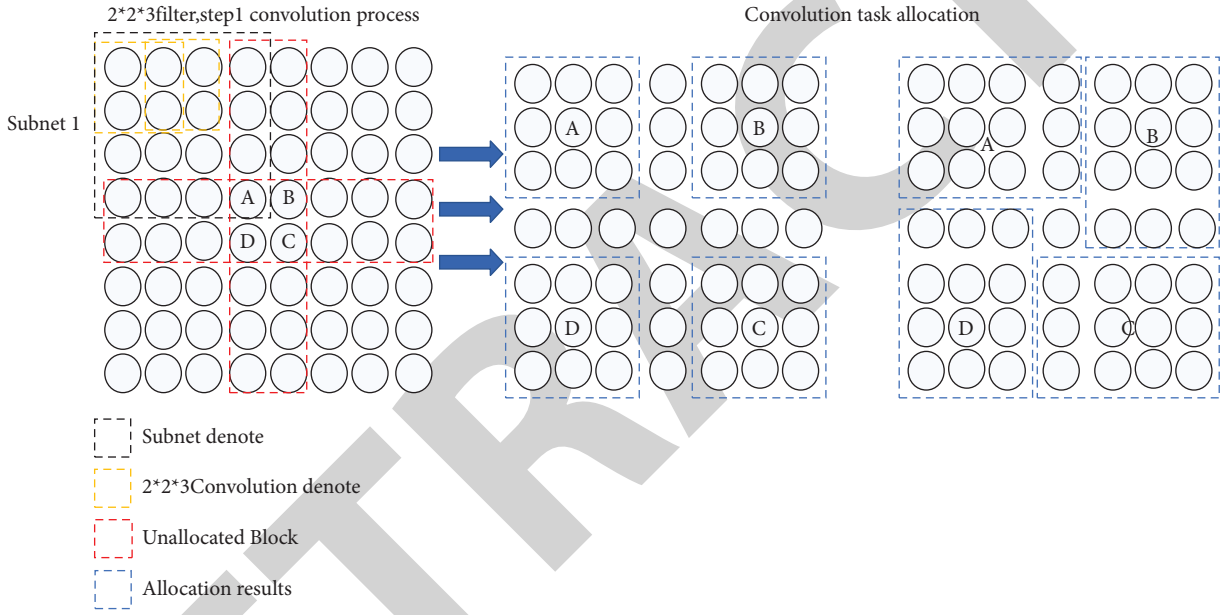

Figure 4: ANN module placement and convolution task allocation.

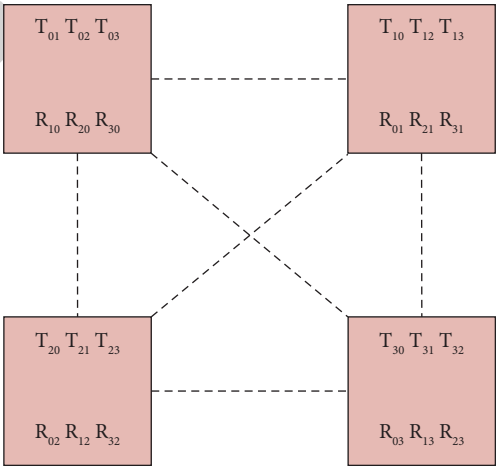

Figure 5: Wireless transceivers allocation.

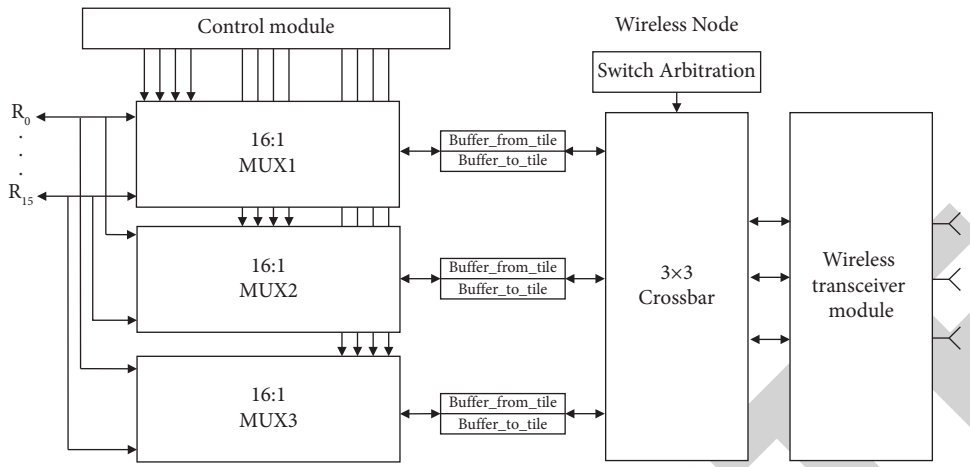

Figure 6: Wireless interface design for intelligent reconfiguration.

opportunities is put in high bit. When data packets from connected routers compete for the same output transceiver, packets from higher priority router are transmitted first.

The control module receives the 12-bit reconfiguration signal from the intelligent nodes output and decouples it into three 4-bit signals that go to the allocator module for processing. The main reason for setting the allocator module is to prevent redundant reconfiguration because the reconfiguration signal may instruct the control module to relink the hub with the same router in the last sampling window. For instance, the OUT_*i* of the previous TS is 0001, 0010, and 0100 in sequence, and the MUX_*i* of the current calculation is 0010, 0111, and 0001 in sequence. If a rigid remapping method is used, OUT_*i*_new will be relinked to 0010, 0111, and 0001 in sequence, then all three MUX in Figure 6 will have to be disconnected and relinked. If a dynamic mapping with the proposed control module is used, when OUT_*i*_new needs to be relinked to 0001, 0010, and 0111 in turn, only one of the MUX needs to be reconfigured.

