

## Research Article

# A Hybrid Selection Strategy Based on Traffic Analysis for Improving Performance in Networks on Chip

Mohammad Trik <sup>1</sup>, Ali Mohammad Norouzzadeh Gil Molk <sup>2</sup>, Fatemeh Ghasemi <sup>3</sup>,  
and Pourya Pouryeganeh <sup>4</sup>

<sup>1</sup>Department of Computer Engineering, Boukan Branch, Islamic Azad University, Boukan, Iran

<sup>2</sup>Department of Computer Engineering, University of Guilan, Rasht, Iran

<sup>3</sup>The Institute of Higher Education, Ministry of Science, Research and Technology, Computer Engineering Department Qazvin, Iran

<sup>4</sup>Electrical & Computer Faculty, Isfahan University of Technology, Iran

Correspondence should be addressed to Mohammad Trik; trik.mohammad@gmail.com

Received 16 November 2021; Revised 21 March 2022; Accepted 31 March 2022; Published 25 April 2022

Academic Editor: Gengxin Sun

Copyright © 2022 Mohammad Trik et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Networks on chip (NoCs) are an idea for implementing multiprocessor systems that have been able to handle the communication between processing cores, inspired by computer networks. Efficient nonstop routing is one of the most significant applications of NOC. In fact, there are different routes to reach from one node to another node in these networks; therefore, there should be a function that can help to build the best route to reach the destination. In the current study, a new hybrid algorithm scored regional congestion-aware and neighbors-on-path (ScRN) is introduced to choose better output channel and thus improve NOC performance. Having utilized the ScRN algorithm, first an analyzer is used to inspect the traffic packets, and then the NoC traffic locality or nonlocality is determined based on the number of the hops. Finally, if the traffic is local, a scoring technique will choose better output channel; however, if the traffic is nonlocal, the best output channel will be chosen based on a particular parameter introduced here as well as the system status using NoP or RCA selection functions. In the end, via Nirgam simulation, the proposed approach was assessed in traffic scenarios through various selection functions. The simulation results showed that the solution was more successful in terms of delay time, throughput, and energy consumption in comparison to other solutions. It showed a reduction of 38% in packet latency, and the throughput increased by 20%. By considering these two parameters, energy consumption decreased by 10% on average.

## 1. Introduction

The growing need for more effective chips has currently led to an increase in complexity of designing integrated circuits (IC) [1]. Some issues have been resolved with the use of smaller transistor manufacturing technology; however, smaller manufacturing technology has led to the problem of imbalance between the connection wire delays and the gate delays [2]. Besides, as the frequency of chips' performance increased, the power consumption rises as well. To deal with these challenges, IC designers focused on increasing efficiency rather than speed, and this change of attitude resulted in placing multiple individual processors in one chip and establishing communication between them

through a single bus. The result was so satisfactory that after a short time, the systems consisted of several sections that ran on a board, relocated in a single chip. This architecture of processor construction became popular as system on chip or SoC [3–5]. However, SoC had issues emerging over time. As the number of separated sections, known as Intellectual Property (IP), increased, the SoC was not responsive [6]. The issues such as unscalability and massive power consumption in the bus encouraged new efforts among IC designers. The solution to these challenges led to a novel architecture named NoC. NoC is a connectional subsystem inside an IC (normally called “chip”), which typically provides the connection between IP cores of the system in a chip [7]. NoC technology uses network theory and in-chip

connection approaches and provides significant progress compared with bus and crossbar-based connections. NoC improves the scalability of SoCs and optimizes the energy usage in complicated SoCs in comparison to other models [8]. The factors that affect NoC design are energy consumption limit, delay, and throughput [9]. In fact, in NoC applications, due to present limits, the proposed algorithms must be designed in a way that reduces the overall energy consumption of the network and packet latency, causes an increase in performance and throughput of the network, and has a sufficient overhead-implementation. One of the important factors which affect NoC performance is the process of selecting the best output channel [10]. By designing and applying efficiently, the selection function can reduce packet latency and, due to more uniform traffic distribution on the network, increase the network throughput, and as a result, decrease the energy consumption [11]. Another challenge of NoC is the discussion of routing in these types of networks. The problem may occur in routing algorithms for instance deadlock, livelock, and starvation. Our proposed method in this study covers all cases so that the deadlock does not get excited and prevents packages from livelock. Also, the ScRN selection strategy makes that there is never any starvation. The performance requirements of today's NoC are also felt to severely affect the performance of these networks, which can be summarized as such: latency, throughput, power consumption, and fault and distraction tolerance. The key contributions of this paper are as follows: (i) Introduce a new hybrid selection function, which is able to use appropriate strategies for each mode depending on the local or nonlocal status of the packets. (ii) Introduce a new density awareness method called ScRN to select the best output channel for packet distribution. (iii) Improve the use of local and nonlocal congestion information: The output selection strategy uses a traffic analyzer to examine packets and then determine whether the packet is local or nonlocal based on the number of hops, and this can improve the network. The major goal of this paper is to develop a hybrid selection strategy with the aim of allocating the best channel that will allow packets to be routed to their destination along a path that is as free of congested nodes as possible. Networks on chip can use dedicated control lines to transport data between routers, unlike traditional computer networks, which can only communicate internode information through packets. This allows useful information about congestion-related aspects like the buffer status of individual nodes to be exchanged without adding additional traffic overhead.

*1.1. Motivation.* The importance of this research is in applying a hybrid solution in order to select the best output channel in routing networks on the chip. For this purpose, first, a traffic analyzer is used, and according to the number of hops of a package, it is determined whether it is local or nonlocal; then through it, a decision is made about the type of selection strategy. Accordingly, if the package is local, the optimized strategy is used for local packages, and in the nonlocal case, special strategies are used for nonlocal packages. Using this technique, packets can be routed through the best output channel, and as a result, network-level bal-

ance can be established. This can prevent hotspots, increased energy consumption, and long delays. The function of our solution is to use the information of the neighbors close to the node to which the packet has reached, to dynamically check the local and global network traffic and route the route in such a way that traffic and congestion are minimized. As a result, by creating a kind of load balance through the distribution of traffic in different routes, heat is generated, and thus energy consumption is increased. This solution is independent of the type of topology and can be used in network on chip based on neuromorphic and even wireless networks.

*1.2. Paper Organization.* Our paper is organized as follows. In the next section, a list of related works is stated in two groups: the previously used algorithms in NoC along with selection functions and performance techniques. In Section 3, we propose a definition of the system model in a descriptive way and network architecture. In Section 4, the proposed hybrid method is stated to propose a hybrid selection function (ScRN). In Section 5, the results of analyzing the proposed model in different scenarios are shown, and finally, we explain these scenarios in Section 6.

## 2. Related Works

The content of related works is divided into two parts. The first part belongs to the examination of previously utilized algorithms in NoC along with selection functions, which in the end, we explain them briefly in the form of a table. The second part evaluates some performance techniques, including energy consumption, throughput, and delay. Also, the comparison between these techniques in various previous studies is summarized in a table.

*2.1. Previous Designs Related to Routing Algorithms and Selection Functions.* Over recent years, numerous researchers have studied different utilized algorithms along with selection functions for different fields in NoCs, and we examine some of the performed works in these subjects in the following sections. A selection strategy named EnPSR is introduced in [12] for better performance of the network. This approach has the ability to reduce the hardware overhead through access to the data aware of the output channels. The evaluation results showed that compared to other methods, this method is significantly improved in terms of packet latency, throughput, area, and the energy consumption. A congestion-aware routing algorithm called DBAR is proposed in [13]. This approach overcomes local and global adaptive routing problems and provides an entirely adaptive, efficient routing to avoid congestion. In another study, researchers proposed an adaptive nonminimum routing algorithm called LEAR, which avoids congested routes from source to destination [14]. In reference [15], an MILP approach is proposed for unicast and multicast traffic distribution in networks on 3D mesh-based chip. This method was based on the Hamiltonian path and proposed to avoid congestion. In order to increase fault tolerance for NoCs in [16], the EDAR algorithm was introduced. This approach is based on the weighted path selection strategy, which

TABLE 1: Summary of utilized algorithms in NoC along with selection functions.

Work	Outlines	Features	Selection strategy name
[12]	Locally congestion-aware	Hardware overhead optimization	EnPSR
[18]	Input selection strategy	Throughput and latency improvement	DCA
[16]	Fault-tolerant improve	Higher throughput, low overhead	Weighted path
[15]	Distribute the unicast and multicast traffic	Performance improvement	—
[17]	Congestion detection	Fast evaluation	—
[13]	Congestion avoidance locally	Better performance	NoP
[14]	Congestion avoidance locally	Reducing energy consumption	—
[20]	New selection strategy	Performance improvement	NoP-OE
[21]	Congestion controlled	Traffic estimation	Fuzzy controller
[22]	Congestion avoidance locally	Load balancing	CADA

provides NoC true traffic conditions through monitoring modules. In the proposed EDAR, real-time input weights are calculated according to the channel states like idle/busy/congested/false, and least-weighted input is ranked as the near-optimal path toward the sets. In [17], the author proposed a congestion detection algorithm called CACBR that selects the best route using two methods of candidate paths and cluster's congestion information and also uses virtual channels to ensure avoidance of deadlock. In another study, the researchers attempted to decrease packet latency and increase network throughput using an output selection method named DCA. One of the advantages of this method is the capability of utilizing it on any kind of topology and network of different dimensions [18]. The researchers in [19] proposed the adaptive routing method called PT-BAR which uses temperature conditions for packet routing. In this algorithm, the high and low priority packets are routed from high- and low-heat regions, respectively. In [20], a selection function named OE-NoP is proposed which has adaptability with any routing. The purpose of introducing this function is packet routing during traffic creation toward the destination. In order to establish traffic control and balance in [21], a selection function based on the fuzzy controller is introduced. Traffic estimation for free packet routing is one of the properties of this method. Congestion control in wireless sensor networks, especially wireless network on chip, is one of the main challenges for effective performance in these networks. In [22], researchers have proposed a resource control mechanism using the Q learning method with an alternative path approach to reduce congestion. This congestion-aware data acquisition (CADA) mechanism initially identifies the congestion node (CN) where the nodes' buffer occupancy ratio is higher. Devanathan et al. [23] provides a solution for WiNoC communications that minimizes congestion by using effective wireless communication between output channels and routers. In [24], a wireless network architecture is presented on the chip to prevent congestion and load balancing. To do this, they have adopted a virtual output queue scheme to handle HOL blocking, which has significantly improved the network throughput. The list, properties, and type of selection strategy from utilized algorithms for NoC are given in Table 1.

