



an Adaptive Routing Strategy to Reduce Energy Consumption in Network on Chip

Mohammad Trik¹, Saadat Pour Mozaffari^{✉2}, and Amir Massoud Bidgoli³

^{1,2}Department of Computer Engineering, North Tehran Branch, Islamic Azad University, Tehran, Iran

³Computer Engineering and Information Technology Department, Amirkabir University of Technology, Tehran, Iran

trik.mohammad@gmail.com¹, am_bidgoli@iau-tnb.ac.ir²
saadat@aut.ac.ir³

Received: 2021/01/27; Accepted: 2021/11/05

Abstract

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 this study, the performance improvement of Networks on Chip (NoC) is investigated by introducing an optimal selection strategy. One of the most important features of NoC is an efficient and continuous routing. The effect of any routing algorithm depends on the chosen strategy. Accordingly, in the proposed approach, packet traffic is first examined using an analyzer and then based on the number of steps, it is determined whether packets are local or not. Finally, packets are sent through the best output channel using the Regional Congestion Awareness (RCA) selection strategy for local traffic and Neighbors-on-Path (NoP) for non-local traffic. Using a simulation, it is shown that the proposed approach significantly increased the performance. Experiments show that in compared to BufferLevel and Random strategies, this method remarkably reduces the average delay and energy consumption.

Keywords: Network on Chip, Adaptive routing, Energy consumption reduction

1. Introduction

The increase in the number of components in a system on chip (SoC) coupled with the growth of interference problems caused by the bus system led to the appearance of NoC. Using millions of transistors on a chip is possible, due to the size of transistors. A large number of processors also can be assembled on a single chip. In such systems, the connection between processors are important, and system-on-chip (SoC) and the Network on Chip (NoC) are two implementation techniques for these systems. Network on Chip or (NoC) is a communication subsystem within an integrated circuit, which typically provides the relationship between IP cores in a system within a single processor. NoC's can cover the synchronous and asynchronous clock domains, or use asynchronous logic without clock. NoC technology uses network theory and modern communication methods in chips and brings substantial progress towards the connection based on early bus and crossbar. NoCs improve the scalability in SoCs and optimizes the energy usage in complex SoCs more than other designs [1, 2,37]. Figure 1a shows a simple design of a Network on Chip. Generally, the connection between processing elements are connected by a bus. By increasing the number processors on a chip with a large number of processing elements, the bus becomes a bottleneck in terms of scalability and energy consumption.

Therefore, the idea of on-chip network which includes processing elements and routers that are connected by links and communicate with each other by sending packets was raised. Network-on Chip is an efficient communication architecture from the power perspective for systems on chip with several tiles [4]. The Network-on-Chip has eliminated the disadvantages of the System-on Chip and has high performance and reusability. Figure 1b shows a schematic of a system-on-chip that each of IPs can be different cores. A System-on-chip can include various elements such as a CPU, input and output units, and various types of memory. However, SoCs have disadvantages such as lack of reusability, low scalability, design complexity, and long time to be achievable to the market [3].

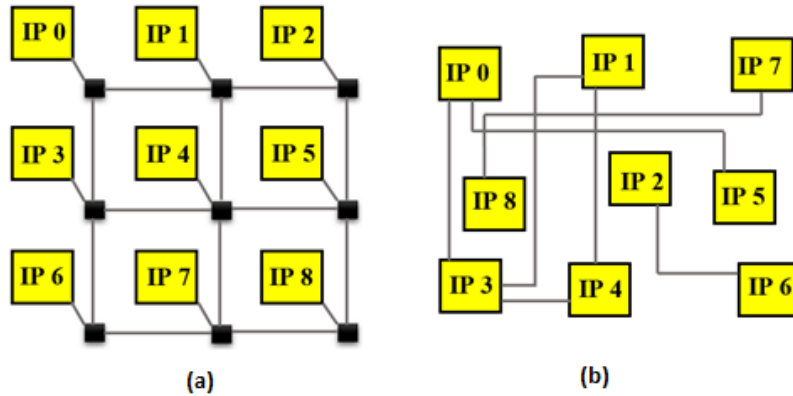


Figure 1. Network on chip and system-on-chip schematic

In the adaptive algorithms the calculated path is stored in the packet header and used in the middle nodes to hold the channels. When the routing algorithm returns a set of output channels, the selected function is used to select the output channel to which the packet is sent. Since the appropriate pattern selection strongly affects the overall performance of the selected routing, the routing algorithms should ensure that the deadlock is not created in the network under the algorithm and no wander is created for the packets. A deadlock occurs when each packet the channel that is available to another packet to continue its path; wandering also means that the packet does not arrive at the destination for long and unreasonable time. 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. In this case, a traffic analyzer is used that determines the type of function according to the type of local or non-local traffic. As a result, in this case, according to the type of traffic, the appropriate strategy can be determined with it. Accordingly, in this research, a hybrid function with traffic analyzer for adaptive routing algorithms is presented, in which, in addition to increasing the efficiency of NoC, it would be possible save power consumption by creating the balance in this infrastructure [4,39].