Available_*i* ($i = 1, 2, 3$) is a one-bit register to approve the relink considering the data packets in transmission which may be interrupted. Because the minimum flow control unit used in this solution is flit, when reconfiguring the MUX, it is possible that there may be cases where the packet has not been transferred completely, e.g., only the header microchip of the packet has been transferred. If the MUX is reconstructed at this point, the rest of the packet will be lost. Therefore, we need to ensure that when the MUX is reconfigured, it must have already transmitted the entire packet. This can be achieved by controlling the Buffer_to_tile and Buffer_from_tile status. When Available_1 is 1, the MUX in Figure 7 selects OUT_1_new for Control_signal_for_MUX1, then OUT_1 will be immediately set to OUT_1_new, then MUX1 in Figure 6 will select the router corresponding to Control_signal_for_MUX1, and finally MUX1 of the wireless node is reconfigured. When Available_1 is 0 then OUT_1 ($i = 1, 2, 3$) is selected. Available_i can be set to 1 when the following two conditions are both satisfied.

(1) The Buffer_to_tile corresponding to MUX*i* ($i = 1, 2, 3$) is empty or the buffer header is the head flit.

(2) The Buffer_from_tile corresponding to MUX*i* ($i = 1, 2, 3$) is empty or the end of the buffer is the tail flit.

All buffers are first-in, first-out queens. The Buffer_to_tile stores data packets received from the wireless interface. Rule 1 can guarantee that in the timing when hub-node connection is reconstructed, no packets are under transmission from WI to node. Since the buffer is either empty, or the header is the head flit. Otherwise, if certain packets are under transmission, flits in the buffer header should be body or tail flit. The Buffer_from_tile stores packets transmitted to the wireless interface. When it is empty, no packet is under transmission. When the end of the buffer is tail flit, we can infer that the data packet from the tile has been completely transmitted, otherwise it should be body or head flit.

5.2. On-Chip Router Design. Subject to the above conditions, when the mux is reconstructed, there are no half-transmitted packets in the buffers it is linked to. Therefore, it is guaranteed that packets passing through the mux will not be lost during reconfiguration. However, there still could be packet dropping in the receiver end. Let us consider a situation where packet P is injected into the network via router *R*1 at the TS-4 cycle. *R*1 belongs to Subnet1, and the destination node of P is router *R*2, which belongs to Set2. *R*1 and *R*2 are linked to the hubs in the belonging subnet, and the wired hop distance between them is greater than 2. P will be routed to the destination node through wireless communication. P then obtains a crossbar access at the TS-3 cycle and its head flit reaches the router directly connected to Buffer_from_tile of *R*1 at the TS-2 cycle, then both *R*1 and *R*2 are unselected wireless hub in their belonging subnet in the TS cycle, so the mux in router needs to be reconfigured. Because packet P is not fully routed to Buffer_from_tile during the TS cycle, the mux of *R*1 cannot be reconfigured immediately, while the mux of *R*2 is reconstructed in the TS cycle. So when packet P arrives at subnet2, there is no Buffer_to_tile corresponding

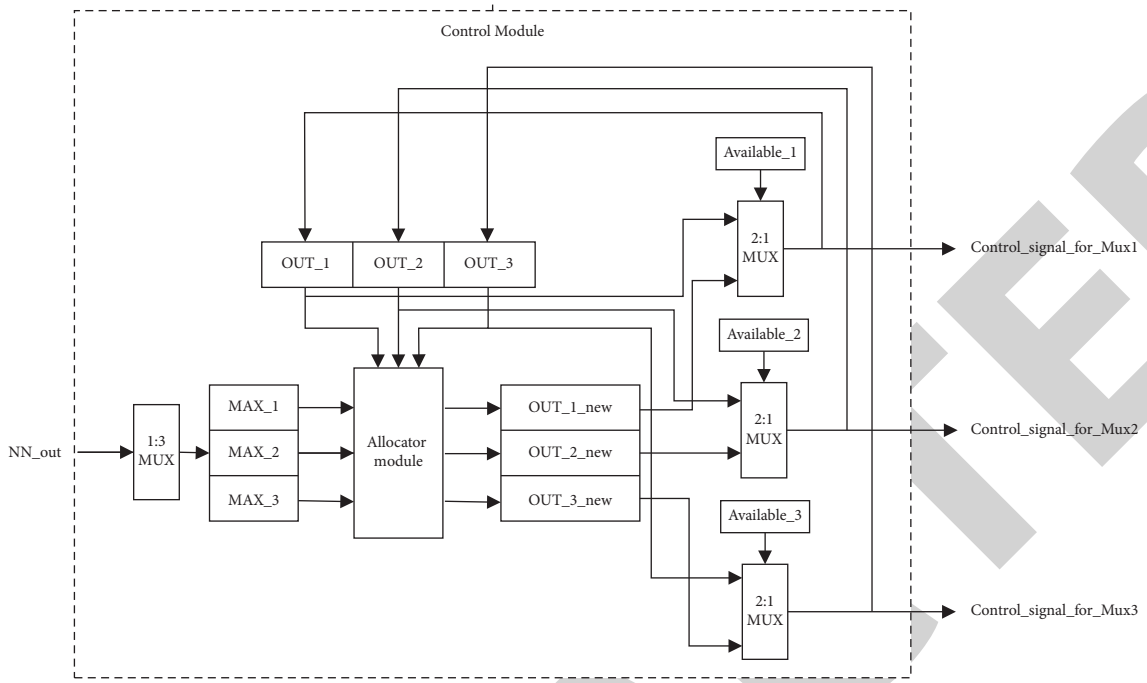

Figure 7: Control module for wireless interface.

to its destination node *R2*. Although the probability of this case is low, we need to redesign the router to totally solve it.