As indicated in Table 1, some of the algorithms use selection function, while presence of a selection strategy can have a significant effect on the performance of routing algorithms and as a result the performance of the entire network.

*2.2. Previous Designs Related to Performance Techniques.* As was mentioned in the first section, NoCs are primary adaptive connection infrastructures for system on chips (SoCs). One of the important issues in NOC is system performance, such as delay, throughput, and energy consumption of the system, which, along with scalability in these networks, have special importance [25]. In [26], the microkernel idea was introduced to reduce energy consumption in multicore-based operating systems (OS). The proposed method is in such a way that OS is divided into microkernel and other system modules and distributes in the network to provide service for user applications. In [27], a self-adaptive mapping named SCSSO is introduced based on the mapping method. The proposed method uses the k-NN method to significantly improve system performance in terms of energy consumption level, delay, and throughput. In [28], the ALO routing method is proposed to deal with energy loss routers in which routing evaluation is ran using spin, octagon, and cliché topologies. In [29], an intelligent task mapping algorithm on protocol-level is introduced to optimize energy consumption. This method evaluates the energy modeling in the protocol level so that the energy consumption level minimizes based on the protocol activity. Since links of the on-chip networks consume about 50% energy, and this issue has great importance in NoC, in [30], an energy consumption estimation method for links using virtual channels is proposed for precise estimation of energy consumption from data-dependent links. In [31], two NoC architectures are proposed, which are based on the circuit and packet switching. For both architectures, energy consumption models are proposed in which the energy consumption levels of them are estimated based on the prediction in each transferred bit. Another method for decreasing energy loss is proposed in [32]. In this research, the dissipated energy in links (links lose a large portion of energy in on-chip network although this energy loss can increase in future technologies) is reduced using some set of encrypted programs. Researchers in [33] proposed a method for reducing the

TABLE 2: A summary of used techniques for optimizing the system performance for NoC.

Technique(s)	Reduced latency	Reduced area	Energy efficient	Ref.
Microkernel	√	—	√	25
SCSO	√	—	√	26
ALO-bufferless	—	√	√	27
Smart protocol-level task mapping	√	—	√	28
Estim-NoC	√	—	√	29
EA-NoC	—	—	√	32
HS	—	√	√	33

energy consumption named EA-NoC that avoids unnecessary energy consumption using the most optimized path between source and destination and also optimizes the dynamic energy. Moreover, the proposed method can be efficient for parameters such as delay and throughput. In [34], the author examined the energy consumption in asynchronous NoCs. In that research, five optimizing approaches are analyzed for reducing energy consumption. Among these methods, the HS algorithm is the most efficient method which consumes the least energy by recognizing the shortest path. This research offers a multihop routing algorithm based on path tree (MHRA-PT) to minimize network energy consumption by addressing difficulties such as random cluster head selection, redundancy of working nodes, and building of cluster head transmission path. The suggested algorithm may successfully minimize network energy consumption, balance network resources, and extend network life cycle, according to simulation results [35]. In [36], on the assumption that the number of available channels is infinite, this study offers a one-shot time division multiple access (TMDA) scheduling with unlimited channels. To resolve slot conflict, the study presents scheduling with limited channels (SLC) and employs a lookahead search technique. A distributed implementation based on token change is offered for the algorithm's scalability. In Table 2, a summary of this section is presented based on different parameters for optimizing the system performance in NoC.

### 3. System Model

*3.1. Network on Chip Architecture.* NoC is a standard approach for multicore applications which consists of four main sections of routers, routing algorithms, IP cores, and network adaptor. These four sections are the major backbone of this type of network which exist in a node and are connected by wires. IP cores are processing units of the network. The network adaptor is used for connection of one core with other cores, and routers are responsible for network routing [1]. The task of routers is to navigate and transit the packets using routing algorithms in the network; more details of which are presented in the next section. NoC architecture is designed based on virtual channels and wormhole-based switching. Figure 1 presents the standard  $4 \times 4$  mesh network along with details of a router [4, 37–39].

*3.2. Switch and Router Structure in NoC.* Routers play an essential role in the performance and efficiency of network

on-chips. For instance, the design accuracy and use of routers can reduce the consumption power and delay and increase the NoC performance [3]. As can be seen in Figure 2, a router consists of different sections, including a switch, input and output buffers, routing and judgment unit, link controller, and injection and output channels. Buffers must be able to save data temporarily to prevent congestion for input and output routers during the network chaos. The switch establishes the connection between input and output buffers [18, 25, 40]. The routing unit is responsible for running the routing algorithms. The link controllers coordinate the packets flux on the channels, and output and input channels establish the connection of one processor with adjacent routers.

*3.3. Selection Function.* When the routing algorithm returns more than one output channel, the selection function is used to choose the output channel to which the packet is sent because the adequate selective pattern has a significant impact on the overall performance of selection routing. Namely, the adaptive routing algorithm measures a set of acceptable output channels regarding the paths that the packet can pass through to reach the destination. Afterwards, according to the network characteristics, including the congestion rate or the length of one of the routes of the output channel, the selection function will be utilized to choose the output channel from a set of permitted output channels. The overall schematic of using the routing algorithm and selection function in ScRN is presented in Figure 3 [4, 20, 25].

### 4. The ScRN Algorithm

In this section, an efficient selecting approach is proposed for choosing adaptive routing algorithms. In this approach, the local and global traffic condition of the network is dynamically examined by using the information obtained from neighbors near the node which the flit reached. This method routes the packet in a way that the traffic and congestion minimize, which consequently prevents heat generation in one section and unnecessary energy consumption by establishing a load balance through traffic distribution in different routes. First, Figure 4 presents the overall architecture of the approach where the routing algorithm finds output paths and the selection function with defined strategy selects the best output channel. This architecture comprised of input/output ports, input buffers, units for Traffic Analyzer

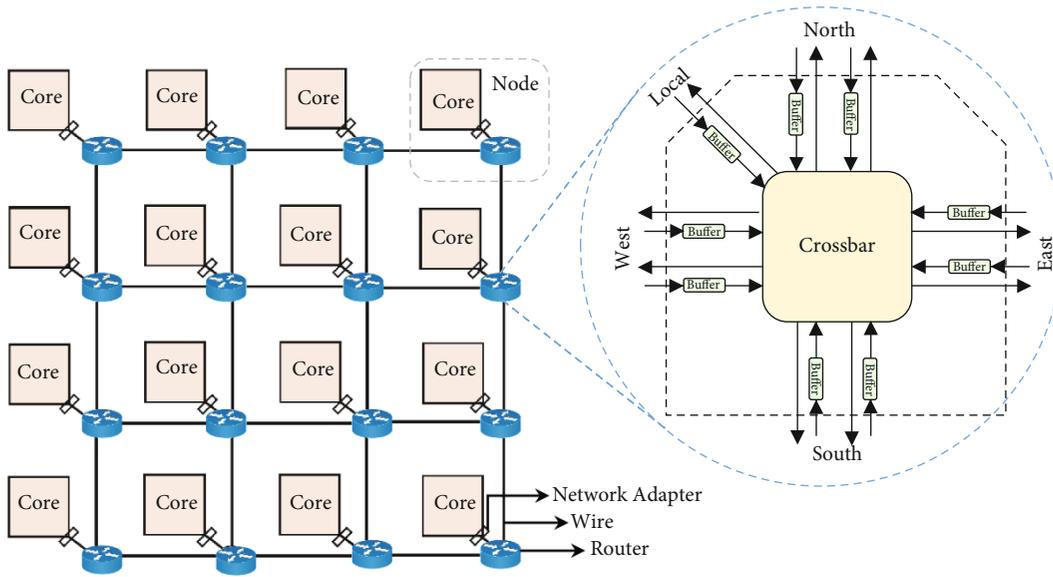


FIGURE 1: The standard 4 × 4 mesh network.

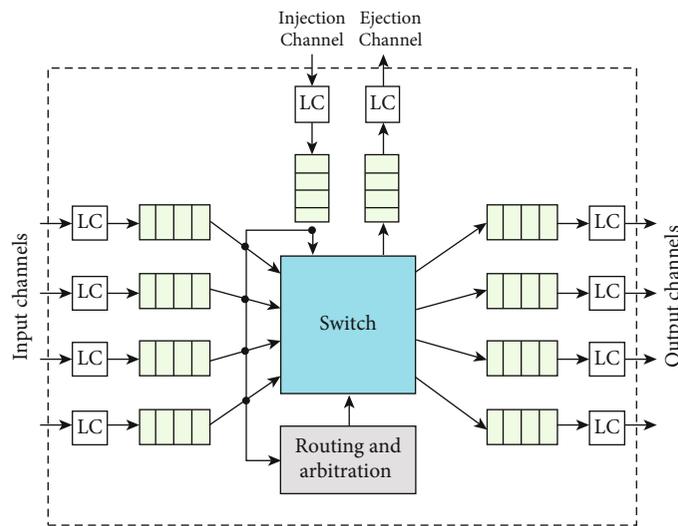


FIGURE 2: The internal structure of a router [8].

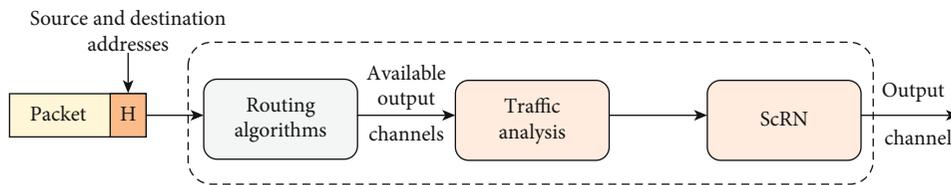


FIGURE 3: Routing structure and selection route blocks.

(TA), NoP,RCA and Scored selection strategy, and a crossbar switch. Then, Figure 5 shows the flow chart of the proposed approach. In the first hop, a traffic analyzer is utilized, and with the help of this analyzer, first, the traffic type is defined, and then, the best selection function corresponding to traffic type is used. In this case, the locality condition of the traffic is examined, and if the hop length corresponding to the packet is less than 2, the traffic is local, and otherwise, it is nonlocal. Also, in the nonlocality case,

the hop number is examined, and if it is equal to 2, the NoP function is used, and if it is more than 2, the RCA selection function is employed to determine the best route. If the traffic is local, a scoring-based strategy (scored strategy) is applied to determine the output.