1.1. Contribution

In previous work, various selection strategies have been proposed to improve routing algorithms, each of which poses challenges in research results. For example, in reference [36,38], using a virtual circuit switch, routing is minimized and thus energy consumption is reduced. In [15,12,35] the selection function is presented based on input and output choices and NOP technique in which they have been able to reduce energy consumption. Also in the reference [32] by using a selection strategy in XY routing to achieve improvements in reducing latency. In the proposed method, we have first separated a number of calculations that can be done offline from the main processing steps. In this way, the processing overhead can be reduced each time it is run. In the next step, a selection strategy is presented according to the traffic situation. In this case, a traffic analyzer is used, which determines the type of selection function according to the type of local or non-local traffic. As a result, in this case, according to the type of traffic, the appropriate strategy can be determined with it. In this way, in addition to reducing energy consumption, other parameters such as latency and congestion can be reduced compared to other solutions.

1.2. Paper Organization

Our paper is organized as below. In the next section, related works is stated for the previously used algorithms in NoC along with selection functions. In section 3, the suggested combined method is stated to propose a combined selection function. In section 4, the results of analyzing the suggested model in different scenarios are shown, Finally, Section 5 concludes this paper.

2. Related Works

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 preformed works in these subjects in the following sections. Adaptive routing algorithms will be separated into two subgroups: partially adaptive routing and fully adaptive routing algorithms. Within the previous case, to avoid the construction of deadlock, the algorithm confines packets to be delivered toward some directions by utilizing turn models [5,0]. Within the further case, the algorithm can route packets toward all paths. The fully adaptive routing algorithm furnish high path variety to assist congestion; however, it causes the problem of livelock and deadlock. Many adaptive routing algorithms do not perform well due to their local. In recent years, adaptive routing algorithms have been proposed that use local or non-local information for NoC. Due to the congestion of information in each router, Regional congestion aware NoC architectures can be divided into two categories: In some architectures, a router control the status of the entire network, while in other architectures the routers are aware of the status of part of the network[6]. Regional Congestion Awareness (RCA) uses a lightweight network to accumulation and dissemination congestion information [10]. Other studies [11,13,36,37] also provide information that global congestion has been reviewed and collected. Another type of architecture in NoC that examines only a few non-local nodes instead of all nodes is the NoP architecture, which is based on destination-based adaptive routing. In [12], 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. Buffer depth information The current node as well as the neighbor's buffer depth that is in the path of the message candidate is considered by the NOP. In this architecture, non-local congestion information is confined to the distance of a hop from the current router. Locally adaptive routing methods are made based on local congestion information. authors use of the number of gratis VCs as a congestion metric and plectron the port with the higher number of gratis VCs [14]. A selection strategy named EnPSR is introduced in [7] for better performance of the network. This approach has the ability to reduce the hardware overhead thorough 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 [8]. 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 non-minimum routing algorithm called LEAR, which avoids congested routes from source to destination [9].

In [15], 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 another study, researchers [16] used buffer as a metric congestion in neighboring nodes, while in [17] the output queue length was used for the same application. In order to establish traffic control and balance in [18], a selection function based on the fuzzy controller is introduced. Traffic estimation for free packet routing is one of the properties of this method. . In [19], a selection strategy called CBWA is proposed to avoid congestion. This method attempts to establish traffic distribution in network routes to create load balance using router data, bandwidth capacity, and the number of sent messages. In recent studies, a small number of globally aware adaptive routing methods have been proposed. in Ref.[20] provides a NoC architecture called Adaptive Toggle Dimension (ATDOR) that can create a secondary network for transmitting congestion information from one node to another, from which to choose between XY or YX DOR for both pairs Use the source and destination on the network.