Figure.8 shows an on-chip router supporting reconfigurable wireless nodes, which adds a small number of components which are added to the traditional router, including a demux, a mux, and their corresponding control modules. In Figure 8, the Input_port_WI and Output_port_WI are the input and output ports to the wireless direction, which are only valid when the local router is selected by the wireless node of the belonging subnet.

In order to speed up the routing of wireless packets and avoid congestion at wireless nodes, a separate local port is set for packets coming from a wireless node whose destination node is the local router.

If the current router is connected to wireless hub, then Input_port_WI links to the Buffer_to_tile.

In Figure 8, there are two possible destinations for packets in the Buffer_to_tile, the local router, or another router in the same subnet.

When the destination node is the local router, DEMUX connects Input_port_WI and Output_port_Local2. When the destination node is another router, DEMUX connects Input_port_WI and local direction input buffer, Iutput_port_Local. To prevent packets injected into the local input buffer from disorder, only when the tail of this buffer is tail flit, the mux selects Input_WI. The packet is routed to its destination node via a purely wired link to its destination node.

5.3. Reconfiguration Activation. In this paper, we use the s2s ABP protocol, in which the upstream and downstream routers pass the send and receive signals through signal lines. The upstream router sends a req signal to the downstream router before sending data, and the downstream router transmits data after satisfying the reception conditions, deposits it into the reception buffer, and then feeds back to the upstream router ack signal, thus ensuring data integrity. If the network is congested and the downstream router buffer is full for a long time, the req signal does not get feedback, the packet to be sent will stay in the current router and be re-routed. If a certain number of cycles is reached and the packet still cannot be sent, the packet is discarded. The packet loss rate is one of the critical metrics of network performance. Whenever a packet is lost in a router, the current router sends a packet loss signal to the intelligent node in its subnet, and the reconfiguration counter value in the intelligent node increases by 1. When the reconfiguration counter of the intelligent nodes in all subnets exceeds the threshold, the reconfiguration process starts. The setting of this threshold has a significant impact on the performance of the reconfiguration mechanism. We experimentally determine the impact of the threshold on performance and identify the appropriate threshold under mixed traffic pattern. Frequent reconfiguration can negatively affect network performance since the additional information data packets for the reconfiguration algorithm aggravate network congestion. While insensitive reconfiguration performs inefficiently in relieving network congestion in a timely manner. As shown in Figure 9, choosing 3 dropping packets as the reconfiguration threshold can maximize the improvement on network latency.

6. Routing Algorithm and Deadlock Avoidance

Routing algorithm is applied to regulate data path from the source node to the destination node. When a packet needs to be routed, routing computing module calculates the Manhattan distance from the current node to the

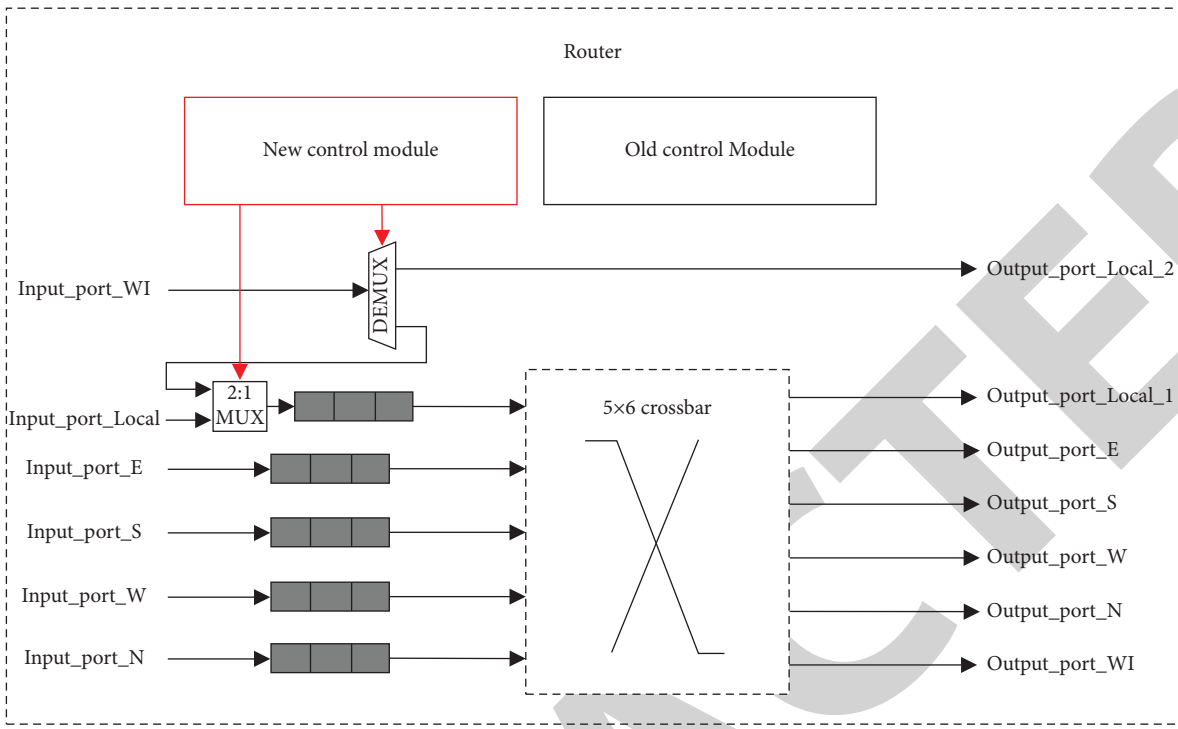

Figure 8: Router architecture modifications.