*4.1. Formulation of Energy Consumption.* In this study, first, the mean energy consumption of sending a data flit between two neighbor tiles, including energy consumption in both

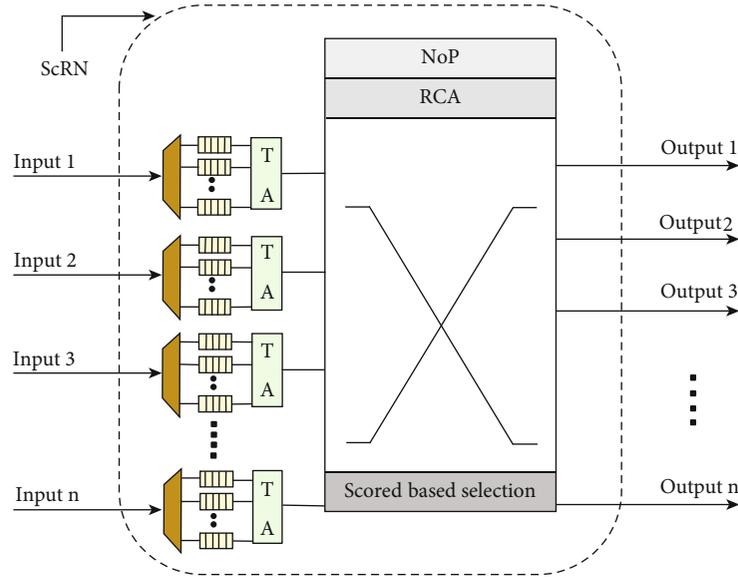


FIGURE 4: ScRN router and selection architecture.

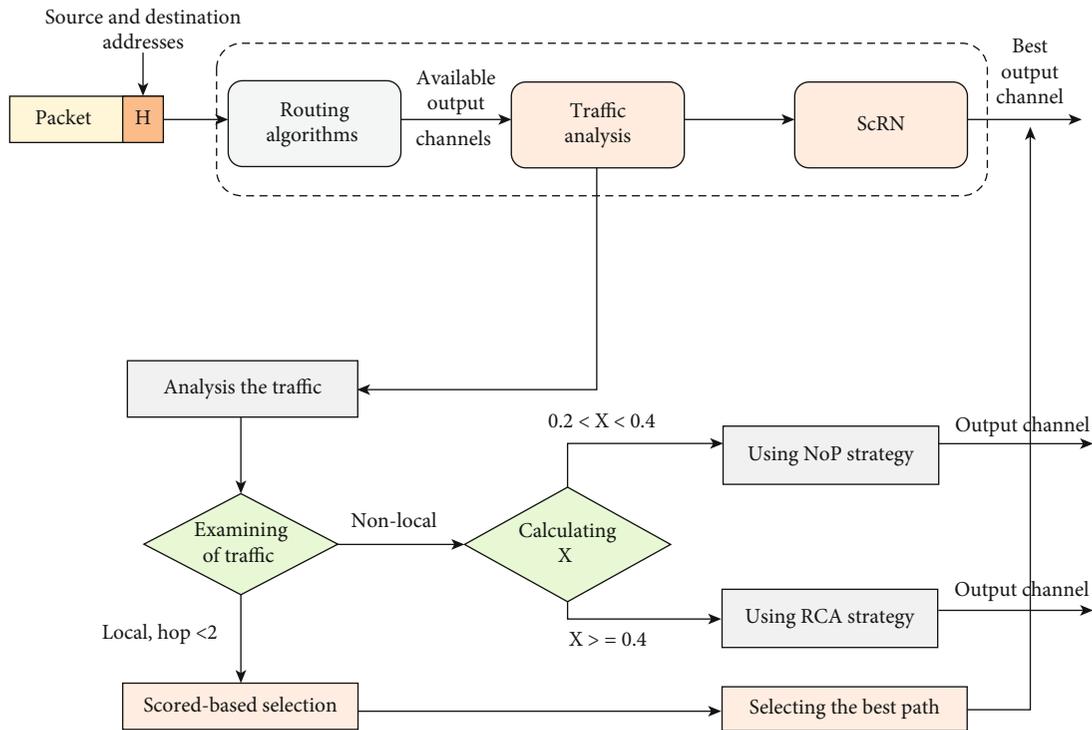


FIGURE 5: Flow chart of the proposed approach for ScRN algorithm.

router and connection links of them, is presented in Figure 6 to model energy consumption for each flit which is the smallest physical unit of data exchange in NoC. We have used Ref. [14] to calculate energy consumption.

$$E_{\text{Neighboring-routers}} = E_{1\text{-hop}} = E_{\text{Intra-router}} + E_{\text{Inter-router}} \quad (1)$$

In Equation (2), exchange energy between two neighboring routers is divided into two parts of the inside of the

router and between routers. The inside-router energy consists of three sections of intersection switch, the buffer related to virtual channels, and wirings inside the router, according to Figure 1. Hence,

$$E_{\text{Intra-router}} = E_{\text{Crossbar}} + E_{\text{buffer}} + E_{\text{wire}}. \quad (2)$$

On the other hand, the connection between two routers depends on the defined number of bits for flits on NoCs,

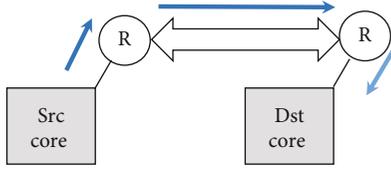


FIGURE 6: Route with a hop.

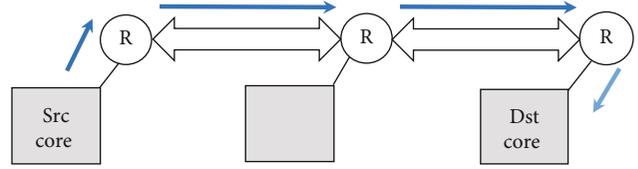


FIGURE 7: Route with two hops.

and showing the energy consumption in each of these wires with  $E_{\text{Inter-tile-Link}}$ , we have

$$E_{\text{Inter-router}} = \text{phit\_size} \times E_{\text{Inter-tile-Link}}. \quad (3)$$

Therefore, NoC energy consumption in the simple case of connection of two neighboring routers can be calculated as

$$E_{\text{NoC-1hop}} = E_{\text{Crossbar}} + E_{\text{buffer}} + E_{\text{wire}} + E_{\text{Inter-router}}. \quad (4)$$

The length of the connection wires of each pair of tiles in NoC is usually in millimeters (mm), while the length of the router wires is usually in micrometers ( $\mu\text{m}$ ). Therefore, energy consumption in the internal buffers of the router ( $E_{\text{buffer}}$ ) and internal wires of the router ( $E_{\text{wire}}$ ) is insignificant compared with energy consumption between routers ( $E_{\text{Inter-tile-Link}}$ ):

$$E_{\text{buffer}} + E_{\text{wire}} \ll E_{\text{Inter-tile-Link}}. \quad (5)$$

Thus, Equation (5) is simplified as

$$E_{\text{NoC-1hop}} = E_{\text{Crossbar-1hop}} + E_{\text{Inter-router}}. \quad (6)$$

The energy consumption  $E_{\text{crossbar-1hop}}$  includes energy consumption of two intersecting sections of source and destination routers, in other words:

$$E_{\text{Crossbar-1hop}} = E_{\text{Crossbar-Src}} + E_{\text{Crossbar-Dst}}. \quad (7)$$

Thus, in homogenous architecture, the routers' structure is similar. Equation (8) is simplified as

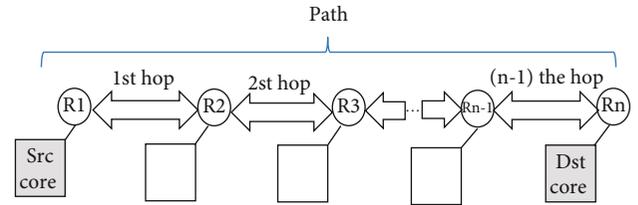
$$E_{\text{Crossbar}} = 2 \times E_{\text{Cross-router}}. \quad (8)$$

As a result, the exchange energy consumption between two neighboring routers will be

$$E_{\text{NoC-1hop}} = 2 \times E_{\text{Cross-router}} + E_{\text{Inter-router}}. \quad (9)$$

According to Figure 7, Equation (9) for a route with a length of 3 changes as

$$E_{2\text{-hop}} = 3 \times E_{\text{Cross-router}} + 2 \times E_{\text{Inter-router}}. \quad (10)$$

FIGURE 8: Route with  $n$  hops.

Equations (9) and (10) can be generalized, and considering Figure 8, the average energy consumption of sending data flit from  $\text{tile}_{\text{Src}}$  to  $\text{tile}_{\text{Dst}}$  in general can be calculated as

$$E_{\text{Path}} = E_{n\text{-hop}} = n \times E_{\text{Cross-router}} + (n - 1) \times E_{\text{Inter-router}}. \quad (11)$$

In Equation (11), variable  $n$  indicates the number of routers existing in route. Hence, this equation shows that the mean energy consumption of sending data from the source core to the destination core depends on the number of hops of the route. Hop number in the mesh between source and destination is determined by Manhattan distance between two cores. Manhattan distance is an indicator of distance between two points and is equal to the sum of the absolute values of the difference between width and length of those two points.

If a vector of length  $n$  is to be used for addressing the routers in the  $n$ -dimensional case,

$$\begin{aligned} V &= (v_1, v_2, \dots, v_n), \\ W &= (w_1, w_2, \dots, w_n). \end{aligned} \quad (12)$$

The Manhattan distance of the two vectors is

$$|V - w| = \sum |v_i - w_i|. \quad (13)$$

**4.2. Identification of Traffic Type.** A traffic analyzer is used to identify the traffic type. This analyzer obtains the destination address of each package, which is directed through the router and examines its data in each T hour cycle. Based on this, a two 5-bit counter is used for the determination of locality or nonlocality of requests in the router. If the desired destination packet is two hops or farther from the current router, it is regarded nonlocal; otherwise, it is local. In fact, the analyzer calculates the hops associated with packet periodically and, based on that, updates the local (L) and nonlocal (N) counters. This data is sent to switch for decision-making regarding the selection strategy. The

---

```

1: Input: Packet hops (pkt_dst_hops),
2: Out put: Local and Non-Local value (L = Local value, N = non_Local value)
3 Initializing: L = 0, N = 0
4:     if pkt_dst_hops ≥ 2 then
5:         N++;
6:     else
7:         L++;
8:     end

```

---

FIGURE 9: Pseudocode corresponding to traffic type recognition.

counter is erased after each T hour cycle. Figure 9 shows the pseudocode corresponding to the traffic analyzer.

In fact, using the analyzer, proper information about traffic rate and their convergence toward local or nonlocal traffics can be obtained, and then based on that, routing can be performed in the next hop.