In another study, a method called DAR was proposed, which uses a separate network to communicate with other networks to be aware of congestion information[21,22]. In this method, each node determines the amount of traffic load that must be distributed among the candidate output ports for a specific destination. In the Ref[36], it uses the HRA (Heuristic based Routing Algorithm) method for traffic distribution, which can optimize latency, increase reliability and fault tolerance. Table 1 shows a summary analysis of the routing plans in view of the Selection Strategy, Reduced latency, throughput, energy efficiency, and congestion avoidance. Additionally, the heading "Time" describe to whether the specific adaptive algorithm deals with the congestion in "Time"; since, "Local" and "Global" shows if the algorithm utilizes the information on the local or the global congestion, relatively. Also for Table 1 and the succeeding tables, the "√" and "×" tokens shows if a particular characteristic/parameter is effectuate or not, respectively; whereas, "-" represents that a particular characteristic/parameter is not investigated. Therefore, our solution has the ability to solve all the challenges of the methods presented in the Table 1.

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

Work	Selection Strategy	Reduced latency	Energy efficiency	throughput	Congestion avoidance		
					Time	Local	Global
[7]	EnPSR	√	√	—	√	×	—
[8]	DBAR	—	√	√	×	√	√
[10]	NoP	√	√	—	√	√	×
[12]	CACBR	—	√	√	×	—	×
[15]	NoP-OE	—	√	√	×	×	×
[16]	DBAR,RCA	√	√	—	—	—	—
[20]	ATDOR	√	√	—	—	—	√
[36]	HRA	√	√	—	×	√	×

3. The Proposed Method

The proposed strategy that focuses on the selection algorithm develops within the framework of the infrastructure sections of these systems. Initially, the main requirements of the proposed solution are examined and the main core of the research, which is the expression of the traffic analyzer and hybrid selection function, is presented and its capabilities are expressed. Part of the calculations of the selected strategy is done before the simulation operation; the offline execution of the calculation allows the algorithm to reduce its delay due to reduced computational time.

3.1. Offline Computational

Parts of the selection of computational strategies in this research, such as the link connection and the equality of resistances are computed offline and before the algorithm is implemented which is explained below.

Link Contention (C): Link contention is referred to as traffic value which can pass through a specific link based on the communications provided in the communication graph [23]. If P_i is a path for arbitrary communications, ρ_{comm} is the set of all possible paths, t_{comm} is the traffic generated by communications, and n_{comm} is the number of all communications that are specified in the program traffic and extracted from the communication graph, so link contention L can be expressed as follows:

$$CL = \sum_{comm=1}^{n_{comm}} (\mu * t_{comm})$$

$$\mu = \begin{cases} 1: \exists \rho_i \in \rho_{comm} : L \in \rho_i \\ \text{else} : 0 \end{cases}$$

Equivalent Resistance (ER): By defining this concept (ER) for each given communication, and using the electrical concepts of the Kirchoff law, each node in the topology is considered as a circuit node and each link is deemed as a resistor with a volume equal to the contention of that link [19, 20].

3.2. Requirements of online function

During the execution of the algorithm, other additional available data are also used for routing, which is as follows:

- Free buffer rows [d] (B): The number of rows in the input buffer is the neighbors adjacent to path d, in which d is one of the north, south, east, and west directions.
- Instantaneous power (Δp): The instantaneous power is the difference between the power consumed by the router at t and t-1, where t is the moment the final output channel is selected for the packet;

$$\Delta p = \text{power}(t) - \text{power}(t-1)$$

Using this information, a better estimation can be obtained for the traffic load of the network and a more appropriate decision would be made at the moment of the next hop.

3.3. Traffic analyzer

In order to avoid deadlock in the routing algorithm and reduce the delay time, an analyzer and selection function are added to the routing algorithm, so that it could be used to select the best outlet based on the local or non- local nature of the packet. Accordingly, the analyzer first extracts the destination address of each packet that is routed through the router in each T-cycle and examines its data. For this purpose, two 5-bit counters are used to determine the local or non- local nature of requests in the router [21]. If the intended destination of the packet is two hops away or higher from the current router, it is considered as a non- local packet, otherwise it is considered local. The analyzer calculates packet hops periodically and accordingly it updates the counter of locality and (L) and non- locality (N) of the packets. This information is sent to the switcher to decide on the selection strategy. The counter is cleared at the end of every T cycle. The pseudo-code for determining the traffic type is shown in Fig. 2.

```

for every T clock cycles do
  L and N value from analyzer;          Catch
  Compute  $x = N/(L+N)$ ;
  if  $x = x < 0.3$  then
    ;Switch to NoP
    else
    ;Switch to RCA
  end

```

Figure 2. A pseudo-code for determining the type of traffic

In fact, with the help of the traffic analyzer it would be possible to obtain appropriate information about the rate of traffic and its convergence to local or non-local traffic [22], and then, in the next step, the routing operations can be done accordingly. Figure 3 shows the schematic view of the solution.