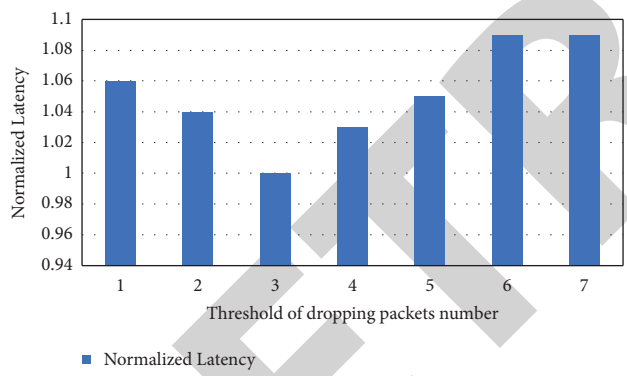

Figure 9: Latency under different threshold of packets dropping number.

destination node. If the distance is greater than the sum of the distance from the current node to the wireless node in the subnet and distance from the wireless node to the wireless node in the destination subnet plus 2, then wireless transmission is used, otherwise direct XY is used. Because the simulation tool Noxim used in this paper has the wireless rate set to 16 Gb/s and the wired link width is 32 bit, one wireless hop is equivalent to two wired hops. Both the wireless and wired transmission XY routes in this routing algorithm are inherently deadlock-free, but when used simultaneously, packet backtracking may occur during wireless transmission across subnets, forming a resource-dependent loop and thus a routing-level deadlock. As shown in Figure 10, the current node is marked as yellow, and packets are transmitted through the wireless hub in subnet3 to subnet0 then from the north port of the router attached to the subnet0 wireless hub to the destination router. The path of the packet is first two hops east and then five hops back south. At this point, if the destination router is also ready to send data to the wireless router (2, 7) in subnet0, and the router (2, 7) is ready to send data to the (0, 7) router in subnet3, a resource dependency loop is formed in the nodes involved in this routing. This is essentially because wired and wireless transmissions are crossed, breaking the unidirectional transmission characteristic of XY routing. In this paper, two virtual channels are designed for each buffer, wireless and wired, in which packets transmitted via wireless are transferred to the wired virtual channel once the wired transmission is completed and forbidden to reenter the wireless virtual channel, strictly dividing the boundary between wired and wireless transmission media, to break the resource dependency loop to achieve a free deadlock.

7. Evaluation

7.1. Simulation Environment. We modified Noxim [19], a systemC-based cycle accurate NoC simulator to implement our design and evaluate it in respect of latency and throughput. The basic parameters of the experiment are set as shown in Table 1. We mixed three different synthesized traffic patterns, transpose1, transpose2, and Ulocal to simulate task immigration. Under the traffic pattern transpose1, data packets generated in node (x, y) are transmitted to node (mesh_dim_x-1-y, mesh_dim_y-1-x). Hence, packets generated near the network center region require fewer hops to the destination node. Under the traffic pattern transpose2, data packets generated in node