**4.3. Selecting the Best Output Channel.** Using this approach, it will be possible to use the best selection strategy based on the traffic type. The considered condition is examined in each T hour cycle. It should be noted that if the T value is assumed large, network response to changes in traffic patterns reduces, and if this value is considered low, high switches cause a reduction in the efficiency. Overall, studies have shown that if the value of T is assumed a 32 cycle hour, the maximum efficiency is achieved. Based on this, at the end of each 32 cycle hour, the traffic pattern is determined using the output of the analyzer. According to the evaluations, this has been achieved that if traffic pattern tends toward local traffic destinations, scoring-based selection strategy is activated. Consequently, the score of one output channel is calculated through

$$Score[d] = \alpha \times Const[d] + \beta \times (B[d]/max\ buffer + \gamma) \times (\Delta P/max\ power), \quad (14)$$

where  $\alpha$ ,  $\beta$ , and  $\gamma$  are the weight factors for the probability of link selection, free buffer rows, and instantaneous power consumption, respectively. Since free buffer rows ( $B$ ) and instantaneous power consumption ( $\Delta p$ ) have different units, they are normalized using max-buffer and max-power factors. Also, considering that  $Const$  is in the range of (0,1), no normalization is needed. Afterward, the score of adaptive routing functions and all possible values of  $\alpha$ ,  $\beta$ , and  $\gamma$  are evaluated, and the adequate coefficients for each of the routers are obtained through

$$\alpha + \beta + \gamma = 1, \quad (15)$$

$$\gamma = 1 - (\alpha + \beta), \quad (16)$$

$$\beta = 0, 0.1, \dots, (1 - \alpha), \quad (17)$$

$$\alpha = 0, 0.1, \dots, 1. \quad (18)$$

For instance, the best values for  $\alpha$ ,  $\beta$ , and  $\gamma$  in even/odd routing under the MMS traffic scenario are 0.3, 0.4, and 0.3, respectively. Another important property of ScRN is its adaptability to any network topology and adaptive routing function. However, if traffic is nonlocal, a strategy based on RCA and NoP is activated as a proposed strategy for nonlocal traffics. In other words, under these conditions, the locality and nonlocality of the router will be determined based on the traffic pattern rate based on the traffic pattern rate. For non-edge routers in an  $8 \times 8$  mesh, the local traffic penetration coefficient to nonlocal traffic is considered higher because it affects the overall network performance. Based on this, provided that a minimum rate of nonlocal traffic exists in routers, the ScRN method is activated. As a result, the local to nonlocal traffic rate ( $X$ ) should be considered an effective parameter. Based on the evaluations, this parameter is considered a constant equal to  $X = 0.4$ , and it has been argued that it can induce maximum performance in the network. In other words, if at least 40% of the router's traffic is routed toward nonlocal destinations, the intended strategy needs to be activated. Therefore, following these principles in this study, the operations regarding the switching are expanded as

$$\begin{cases} 0.2 < X < 0.4 & \text{NoP Switching} \\ X \geq 0.4 & \text{RCA Switching} \end{cases} \quad (19)$$

The ScRN algorithm associated with switching operation based on the traffic analyzer is shown in Figure 10. The input data to this algorithm are the type of local or nonlocal data, and the output is the best selection strategy. Also, it should be noted that since analyzer and switch receive only router data at once, consequently, no overhead in network communications is produced. In this Figure N and L represent Local and Non-Local packets and T represent clock cycles.

## 5. Evaluation and Simulation Environment

A Nirgam simulator is used to evaluate the proposed algorithm whose capabilities are listed in Table 3 [5]. Nirgam is a scalable, modular simulator based on the system C hardware description language, enabling various options at every stage of NoC design, including topology, switching methods, virtual channels, buffer parameters, and tested routing mechanisms. Moreover, the configuration parameters for

```

1: forevery  $T$  clock cycles do
2:   Catch  $L$  and  $N$  value from analyzer;
3:   Compute  $x = N/(L+N)$ ;
4:   If  $x = 0.2 < x < 0.4$  then
5:     Switch to NoP;
6:   elseif  $x \geq 0.4$  then
7:     Switch to RCA;
8:   else
9:     Using score strategy;
10:  end
11: end

```

FIGURE 10: Pseudocode of switching procedure and strategy type selection in ScRN.

TABLE 3: Main capabilities of Nirgam simulator.

Types of production traffic	Routing algorithms type	Switching mechanism	Topology type
Constant bit rate trace and Bursty based	Odd-even, XY	Wormhole	Torus, mesh

TABLE 4: Simulation parameters.

Parameter	Configuration
Network size	8*8 mesh
Schemes	DICA [18], RCA [21], NoP [10], ScRN
Packet size	8 flits
Buffer size	4 flits
Traffic profile	Random, transpose, hotspot-center, hotspot-row
Simulation time	10

the analysis and simulation of the proposed method are given in Table 4. For the type and size of the utilized network in simulation, an  $8 \times 8$  mesh network with a wormhole switching mechanism is considered [20, 41]. The routing function used in evaluations is odd-even algorithm, and the capacity of input buffers was 4 flits; the queue type was FIFO, and the size of each packet was defined as 8 flits. Simulation was performed for 200,000 cycles, and the first 20,000 cycles were determined as the warm-up time for stabilization of results. The entire simulation scenario was repeated ten times for more accuracy of the results whose average was calculated [18].

**5.1. Traffic Scenarios Used in Algorithm Evaluation.** In simulations performed to evaluate the selection functions, traffic scenarios are utilized. In a random traffic pattern, a node sends the packets with the same probability to other nodes. In transpose traffic pattern, a node in position  $(x, y)$  only sends the packets to coordinates  $(n-1-y, n-1-x)$ . In this traffic pattern,  $n$  is the mesh network size (number of columns or rows). The performance of the proposed algorithm is studied for hotspot traffic as well. The hotspot traffic is like random traffic, which receives more traffic percentage than other nodes, in addition to nodes from the network. As shown in Figure 11, two types of hotspot traffic patterns are utilized in evaluations. One is

named the hotspot-center traffic, and the nodes which are located in position (4,4) and (5,5) receive 10 percent more traffic than other nodes. The other hotspot traffic pattern is hotspot-row in which the points located in one row with coordinates (2,2), (3,2), (4,2), (5,2), (6,2), and (7,2) receive two percent more traffic than other nodes of the network.

**5.2. Evaluated Parameters.** The average packet latency, network throughput, and energy consumption in various injection rates and under different traffic patterns are evaluated to show the performance of the ScRN algorithm. The average packet latency is equal to the average delay of all packets received at the destination. In other words, the interval between injection of header flit in the network of the source node and receiving one flit sequence in the destination node is packet latency. In Equation (20),  $K$  is the total number of delivered messages in the destination node, and  $L_i$  is the delay of the  $i^{\text{th}}$  message [5]:

$$L = \frac{1}{K} \sum_{i=0}^K L_i. \quad (20)$$

The network throughput is mainly based on the maximum number of packets delivered in a specific interval and is determined via the equation given below [20, 42]. In Equation (21), the total number of received flits, i.e., the total number of delivered flits to the destination and cycles, is the number of simulation cycles between injection of the first message into the network and reception of the last message delivered to the destination node:

$$TP = \frac{\text{Total received flits}}{\text{Number of nodes} \times \text{Total cycles}}. \quad (21)$$

**5.3. Simulation Results.** To evaluate the proposed method, parameters of throughput, average packet latency, and

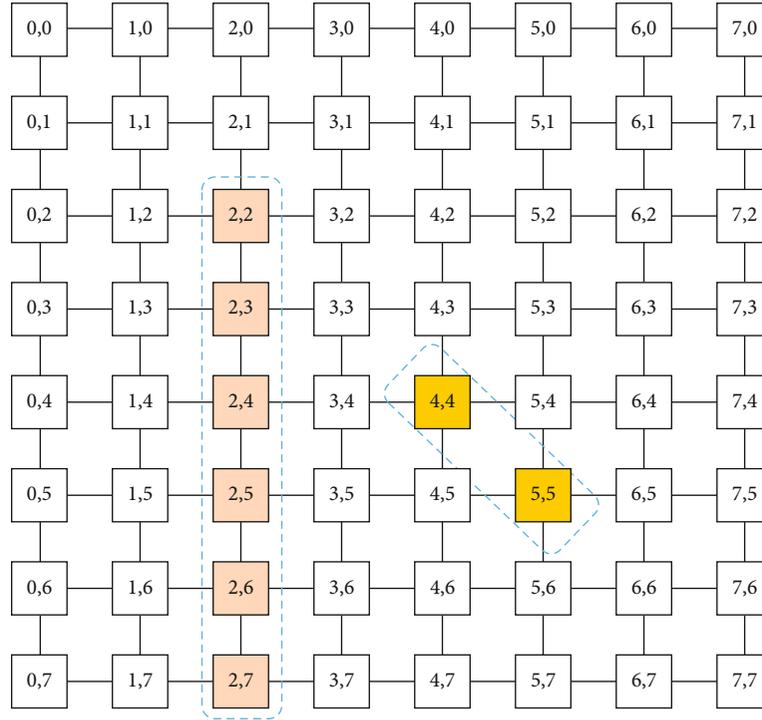
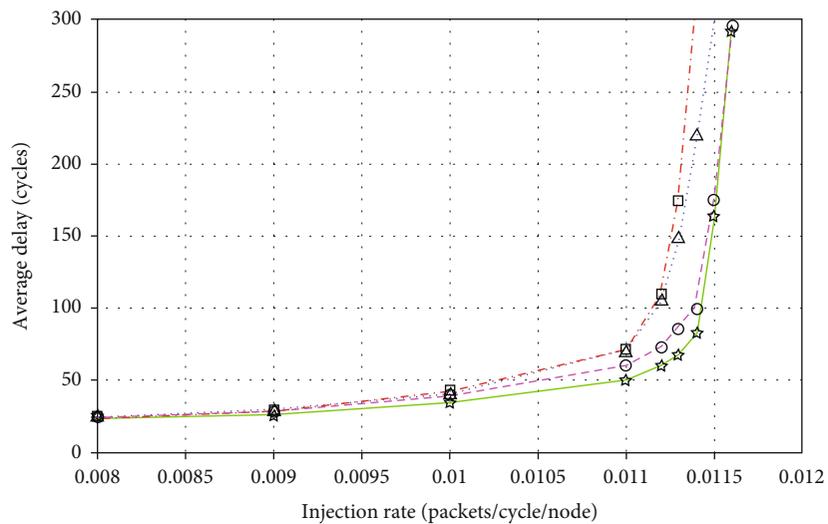


FIGURE 11: Performance of ScRN algorithm in  $8 \times 8$  mesh.