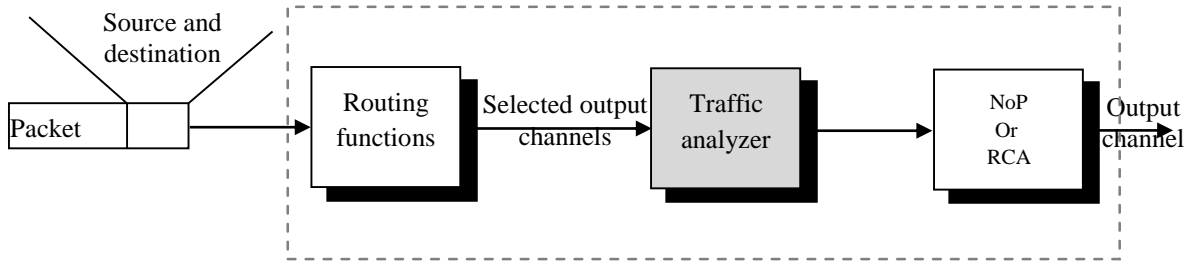


Figure 3. The solution's schematic view

Accordingly, at the end of every 32 cycles, the traffic pattern is determined by the analyzer's output, and the local traffic rate is calculated as non-local (x). If $x \geq 0.3$, then traffic is non-uniform and the RCA algorithm [16] must be used, otherwise the NoP [12] selection strategy will be used. In other words, if the traffic pattern is oriented towards local traffic destination, the NoP-based selection strategy is activated; otherwise, the RCA-based strategy will be activated as a proposed strategy for non-local traffic. The general algorithm for switching operation based on the traffic analyzer is presented in Fig. 4. The data input to this algorithm is of local or non-local data type and the related output is also the best strategy. It should also be noted that since the analyzer and the switch only take the router data at any one time, there is no additional overhead in network communications.

```

Data: Packet hops (pkt_dst_hops), (Initializing: L=0, N=0)
Result: Local and Non-Local value (L=Local value, N=non_Local value)
2 then =if pkt_dst_hops >
N++;
else
L++;
end

```

Figure 4. Pseudo-code related to perform switching operations

3.4. Formulation of the solution

When the routing function receives multiple outputs, by reviewing the reservations table, the selected algorithm for each of these outputs checks whether the channel is available to transfer the packet (header flit) or its reserved by the other header flits [25]. The channel must be available so that the selected score is calculated and eventually the channel with the highest score is selected. If more channels have the same score, the first one will be chosen. The calculation of the score is done by the following formula:

$$\text{Score}[d] = \alpha \times \text{Psel}[d] + \beta \times (\text{B}[d]/\text{max_buffer_size} + \gamma) \times (\Delta P / \text{max_power}) \quad (1)$$

α , β , and γ are the weight factors for the probability of selecting links, open buffers, and instantaneous power consumption. These coefficients result in the full dynamic adaptability of the selected algorithm and thus the set values will be at their best state [26]. Since open buffers (B) and instantaneous power consumption ($p\Delta$) have different

units, they are normalized using max-buffer-size and max-power factors. Also, since P_{sel} is within the range (0 and 1), there is no need for normalization. Then using the following formula the score of the adaptive routing functions and all possible values of α , β and γ are evaluated and the best coefficients are obtained for each of the routers:

$$\alpha + \beta + \gamma = 1 \ ; \ \alpha=0,0.1,\dots,1 \ ; \ \beta = 0,0.1,\dots,(1-\alpha) \ ; \ \gamma = 1-(\alpha+\beta) \quad (2)$$

For example, the best values of α , β , and γ in even-odd routing are 0.3, 0.4 and 0.3 under the MMS traffic scenario. Another important feature of this algorithm is its adaptability with any network topology [27].

4. Experiments and Simulation Environment

In this section, a platform for simulation and the structure of the on-chip networks is provided. This section provides a platform for simulating the structure and framework of NoC. A Nirgam simulator is used to evaluate the suggested algorithm whose capabilities are listed in Table 2 [13,15,31]. The main components in this simulator are routers, processing elements, links and buffers [28, 29]. Moreover, the configuration parameters for the analysis and simulation of the suggested method is given in Table 3. Also the average delay, maximum delay, and power consumption are considered as the efficiency criteria. Delay is assumed as the time between the entering the header flit to the network and the arrival of the tail flit to the destination node. In order to evaluate the proposed method, Random, RCA and NoP strategies have been compared [30]. Here the results of the study are shown in various traffic scenarios.