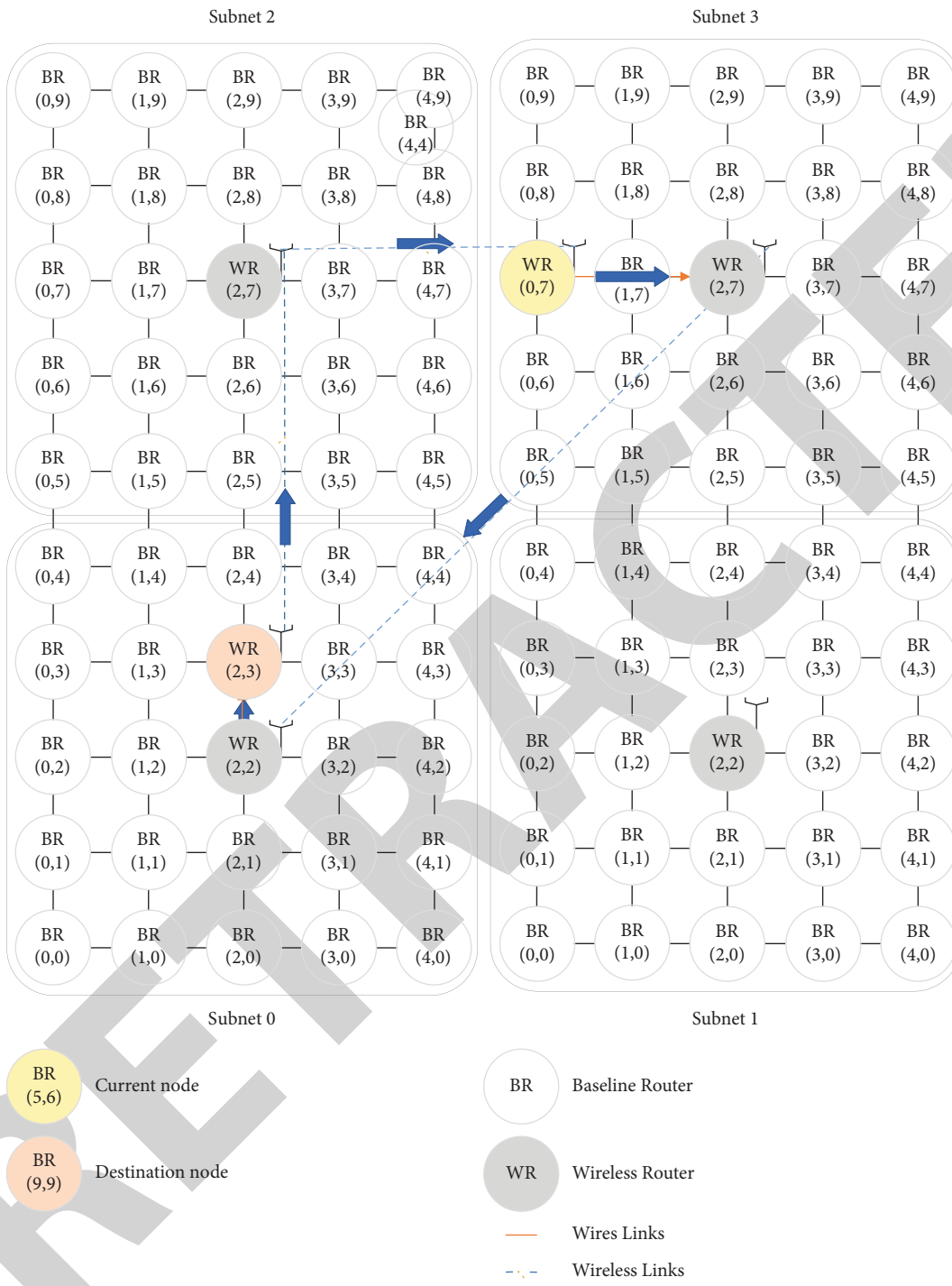

Figure 10: Possible transmission deadlock.

(x, y) are transmitted to node (y, x), leading to more inter-subnet transmission. Under the traffic pattern Ulocal, long-distance multihop packets takes up more proportion. For traffic Mix1, data packets are generated under Transpose1 and Transpose2 alternately for every 4000 cycles. For traffic Mix2, Transpose1 and Ulocal are set accordingly. For traffic Mix3, Transpose1, Transpose2, and Ulocal are set accordingly.

7.2. Comparation Schemes. To evaluate the proposed schemes, we design another reconfigurable wireless interface based on single-dimension historical traffic information. This scheme shares similar hardware architecture with the proposed intelligent interface. A counter is set in each router to record wireless packet transmission. Control modules relink the wireless interface to nodes with higher wireless utilization in last sampling window. We also evaluate the effect of sampling window duration, TS, on network latency. We

Table 1: Experiment settings.

Topology:	Hybrid mesh
Network size:	8×8
Buffer size:	Four flits
VC number:	2
Switch tech:	Wormhole
Traffic pattern:	Mix1, Mix2, Mix3, Random
MAC	Token-hold
Packet size:	Four flits
Simulation time:	21000 cycles

attached four wireless hubs to (1, 1), (2, 1) and (1, 2), (5, 1), (6, 1) and (5, 2), (1, 5), (2, 5) and (1, 6), (5, 5), (6, 5) and (5, 6), respectively, and run the simulation under Mix1. The average latency is reserved as normalized baseline, MAX. Figure 11 shows the normalized average latency increasement with historical traffic-based reconfiguration under several TS compared to MAX. The network latency achieves best performance on 20 cycles sampling window. The sampling window duration influences the convergence time for the reconfiguration mechanism to search appropriate hub-node pairs. Shorter sampling window duration requires smaller wireless counter, and feeds back more sensitively on traffic flow variation. This historical traffic-based scheme is marked ht_rec in the following comparation. The other comparation schemes are randomly fixed hub-nodes connection and wired mesh, marked as fixed_link and mesh, relatively.

7.3. Analysis on Latency Results. As shown in Figure 12, except under random traffic, the proposed scheme overcomes other counterparts under all mixed traffic patterns in respect to latency. Overall, the proposed scheme achieves 6%–10% latency reduction compared to ht_rec scheme, 10%–16% latency reduction compared to fixed_link scheme, and up to 26% reduction compared to mesh scheme when injection rate reaches to more than 0.028 packet/cycle/node.

In Figure 12(a), under MIX1 traffic pattern, ht_rec scheme can locate high-frequency wireless communication pair nodes in short order within the last sampling window. While in next coming traffic, the located pair nodes are highly possible replaced by other pairs. Although the replaced pairs can still be around the former pair, the chance is low. The congestion improvement from ht_rec scheme has tended to come more from the increase of wireless utilization. Hence, the ht_rec scheme shows little advantage on fixed_link scheme. The proposed scheme reconfigures the wireless interface to fit the total traffic pattern by 3-dimension traffic information analysis, instead of a certain simulation period. Due to traffic pattern transition, the fixed_link scheme achieves less wireless utilization than reconfigurable schemes, showing around 20% latency reduction compared to the mesh scheme.