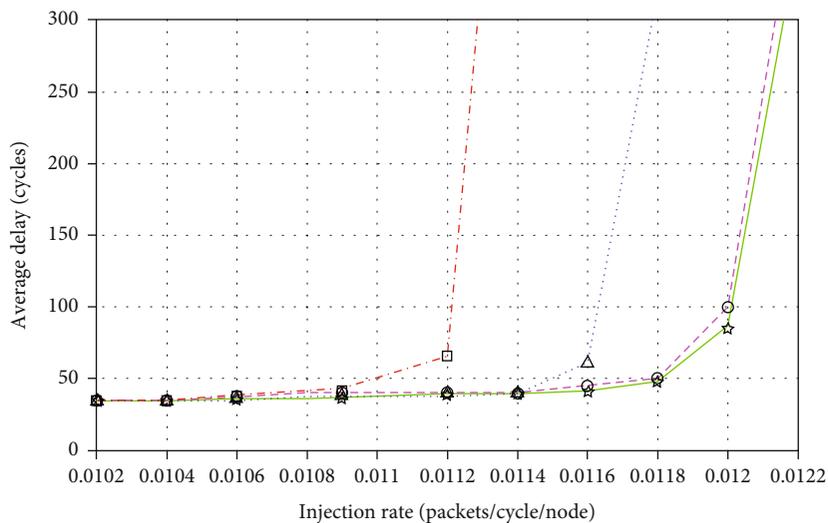
energy consumption level were taken into account for each scenario in different modes of packet injection. Figure 12(a) shows the simulation results for average latency of the packets in the transpose traffic pattern. As shown in Figure 12(a), all algorithms are in the same level of delay in the first three points; however, in other points, the ScRN algorithm performed better than other algorithms. The improvement in average latency of the packets through the proposed ScRN algorithm is 34.64%, 11.85%, and 32.81% compared with NoP, DICA, and RCA, respectively. Figure 12(b) presents the simulation results for the average latency of the packets in the random traffic pattern. As indicated, the ScRN algorithm outperformed other algorithms at every point. The improvement in average latency of the packets using the proposed ScRN algorithm was 46.56%, 6.03%, and 22% compared with NoP, DICA, and RCA, respectively. Figure 12(c) shows the simulation results for average latency of the packet in the hotspot-center traffic pattern. As shown in the figure, the average latency of the packets using ScRN was improved by 20%, 5.92%, and 13.20% in comparison with NoP, DICA, and RCA, respectively. Figure 12(d) shows the simulation results for the average latency of the packets in the hotspot-row traffic pattern. As seen in this figure, improvement for average latency of the packets through the proposed ScRN algorithm was 15.51%, 6.87%, and 22.41% compared with NoP, DICA, and RCA, respectively.

In the performed evaluations, the ScRN output selection algorithm had a lower average latency than all other algorithms. The reason behind this is the usage of channel congestion information and the selection of packet output depending on the traffic type. According to Figures 12(a) to 12(c), in transpose, random, and hotspot-center patterns,

the RCA algorithm, succeeding ScRN and DICA algorithms, and better than NoP had lower average packet latency, which is due to having more global congestion information. However, in Graph 12d, the RCA algorithm performed worse than other algorithms. This is because the selection process is made by putting separate values for traffic in all four quarters of the network. By using this, the RCA algorithm has access to additional congestion information, which is off the short route associated with node addresses of source and destination and utilized it in output selection [41]. Moreover, under hotspot-row traffic, due to the sequence of nodes with hotspot traffic in the same row, this congestion can accumulate throughout a row. Therefore, it can easily cause the output selection function to carry out an unfair selection during the time that hotspot nodes are off the short routes. This unfair selection can cause more congestion and higher packet latency. On the other hand, using congestion information on the route, ScRN can effectively prevent packets from passing through congested routes, which leads to decrease in the average packet latency and improvement in network performance [35]. Table 5 shows the level of improvement in the ScRN algorithm for random, transpose, hotspot-center, and hotspot-row traffic patterns. In this table, the ScRN delay is measured in an injection point which is not network saturated. As can be seen in the table, for all scenarios, including the hotspot-center scenario, our algorithm performed better than similar algorithms because in this scenario, there are important parts such as computing unit, memory unit, and control unit, so the percentage traffic is higher than other scenarios. Selecting the best output channel, the packets arrive at their destination with less delay. On average, the proposed ScRN

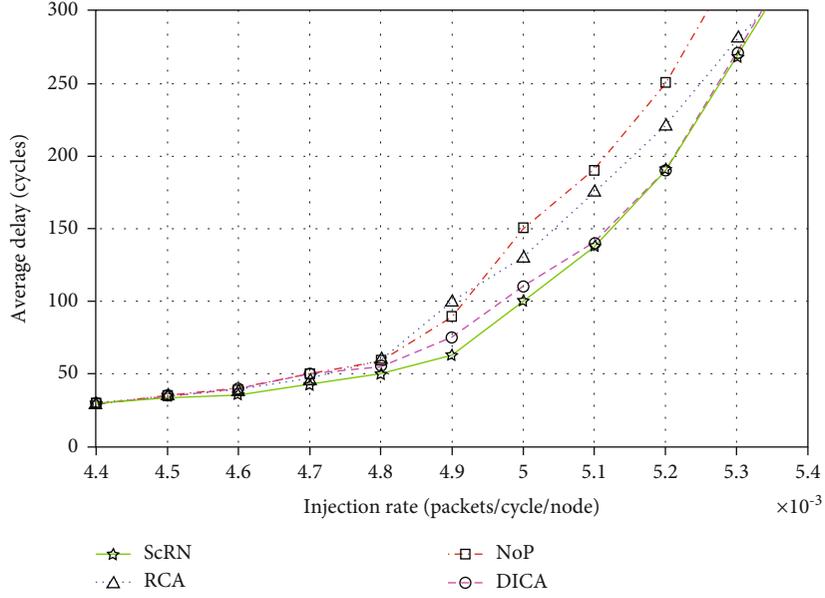


(a)

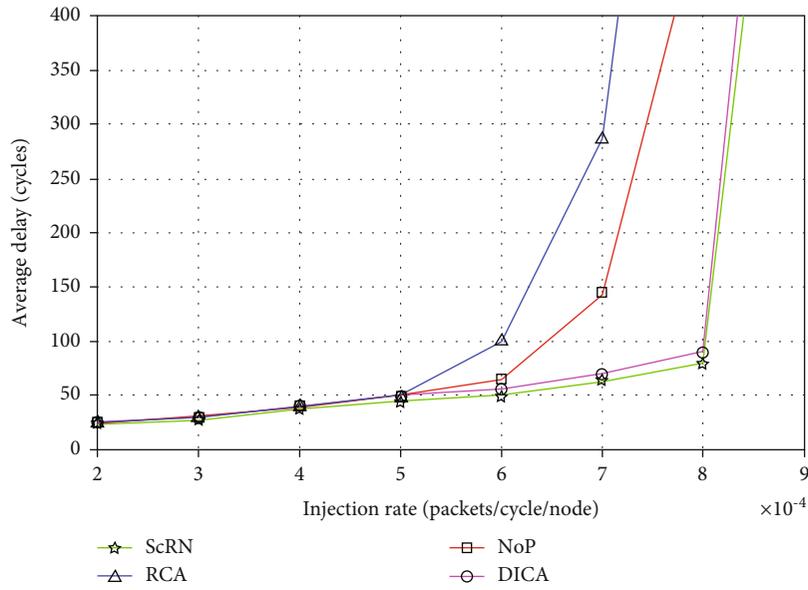


(b)

FIGURE 12: Continued.



(c)

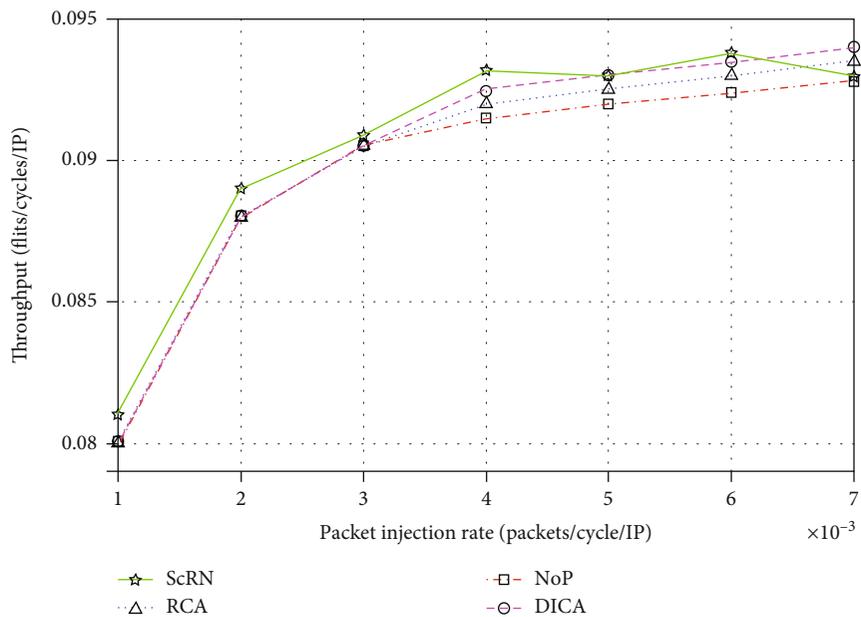


(d)

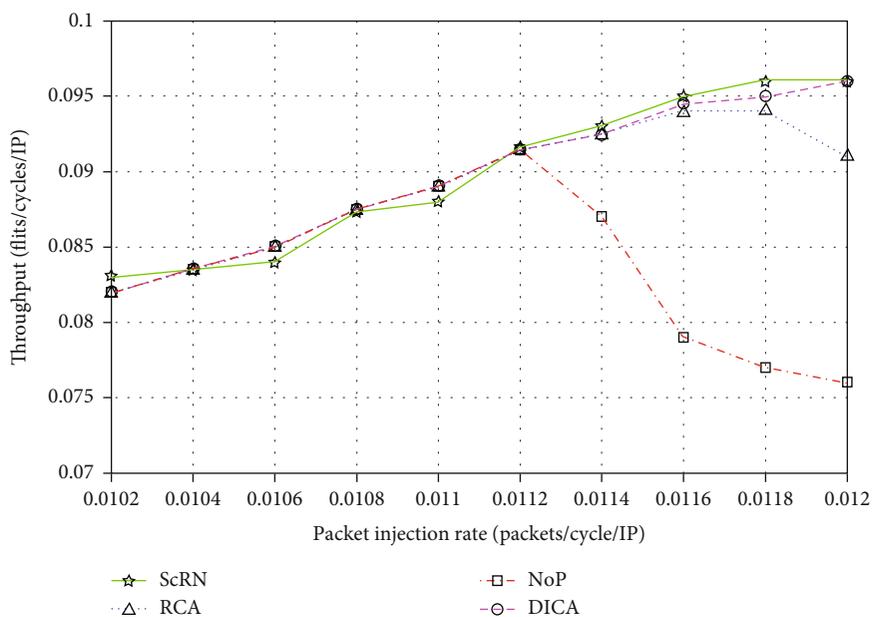
FIGURE 12: (a) Comparison of average packet latency in the transpose traffic pattern. (b) Comparison of average packet latency in the random traffic pattern. (c) Comparison of average packet latency in the hotspot-center traffic pattern. (d) Comparison of average packet latency in the hotspot-row traffic pattern.