Table 2. 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 3. Simulation parameters

Parameter	Configuration
Network size	8*8 Mesh
Schemes	Random, RCA[13], NoP[15],Proposed
Packet size	8flits
Reset_time cycles	5000
Simulation time	10

4.1. Evaluation criteria

Average delay, maximum delay, and power consumption, are selected as performance criteria. Delay, is the time between entering the Fleet header to the network until the destination node receives the header. The average delay is calculated according to the following equation:

$$D = 1/n \sum_{i=1}^n D_i$$

In which, n is the total packets that have reached to the destination and D_i is the delay of the i^{th} packet. The maximum delay (m) is defined as follows:

$$M = \max(D_i) : i = 1, 2, \dots, n$$

Power is the amount of energy that is wasted by routers and links. The selection strategy is evaluated using Multimedia system traffic scenario [10]. Experiments are done for mesh sizes of 5×5 and 4×4 and nodes create 6 fleet packets according to exponential distribution and rate $0 < \text{pir} \leq 1$, which pir is the packet injection rate. For example if $\text{pir} = 0.1$, it means that each node sends the packets in 10 clock cycles. FIFO buffers with the capacity of 4 rows are considered. Each step, the simulations have been performed 10000 cycles as preparation time finally, 100000 clock cycles. To validate the accuracy of the results, the simulation of each pir nodes are repeated several consecutive times and the average of the results is considered as final result.

4.2. Evaluation result

In order to evaluate, the proposed method is performed with odd-even algorithm and is compared with the Random and Buffer Level strategies [11, 12]. As can be seen in figures 5 to 9, the average delay is dramatically reduced more than other methods. It is reduced by 16% and 14% compared to Buffer Level and Random respectively. On the other hand, since a portion of calculations are done in offline manner, the energy consumption is reduced by 18% compared to Buffer Level and 21% compared to Random which is shown in figure 5.

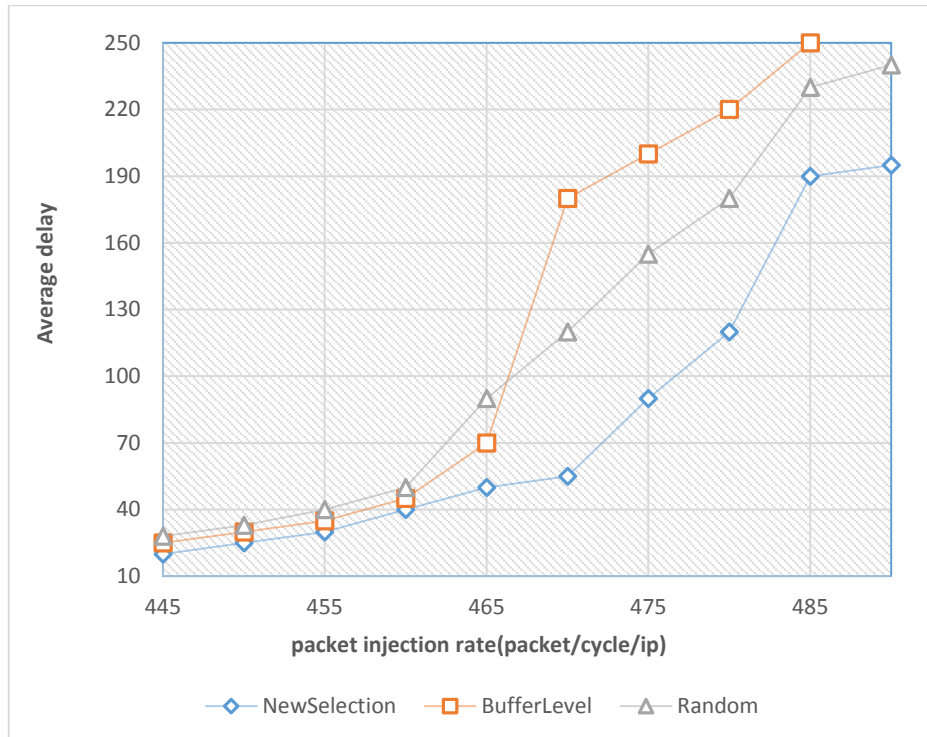


Figure 5. Average delay (m.s)