All schemes perform better under MIX2 traffic pattern because of the intrinsic less congestion generation characteristic of Ulocal traffic. While under this traffic pattern, high-frequency wireless communication pair can hardly be formed. Hence, the ht_rec scheme only functions well when traffic pattern changes, leading little difference in performance

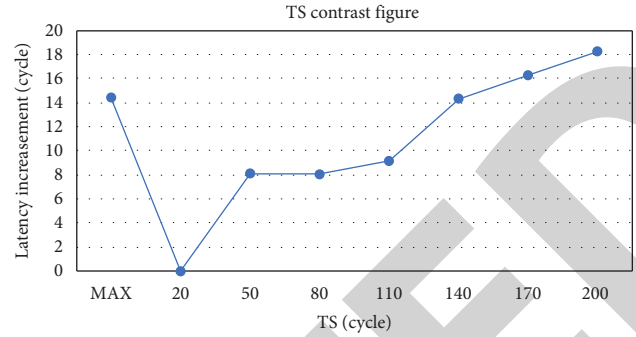

Figure 11: Latency increasement under different sampling window duration.

compared with fixed_link scheme. Also, due to holistic improvement of latency, the chance to break through max latency becomes lower, the reconfiguration process may not be initiated or initiated late in traffic pattern transition. So the effect proposed scheme is not as prominent as in MIX1 traffic.

Under the MIX3 traffic pattern, as shown in Figure 12(c), the transition of three traffic patterns increases the overall average latency, leading the proposed scheme to be launched as expected. The reconfigurable schemes show better performance than in MIX2 traffic. Mesh and fixed_link scheme lack ability to confront with complicated traffic transition, causing more frequent and severe congestion during the simulation period.

Under random traffic pattern, data packets are generated and routed randomly; hence, the simulation results vary a lot. On the average, all schemes perform similarly, and wireless utilization is lower. Due to reconfiguration and data packets carrying traffic information transmission latency, the proposed and ht_rec scheme performs worse than the fixed_link scheme.

Generally, reconfigurable schemes function better in traffic patter with high proportion of long-distance multihop data packets and high wireless utilization rate. The Ht_rec scheme is appropriate for traffic with strong regulatory, especially with regularly generated long-distance communication pairs. The proposed scheme overwhelms ht_rec scheme on the contingent randomness in regulatory since the proposed scheme is designed to fit for the overall generation method of data packets.

7.4. Analysis on Throughput. Across different mixed traffic patterns, the proposed scheme improves throughput between 3% and 9%. Over the ht_rec scheme, 5% to 11% over the fixed_link scheme and 7% to 13% over a wired mesh. The saturated throughput improvement introduced by the proposed scheme is not as prominent as latency improvement. Since saturated throughput is more closely correlated with bandwidth provided by network, the congestion affects throughput by severely blocking packets transmission that the saturated throughput cannot achieve theoretical expectation. The reconfiguration wireless components add more bandwidth to the network, especially to the hotspot: to this respect, the wireless components increase throughput. As shown in Figure 13(a) and Figure 13(c), the proposed

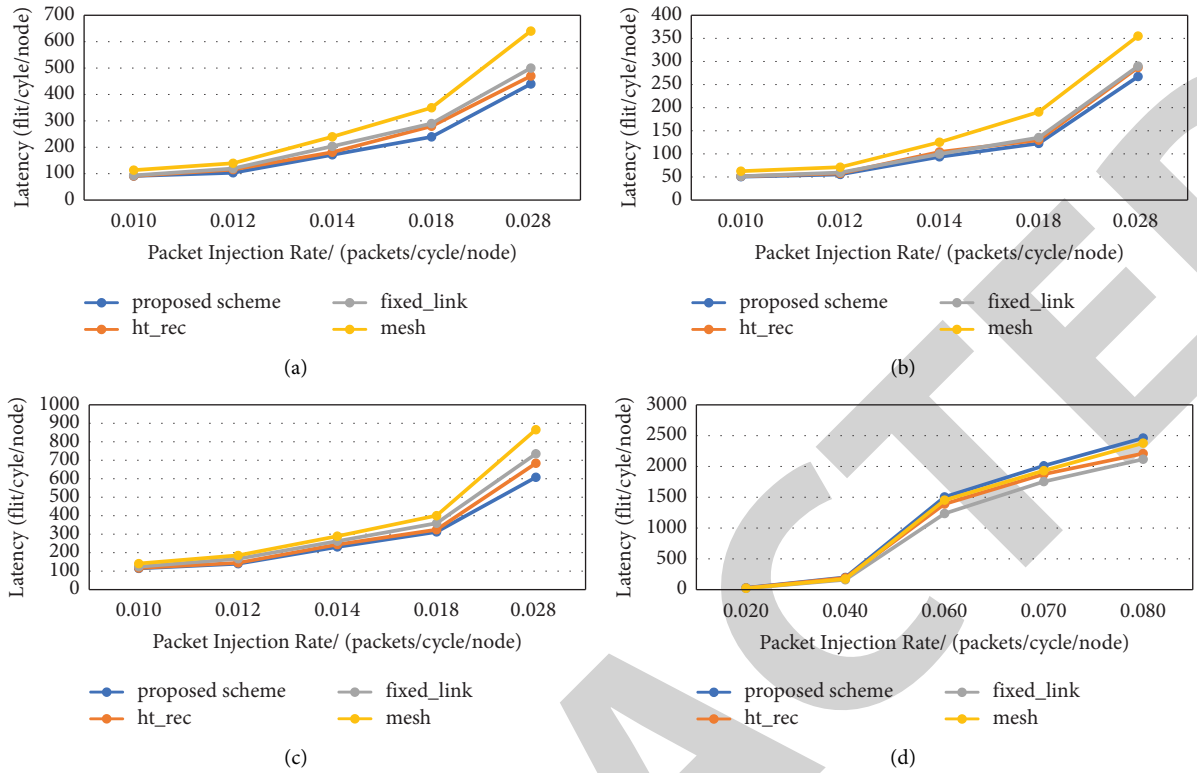

Figure 12: Latency breakdown for different traffic patterns for proposed scheme and other wireless/wired networks. (a) MIX1, (b) MIX2, (c) MIX3 and (d) RANDOM.