TABLE 5: Improvement level of average latency of packets in the proposed ScRN algorithm.

Traffic patterns	Packet injection rate (packet/cycle/node)	Average latency (cycles)				Latency reduction by ScRN		
		DICA	RCA	NoP	ScRN	Versus DICA	Versus RCA	Versus NoP
Random	0.0114	36	42	866	28	22.22%	33.34%	96.76%
Transpose	0.0113	84	145	174	86	2.38%	40.68%	50.57%
Hotspot-center	0.005	109	132	147	94	13.76%	28.78%	36.5%
Hotspot-row	0.0007	70	285	146	75	7.14%	73.68%	48.63%
Average latency reduction						<b>11.37%</b>	<b>44.12%</b>	<b>58%</b>

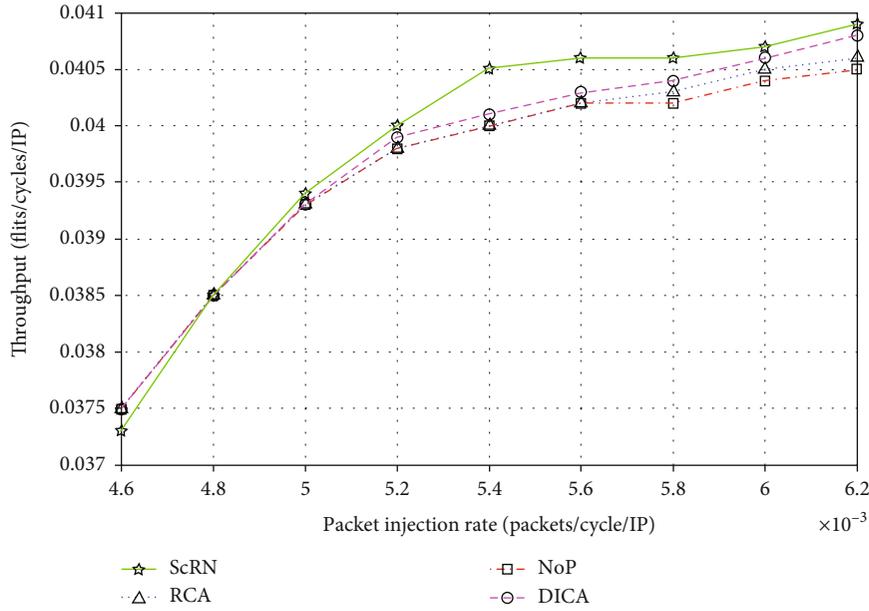


(a)

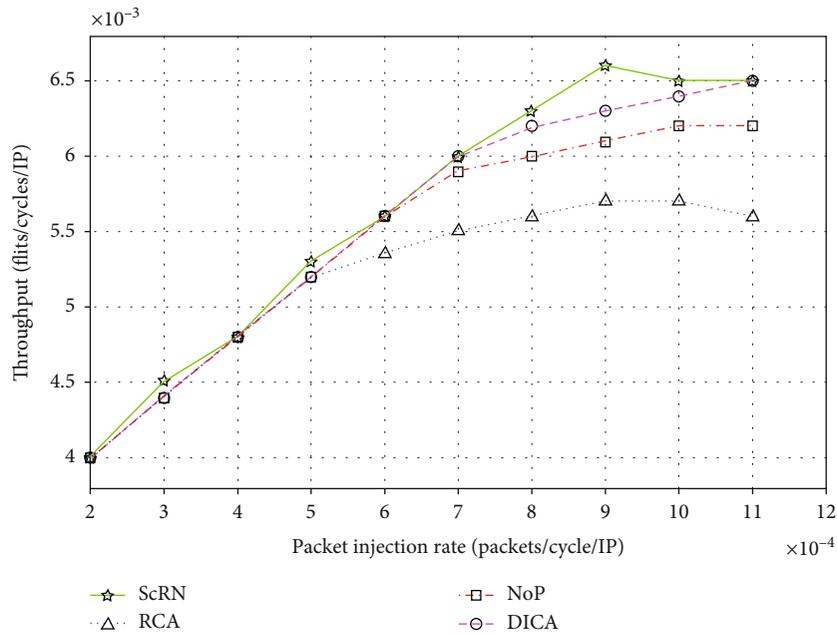


(b)

FIGURE 13: Continued.



(c)

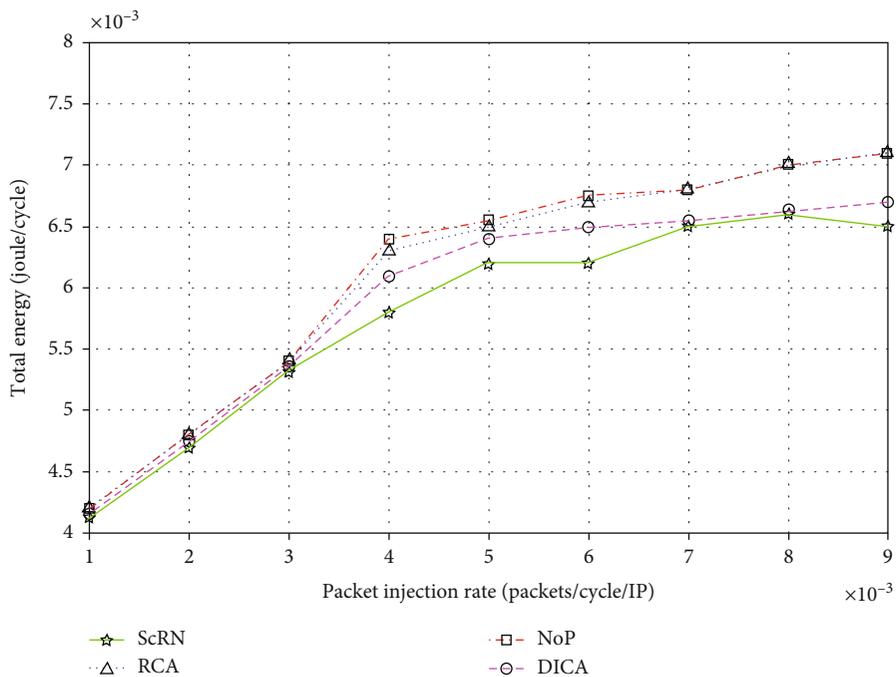


(d)

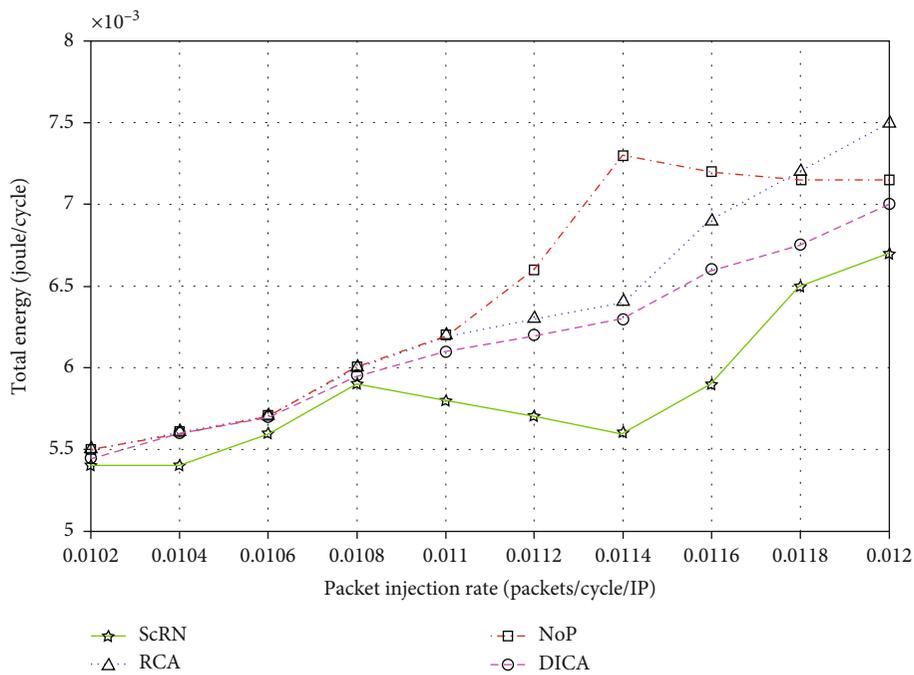
FIGURE 13: (a) Comparison of network throughput in the transpose traffic pattern. (b) Comparison of network throughput in the random traffic pattern. (c) Comparison of network throughput in the hotspot-center traffic pattern. (d) Comparison of network throughput in the hotspot-row traffic pattern.

TABLE 6: Comparison of the improvement level of throughput in saturated point for the proposed algorithm.

Traffic patterns	Saturation throughput (packet/ns/node)				ScRN improvement		
	DICA	RCA	NoP	ScRN	Versus DICA	Versus RCA	Versus NoP
Random	0.0117	0.0113	0.0106	0.0123	5.12%	8.85%	16.03%
Transpose	0.0114	0.01	0.01	0.0111	2.63%	11%	11%
Hotspot-center	0.0050	0.0049	0.0048	0.0056	12%	14.28%	16.67%
Hotspot-row	0.0008	0.0005	0.00059	0.0009	12.50%	80%	52.54%
<b>Average improvement</b>					<b>8.06%</b>	<b>26.49%</b>	<b>24.06%</b>



(a)



(b)

FIGURE 14: Continued.