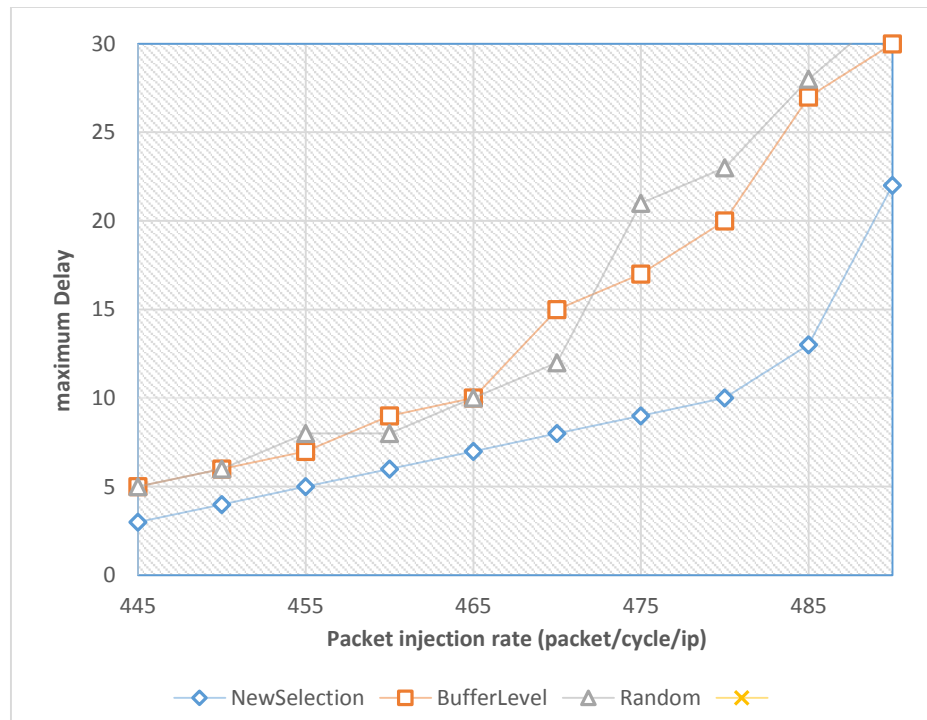


Figure 6. Maximum delay (m.s)

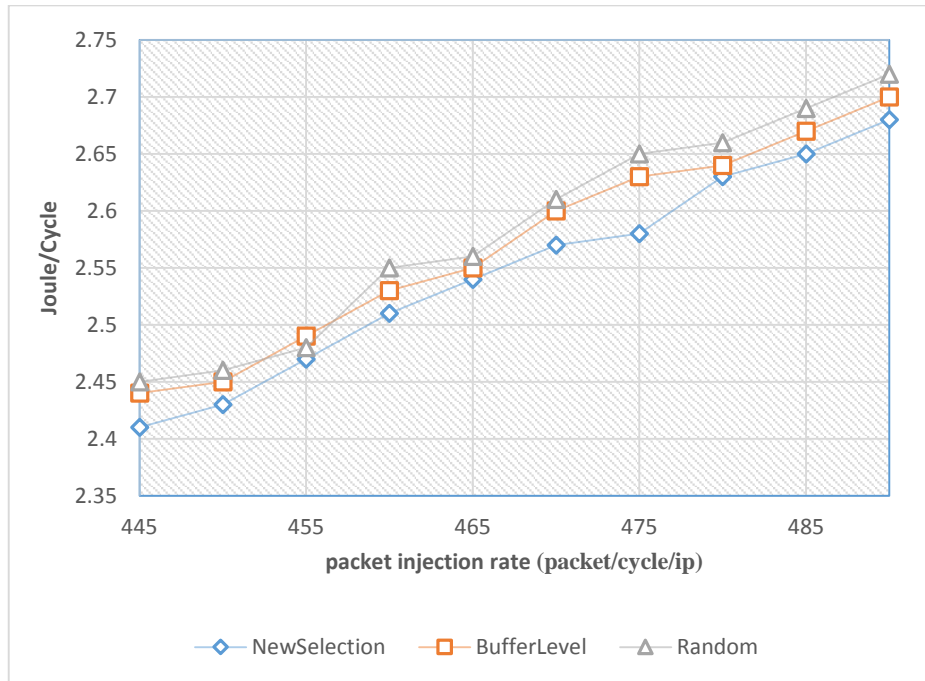


Figure 7. Power consumption (joule/cycle)