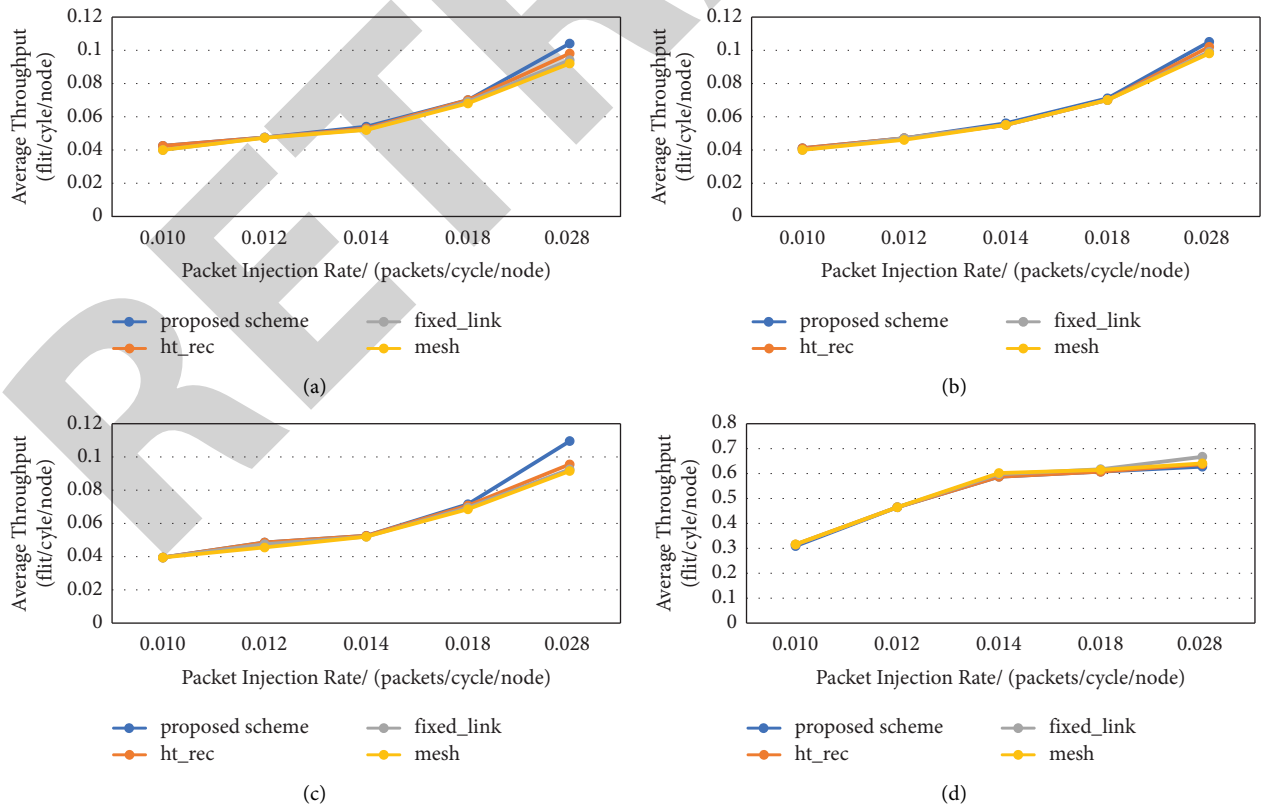

Figure 13: Throughput breakdown for different traffic patterns for proposed scheme and other wireless/wired networks. (a) MIX1, (b) MIX2, (c) MIX3 and (d) RANDOM.

scheme achieves 5% and 9% increasement in throughput over the ht_rec scheme at saturation point. Under a random traffic pattern, the proposed scheme does not have positive effect on latency due to overhead derived from additional packets carrying traffic information transmission and reconfiguration process, the saturation throughput is lower than the mesh scheme.

7.5. Analysis on Area and Power Consumption. We have implemented and synthesized the intelligent modules and additional control modules with Synopsys Design Vision on 65 nm commercial CMOS technology library. To accommodate with frequency setting in Noxim, the targeted frequency is limited to 1 GHz and 1 V supply voltage. The main occupation of additional hardware is the 16 : 1 MUXs and buffers. The area of a typical 5-stage pipeline router with 5 ports and 2-flit buffer is around 1.19 mm^2 and 64 routers are involved in the referred topology. The synthesis results indicate that the additional area overhead is 0.013 mm^2, which takes less than 1.1% compared to all NoC components. Compared with the fixed link method, the area overhead is increased by 0.3%. With Synopsys' PrimePower, we estimate that the configuration modules consume around 1.67 mW during the reconfiguration process. A typical wired router consumes more than 20 mW and hardware support for routing algorithm like FRA [20] consumes around 1.303 mW, so the additional power overhead for reconfiguration is similar to complex routing algorithm, which can be accepted.