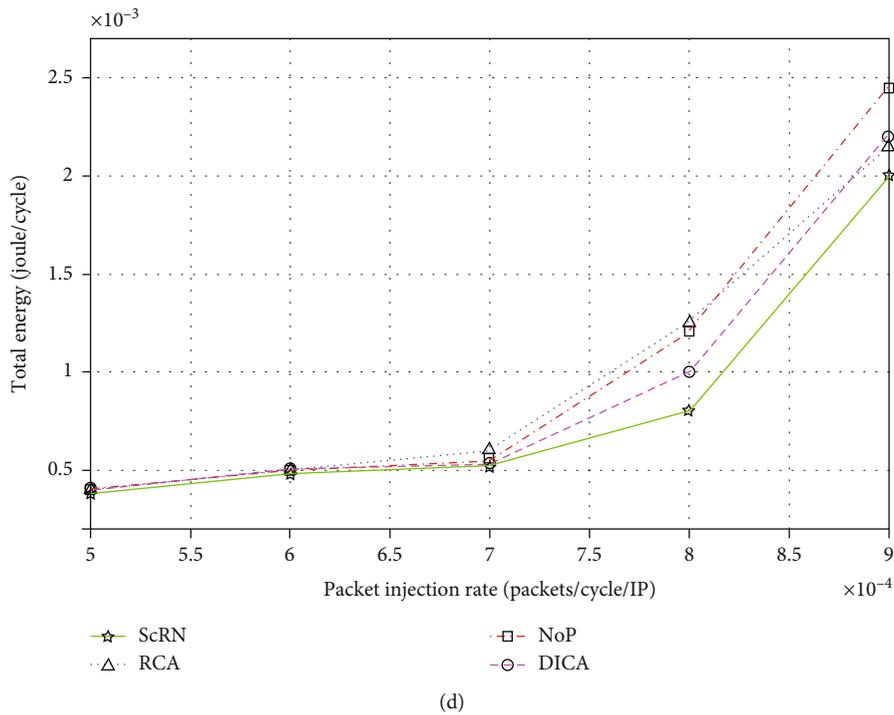
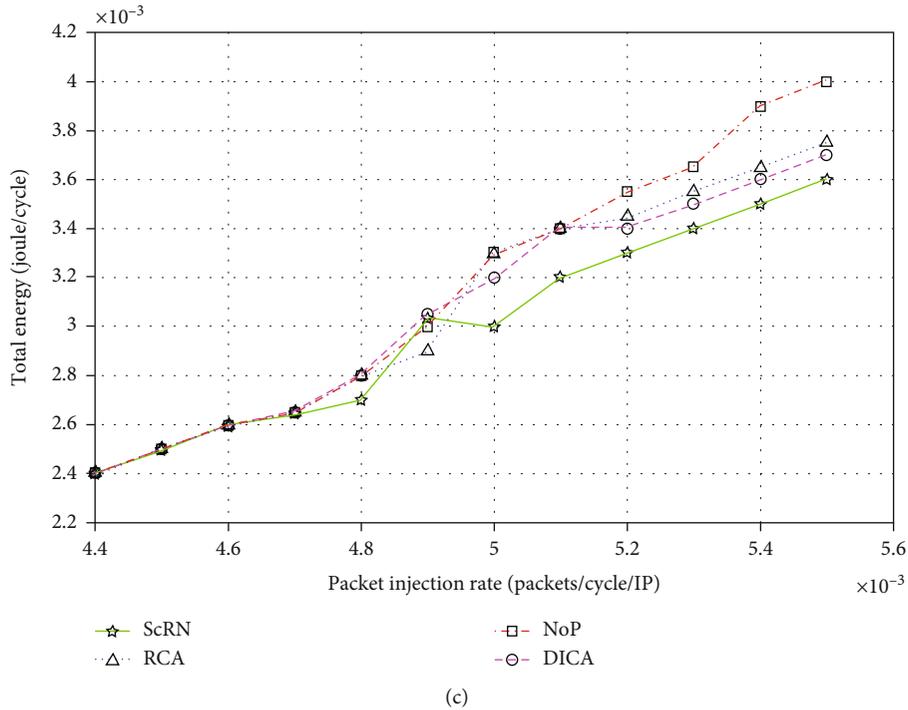


FIGURE 14: (a) Comparison of energy consumption in the transpose traffic pattern. (b) Comparison of energy consumption in the random traffic pattern. (c) Comparison of energy consumption in the hotspot-center traffic pattern. (d) Comparison of energy consumption in the hotspot-row traffic pattern.

algorithm was able to improve the average latency from 11.37% to 58% compared with other algorithms.

Figures 13(a) to 13(d) present the throughput results for all traffic patterns. Simulation results prove that improvement in average delay can enhance the throughput. As can be concluded from the analyses, the ScRN selection strategy

caused a reduction in average packet latency and an increase in throughput for all traffic patterns. This improvement is due to a more uniform distribution of traffic compared with other algorithms and also utilization of local and nonlocal congestion information, which led to more comprehensive information associated with network conditions.

TABLE 7: Comparison of reduction level in energy consumption for the proposed ScRN algorithm.

Traffic patterns	Packet injection rate (packet/cycle/node)	Average energy consumption (cycles)				Energy consumption by ScRN		
		DICA	RCA	NoP	ScRN	Versus DICA	Versus RCA	Versus NoP
Random	0.0116	0.0066	0.0069	0.0072	0.0059	10.60%	18.05%	14.49%
Transpose	0.0113	0.0065	0.0067	0.00675	0.0062	4.61%	8.14%	7.46%
Hotspot-center	0.00506	0.0035	0.00355	0.00365	0.0034	2.85%	6.84%	4.22%
Hotspot-row	0.0009	0.0022	0.00215	0.00245	0.0020	9.09%	18.36%	6.97%
<b>Average energy consumption</b>						<b>6.78%</b>	<b>12.84%</b>	<b>8.28%</b>

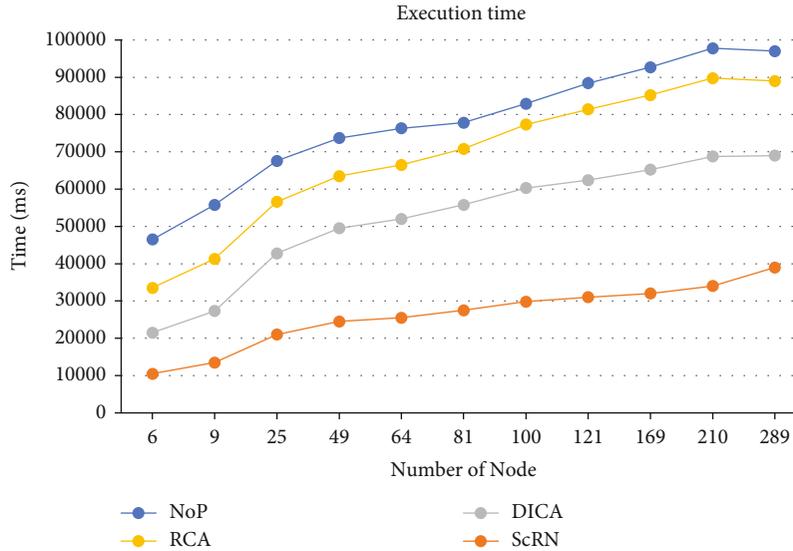


FIGURE 15: Execution time comparison for routing algorithms.

Table 6 presents the details of network throughput improvement for the ScRN algorithm. As can be seen in this table, the throughput for all scenarios has increased. Because the proposed algorithm first examines the amount of empty buffer of each neighboring node in question, and if the amount of buffer is less than normal or not empty, the weight of the node congestion increases, and thus the probability of selecting it decreases. This ensures that, as far as possible, congested or busy routes will have lower priority for closed selection and routing, resulting in reduced latency and increased throughput. The results of the evaluations in all scenarios show the same. Simulation results show that the proposed algorithm had one more throughput in saturated point in the range of 8.06% to 26.49% compared with other algorithms, or in other words, the network is saturated in a higher injection point.

The results show that the improvement level in network throughput can impact total energy consumption. In the present study, the Nirgam simulator is used to calculate energy consumption. As an architecture-level simulation tool, the Nirgam [5] is utilized to assess the energy consumption of the router's main operations like routing, incoming, and forwarding the flits and the output selection. SystemC which is a system description language based on C++ is used to develop the Nirgam simulator. In Nirgam,

Equation (11) has been used to estimate the average power, and each component has been applied in HDL. The ScRN strategy was implemented in VHDL, and synthesis was carried out through the Synopsys Design Compiler. Afterwards, ScRN average power in Synopsys Design Compiler was added to the Nirgam. In the Nirgam which is a signal level and cycle accuracy simulation, each wire is defined as a signal; therefore, energy consumption is computed by taking the overhead of ScRN logic and wiring. As shown in Figures 14(a) to 14(d), the mean network energy consumption using the proposed method is reduced for all traffic patterns. For example, the proposed ScRN algorithm had lower energy consumption with an average improvement of 5% compared with other algorithms in transpose traffic pattern. This decrease is the result of avoiding congested routes using the information provided by the ScRN strategy.

Table 7 presents the reduction in energy consumption of the ScRN algorithm for random, transpose, hotspot-center, and hotspot-row traffic patterns compared with other algorithms. As seen, by an increase in the injection rate, the average energy consumption rises as well. According to the results of previous evaluations, for all traffic scenarios, a reduction in energy consumption was also predictable, because in the proposed solution, the choice of crowded channels is always avoided, that is, routes with high

probability of congestion are in the lowest priority for routing. Avoiding crowded paths can create an optimal balance across the network, thus avoiding congestion of thermal bottlenecks due to congestion, and one of the important results is the reduction of energy consumption, which is shown by the results of the above diagrams. On average, the proposed ScRN algorithm was able to reduce energy consumption from 6.78% to 12.84% compared with other algorithms.

In the following, we compared the execution time of the proposed algorithm in comparison with other algorithms. Figure 15 shows the execution time and energy reductions for all routing algorithms, taking into account NoC size variation. When compared to other algorithms, ScRN achieve an average execution time reductions of over 80% while keeping energy savings to no more than 12% of the best results.