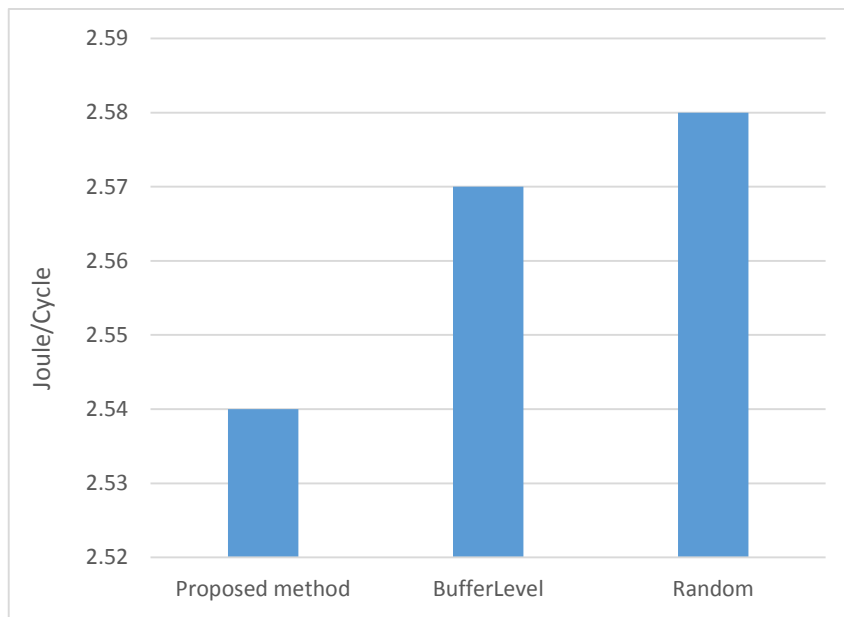


Figure 8. Energy consumption (joule/cycle)

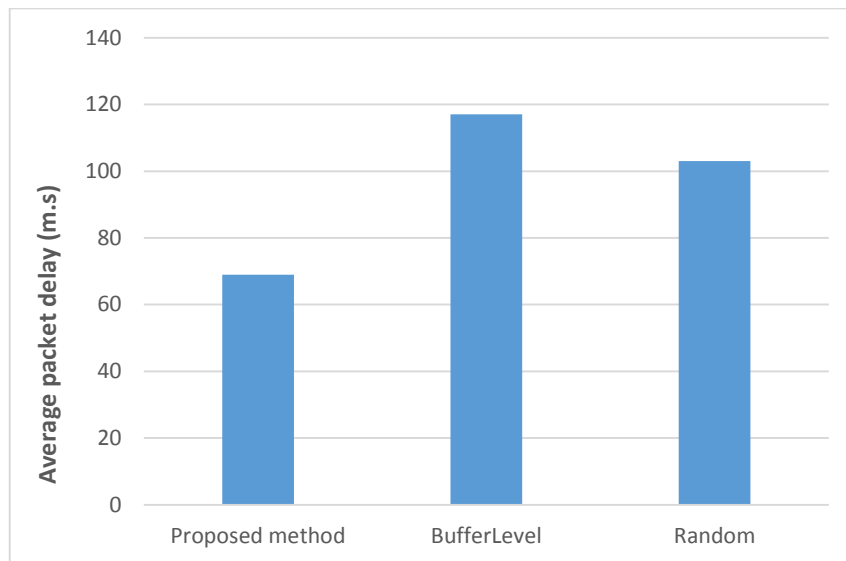


Figure 9. Average packet delay (m.s)

5. Conclusion

There are usually several different paths to reach from one node to another in NoC; accordingly the selection functions are used along with the routing algorithms. The effect of any routing algorithm depends on the selection strategy. This study, improved the efficiency of on-chip networks by offering a new selection strategy. As mentioned above, one of the most important applications of networks on-chips is improved efficiency without routing interruption [13]. The effect of any routing algorithm is dependent upon selection strategy. When the routing function, returns a set of output channels, selection function is used in order to select the output channel to send out packets.

In a simulation environment, it is shown that this approach increase the performance significantly. It is shown also that this method reduces the maximum and average delay in comparison to Random and Buffer Level strategies. On the other hand, given that a portion of calculation is done offline, energy consumption is also reduced significantly.

References

- [1] Jafarzadeh, N., Palesi, M., Khademzadeh, A., & Afzali-Kusha, A. (2014). Data encoding techniques for reducing energy consumption in network-on-chip. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(3), 675-685.
- [2] Zhou, X., Liu, L., Zhu, Z., & Zhou, D. (2015). A routing aggregation for load balancing network-on-chip. *Journal of Circuits, Systems and Computers*, 24(09), 1550137.
- [3] Chang, E. J., Hsin, H. K., Chao, C. H., Lin, S. Y., & Wu, A. Y. A. (2015). Regional ACO-based cascaded adaptive routing for traffic balancing in mesh-based network-on-chip systems. *IEEE Transactions on Computers*, 64(3), 868-875..
- [4] Dimitrakopoulos, G., Psarras, A., & Seitanidis, I. (2015). *Microarchitecture of Network-on-chip Routers*. Springer.