8. Conclusion and Future Work

Wireless network-on-chip has become a promising interconnection technology for on-chip multicore systems while the traditional fixed hub-node links can hardly handle the complicated traffic flow on chip. In this paper, we proposed a workload-aware WiNoC design with intelligent reconfigurable wireless interface to improve wireless resources utilization and mitigate congestion. Evaluation shows that the proposed scheme outperforms other counterparts in network latency and throughput improvement. The CNN algorithm adopted in this work belongs to supervised learning. In future work, we hope to be able to use the unsupervised neural network with the goal of reducing latency and increasing throughput, to learn the best way to connect the hubs autonomously. That is, connecting the node and WI randomly at the initial stage, feed the results of the run back to the neural network, and adjust the parameters automatically.

Data Availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The author(s) declare no potential conflicts of interest with respect to the research, author-ship, and/or publication of this article.

Acknowledgments

This research was supported in part by the National Natural Science Foundation of China (NSFC) Research Projects (61874157).

References

[1] L. Benini and G. D. Micheli, *Networks on Chips*, pp. 203–284, Morgan Kaufmann Publishers, Massachusetts MA US, 2006.

[2] S. Deb, K. Chang, X. Yu et al., "Design of an energy-efficient CMOS-compatible NoC architecture with millimeter-wave wireless interconnects," *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2382–2396, 2013.

[3] M. Ramakrishna, V. K. Kodati, P. V. Gratz, and A. Sprintson, "GCA: global congestion awareness for load balance in networks-on-chip," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 7, pp. 2022–2035, 2016.

[4] Q. Wang, Y. Ouyang, Y. Lu, H. Liang, and D. Zhu, "Neural network-based online fault diagnosis in wireless-NoC systems," *Journal of Electronic Testing*, vol. 37, no. 4, pp. 545–559, 2021.

[5] N. Chatterjee, P. Mukherjee, and S. Chattopadhyay, "A strategy for fault tolerant reconfigurable Network-on-Chip design," in *Proceedings of the 2016 20th International Symposium on VLSI Design and Test (VDAT)*, pp. 1-2, Guwahati India, May 2016.

[6] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of the 2010 IEEE International Symposium on Circuits and Systems*, pp. 253–256, Paris France, May 2010.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 60, no. 6, pp. 1097–1105, 2017.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus OH USA, June 2014.

[9] S. Deb, A. Ganguly, P. P. Pande, B. Belzer, and D. Heo, "Wireless noc as interconnection backbone for multicore chips: promises and challenges," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 2, pp. 228–239, 2012.

[10] D. Ditomaso, A. Kodi, D. Matolak, S. Kaya, S. Laha, and W. Rayess, "A WiNoC: adaptive wireless network-on-chip architecture for chip multiprocessors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3289–3302, 2015.

[11] H. Zheng, K. Wang, and A. Louri, "Adapt-NoC: a flexible network-on-chip design for heterogeneous manycore architectures," in *Proceedings of the 2021 IEEE International*

Symposium on High-Performance Computer Architecture (HPCA), pp. 1603–1616, Seoul Korea (South), February 2021.

[12] T. Krishna, C. O. Chen, W. C. Kwon, and L. Peh, "Breaking the on-chip latency barrier using SMART," in *Proceedings of the 2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*, pp. 378–389, Shenzhen China, February 2013.

[13] M. Ebrahimi, J. Wang, L. Huang, M. Daneshtalab, and A. Jantsch, "Rescuing healthy coresagainst disabled routers," in *Proceedings of the 2014 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, pp. 98–103, Amsterdam Netherlands, October 2014.

[14] Y. Kim, G. Kim, I. Hong, D. Kim, and H. J. Yoo, "A 4.9 mW Neural Network Task Scheduler for Congestion-Minimized Network-On-Chip in Multi-Core Systems," in *Proceedings of the 2014 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, pp. 213–216, KaoHsiung Taiwan, November 2014.

[15] J. Park, I. Hong, G. Kim, B. Nam, and H. Yoo, "Intelligent network-on-chip with online reinforcement learning for portable HD object recognition processor," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 2, pp. 476–484, 2014.

[16] S. Lee, O. Jinwook, M. Kim et al., "Intelligent NoC with neuro-fuzzy bandwidth regulation for a 51 IP object recognition processor," in *Proceedins of the IEEE Custom Integrated Circuits Conference*, pp. 1–4, San Jose CA USA, September 2010.

[17] P. Wettin, J. Murray, P. Pande, B. Shirazi, and A. Ganguly, "Energy-efficient multicore chip design through cross-layer approach," in *Proceedings of the 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 725–730, Grenoble France, March 2013.

[18] R. Kim, G. Liu, P. Wettin, R. Marculescu, D. Marculescu, and P. P. Pande, "Energy-efficient VFI-partitioned multicore design using wireless NoC architectures," in *Proceedings of the 2014 International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES)*, pp. 1–9, Uttar Pradesh India, October 2014.

[19] V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti, "Noxim: an open, extensible and cycle-accurate network on chip simulator," in *Proceedings of the 26th IEEE International Conference on Application-specific Systems Architectures and Processors*, pp. 162-163, Toronto ON Canada, July 2015.

[20] M. Ebrahimi, H. Tenhunen, and M. Dehyadegari, "Fuzzy-based adaptive routing algorithm for networks-on-chip," *Journal of Systems Architecture*, vol. 59, no. 7, pp. 516–527, 2013.