## 6. Conclusion

In this research, a novel output selection strategy called ScRN was proposed based on the parameters that affect the NoC performance. The basis of the proposed approach is using a traffic analyzer in which the traffic type is determined, and then, the best selection function associated with traffic type is used. Based on this, in this approach, if traffic pattern tends toward the destination of local traffics, the scoring-based selection strategy is activated, and otherwise, the strategy based on RCA and NoP is activated as proposed strategies for nonlocal traffics. In other words, in these conditions, the traffic pattern rate determines the local or nonlocal performance of the router. In the end, in the Nirgam simulation environment, and in different traffic scenarios, the proposed ScRN method was compared with and evaluated against other selection functions. Evaluations were done based on different traffic patterns. The most important features of this solution compared to previous works are as follows:

- (i) Use of new hybrid selection function to increase NoC performance
- (ii) Use of analyzer to evaluate local and nonlocal packet traffic
- (iii) Use of scoring strategy to select the best output channel

Based on this, considering the latency and throughput parameters, it was concluded that the proposed approach was effectively able to reduce the energy consumption and delay by analyzing the traffic type and determining the most appropriate function. As a result, it enhanced the throughput in NoCs. Also, Our proposed method includes the following limitations:

- (i) Calculate output channel scores can affect runtime
- (ii) Only works on adaptive algorithms and mesh topology

## Data Availability

Data is available on the following link: <https://s21.picofile.com/file/8443990850/codes.rar.html>.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] P. K. Sahu and S. Chattopadhyay, "A survey on application mapping strategies for network-on-chip design," *Journal of Systems Architecture*, vol. 59, no. 1, pp. 60–76, 2013.
- [2] A. B. Achballah and S. B. Saoud, "A survey of network-on-chip tools," <https://arxiv.org/abs/1312.2976>.
- [3] A. Abbas, M. Ali, A. Fayyaz et al., "A survey on energy-efficient methodologies and architectures of network-on-chip," *Computers & Electrical Engineering*, vol. 40, no. 8, pp. 333–347, 2014.
- [4] M. Trik, S. Pour Mozaffari, and A. M. Bidgoli, "Providing an adaptive routing along with a hybrid selection strategy to increase efficiency in NoC-based neuromorphic systems," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 8338903, 2021.
- [5] P. Mukherjee and S. Chattopadhyay, "Low power low latency floorplan-aware path synthesis in application-specific network-on-chip design," *Integration*, vol. 58, pp. 167–188, 2017.
- [6] J. Wu, D. Dong, X. Liao, and L. Wang, "Energy-efficient NoC with multi-granularity power optimization," *The Journal of Supercomputing*, vol. 73, no. 4, pp. 1654–1671, 2017.
- [7] P. Lotfi-Kamran, A.-M. Rahmani, M. Daneshalab, A. Afzali-Kusha, and Z. Navabi, "EDXY—a low cost congestion-aware routing algorithm for network-on-chips," *Journal of Systems Architecture*, vol. 56, no. 7, pp. 256–264, 2010.
- [8] X. Zhou, L. Liu, Z. Zhu, and D. Zhou, "A routing aggregation for load balancing network-on-chip," *Journal of Circuits, Systems and Computers*, vol. 24, no. 9, article 1550137, 2015.
- [9] L. Zhang, E. E. Regentova, and X. Tan, "Packet switching optical network-on-chip architectures," *Computers & Electrical Engineering*, vol. 39, no. 2, pp. 697–714, 2013.
- [10] G. Ascia, V. Catania, M. Palesi, and D. Patti, "Neighbors-on-path: a new selection strategy for on-chip networks," in *2006 IEEE/ACM/IFIP Workshop on Embedded Systems for Real Time Multimedia*, pp. 79–84, IEEE, 2006.
- [11] A. Osmani, J. B. Mohasefi, and F. S. Gharehchopogh, "Sentiment classification using two effective optimization methods derived from the artificial bee colony optimization and imperialist competitive algorithm," *The Computer Journal*, vol. 65, no. 1, pp. 18–66, 2022.
- [12] L. Liu, R. Ma, and Z. Zhu, "An encapsulated packet-selection routing for network on chip," *Microelectronics Journal*, vol. 84, pp. 96–105, 2019.
- [13] S. Ma, N. E. Jerger, and Z. Wang, "DBAR: an efficient routing algorithm to support multiple concurrent applications in networks-on-chip," *Proceedings of the 38th annual international symposium on Computer architecture*, vol. 39, no. 3, pp. 413–424, 2011.
- [14] P. Kullu and S. Tosun, "Energy-aware and fault-tolerant custom topology design method for network-on-chips," *Nano Communication Networks*, vol. 19, pp. 54–66, 2019.
- [15] N. Chatterjee, P. Mukherjee, and S. Chattopadhyay, "Reliability-aware application mapping onto mesh based Network-on-Chip," *Integration*, vol. 62, pp. 92–113, 2018.

- [16] J. Liu, J. Harkin, Y. Li, and L. Maguire, "Low cost fault-tolerant routing algorithm for networks-on-chip," *Microprocessors and Microsystems*, vol. 39, no. 6, pp. 358–372, 2015.
- [17] F. Bahman, A. Reza, M. Reshadi, and S. Vazifedan, "CACBR: congestion aware cluster buffer base routing algorithm with minimal cost on NOC," *CCF Transactions on High Performance Computing*, vol. 2, no. 3, pp. 297–306, 2020.
- [18] A. Mehranzadeh, A. Khademzadeh, N. Bagherzadeh, and M. Reshadi, "DICA: destination intensity and congestion-aware output selection strategy for network-on-chip systems," *IET Computers & Digital Techniques*, vol. 13, no. 4, pp. 335–347, 2019.
- [19] R. Suraj and P. Chitra, "PT-BAR: prioritized thermo-buffer based adaptive routing protocol for network-on-chip," *Procedia Computer Science*, vol. 93, pp. 324–335, 2016.
- [20] G. Ascia, V. Catania, M. Palesi, and D. Patti, "Implementation and analysis of a new selection strategy for adaptive routing in networks-on-chip," *IEEE Transactions on Computers*, vol. 57, no. 6, pp. 809–820, 2008.
- [21] A. Arora and N. K. Shukla, "A congestion controlled and load balanced selection strategy for networks on chip," *International Journal of Distributed Systems and Technologies (IJ DST)*, vol. 11, no. 1, pp. 1–14, 2020.
- [22] P. K. Donta, T. Amgoth, and C. S. R. Annavarapu, "Congestion-aware data acquisition with q-learning for wireless sensor networks," in *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pp. 1–6, IEEE, 2020.
- [23] M. Devanathan, V. Ranganathan, and P. Sivakumar, "Congestion-aware wireless network-on-chip for high-speed communication," *Automatika*, vol. 61, no. 1, pp. 92–98, 2020.
- [24] C. Sun, Y. Ouyang, and H. Liang, "Architecting a congestion pre-avoidance and load-balanced wireless network-on-chip," *Journal of Parallel and Distributed Computing*, vol. 161, pp. 143–154, 2022.
- [25] M. Trik, S. Pour Mozafari, and A. M. Bidgoli, "An adaptive routing strategy to reduce energy consumption in network on chip," *Journal of Advances in Computer Research*, vol. 12, no. 3, pp. 1–12, 2021.
- [26] W. Hu, H. Guo, K. Zhang, J. Liu, X. Liu, and Q. Shi, "An energy-efficient design of microkernel-based on-chip OS for NOC-based many core system," *The Journal of Supercomputing*, vol. 73, no. 8, pp. 3344–3365, 2017.
- [27] A. Alagarsamy, L. Gopalakrishnan, S. Mahilmaran, and S.-B. Ko, "A self-adaptive mapping approach for network on chip with low power consumption," *IEEE Access*, vol. 7, pp. 84066–84081, 2019.
- [28] N. L. Venkataraman, R. Kumar, and P. Mohamed Shakeel, "Ant lion optimized bufferless routing in the design of low power application specific network on chip," *Circuits, Systems, and Signal Processing*, vol. 39, no. 2, pp. 961–976, 2020.
- [29] J. Wang and D. Wang, "A smart protocol-level task mapping for energy efficient traffic on network-on-chip," *Microprocessors and Microsystems*, vol. 65, pp. 69–78, 2019.
- [30] J. M. Joseph, L. Bamberg, I. Hajjar, R. Schmidt, T. Pionteck, and A. Garcia-Ortiz, "Simulation environment for link energy estimation in networks-on-chip with virtual channels," *Integration*, vol. 68, pp. 147–156, 2019.
- [31] O. I. Khalaf, C. A. T. Romero, S. Hassan, and M. T. Iqbal, "Mitigating hotspot issues in heterogeneous wireless sensor networks," *Journal of Sensors*, vol. 2022, Article ID 7909472, 14 pages, 2022.
- [32] A. B. Gabis, P. Bomel, and M. Sevaux, "Application-aware multi-objective routing based on genetic algorithm for 2D network-on-chip," *Microprocessors and Microsystems*, vol. 61, pp. 135–153, 2018.
- [33] A. V. Salazar and G. Patiño, "Reducing dynamic energy in networks on chip," in *2019 Congreso Internacional de Innovación y Tendencias en Ingeniería (CONIITI)*, pp. 1–6, IEEE, 2019.
- [34] K. Ilamathi and P. Rangarajan, "Intelligent computation techniques for optimization of the shortest path in an asynchronous network-on-chip," *Cluster Computing*, vol. 22, no. S1, pp. 335–346, 2019.
- [35] J. Yong, Z. Lin, W. Qian, B. Ke, W. Chen, and L. Ji-fang, "Tree-based multihop routing method for energy efficiency of wireless sensor networks," *Journal of Sensors*, vol. 2021, Article ID 6671978, 14 pages, 2021.
- [36] Z. Bo, Y. Dong, J. He, and L. Dongming, "An energy-efficient one-shot scheduling algorithm for wireless sensor networks," *Journal of Sensors*, vol. 2021, Article ID 9999403, 15 pages, 2021.
- [37] E. L. Prasad, M. N. Giri Prasad, and A. R. Reddy, "HSRDN: high-speed router design for various NoC topologies," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 281–290, 2020.
- [38] H. Zhao, F. Zhang, L. Chen, and M. Lu, "A method of fast evaluation of an MC placement for network-on-chip," *Journal of Circuits, Systems and Computers*, vol. 30, no. 7, article 2150115, 2021.
- [39] M. R. Naqvi, "Low power network on chip architectures: a survey," *Computer Science and Information Technologies*, vol. 2, no. 3, pp. 158–168, 2020.
- [40] T. S. Das, P. Ghosal, and N. Chatterjee, "VCS: a method of in-order packet delivery for adaptive NoC routing," *Nano Communication Networks*, vol. 28, article 100333, 2021.
- [41] M. Trik and B. Boukani, "Placement algorithms and logic on logic (LOL) 3D integration," *Journal of Mathematics and Computer Science*, vol. 8, no. 2, pp. 128–136, 2014.
- [42] M. Jalalifar and G.-S. Byun, "An energy-efficient multi-level RF-interconnect for global network-on-chip communication," *Analog Integrated Circuits and Signal Processing*, vol. 102, no. 1, pp. 131–143, 2020.