- [5] Asghari, A., Zoraghchian, A. A., & Trik, M. (2014). Presentation of an Algorithm Configuration for Network-on-Chip Architecture with Reconfiguration Ability. *International Journal of Electronics Communication and Computer Engineering (IJECCCE)*, 5(5).
- [6] Das, T. S., Ghosal, P., & Chatterjee, N. (2020). VCS: A method of in-order packet delivery for adaptive NoC routing. *Nano Communication Networks*, 100333.
- [7] Liu, L., Ma, R., & Zhu, Z. (2019). An encapsulated packet-selection routing for network on chip. *Microelectronics Journal*, 84, 96-105.
- [8] Ma, S., Enright Jerger, N., & Wang, Z. (2011, June). DBAR: an efficient routing algorithm to support multiple concurrent applications in networks-on-chip. In *Proceedings of the 38th annual international symposium on Computer architecture* (pp. 413-424).
- [9] Ebrahimi, M., Daneshtalab, M., Liljeberg, P., Plosila, J., & Tenhunen, H. (2012, February). LEAR-- A low-weight and highly adaptive routing method for distributing congestions in on-chip networks. In *2012 20th Euromicro International Conference on Parallel, Distributed and Network-based Processing* (pp. 520-524). IEEE.
- [10] Gratz, P., Grot, B., & Keckler, S. W. (2008, February). Regional congestion awareness for load balance in networks-on-chip. In *2008 IEEE 14th International Symposium on High Performance Computer Architecture* (pp. 203-214). IEEE.
- [11] Mak, T., Cheung, P. Y., Lam, K. P., & Luk, W. (2010). Adaptive routing in network-on-chips using a dynamic-programming network. *IEEE Transactions on industrial electronics*, 58(8), 3701-3716.
- [12] Bahman, F., Reza, A., Reshadi, M., & Vazifedan, S. (2020). CACBR: Congestion Aware Cluster Buffer base routing algorithm with minimal cost on NOC . *CCF Transactions on High Performance Computing*, 1-10.
- [13] Akbar, R., & Safaei, F. (2021). A novel heterogeneous congestion criterion for mesh-based networks-on-chip. *Microprocessors and Microsystems*, 84, 104056.
- [14] Ascia, G., Catania, V., Palesi, M., & Patti, D. (2006, February). A new selection policy for adaptive routing in network on chip. In *proceedings of the International Conference on Electronics, Hardware, Wireless and Optical Communications*.
- [15] Ascia, G., Catania, V., Palesi, M., & Patti, D. (2008). Implementation and analysis of a new selection strategy for adaptive routing in networks-on-chip. *IEEE Transactions on Computers*, 57(6), 809-820.
- [16] Ramakrishna, M., Kodati, V. K., Gratz, P. V., & Sprintson, A. (2016). GCA: Global congestion awareness for load balance in networks-on-chip. *IEEE Transactions on Parallel and Distributed Systems*, 27(7), 2022-2035.
- [17] Manevich, R., Cidon, I., Kolodny, A., & Wimer, S. (2011, August). A cost effective centralized adaptive routing for networks-on-chip. In *2011 14th Euromicro Conference on Digital System Design* (pp. 39-46). IEEE.
- [18] Arora, A., & Shukla, N. K. (2020). A Congestion Controlled and Load Balanced Selection Strategy for Networks on Chip. *International Journal of Distributed Systems and Technologies (IJ DST)*, 11(1), 1-14.
- [19] Samman, F. A., Hollstein, T., & Glesner, M. (2012). Runtime contention and bandwidth-aware adaptive routing selection strategies for networks-on-chip. *IEEE Transactions on Parallel and Distributed Systems*, 24(7), 1411-1421.
- [20] Touati, H. C., & Boutekkouk, F. (2020). Reliable Weighted Globally Congestion Aware Routing for Network on Chip. *International Journal of Embedded and Real-Time Communication Systems (IJERTCS)*, 11(3), 48-66.
- [21] Werner, S., Navaridas, J., & Luján, M. (2017). A Survey on Network-on-Chip Architectures. *ACM Computing Surveys (CSUR)*, 50(6), 89.
- [22] Rezaei, A., Daneshtalab, M., Safaei, F., & Zhao, D. (2016). Hierarchical approach for hybrid wireless network-on-chip in many-core era. *Computers & Electrical Engineering*, 51, 225-234.
- [23] Xie, R., Cai, J., & Xin, X. (2016). Simple fault-tolerant method to balance load in network-on-chip. *Electronics letters*, 52(10), 814-816.
- [24] Xie, R., Cai, J., Xin, X., & Yang, B. (2017). MCAR: Non-local adaptive Network-on-Chip routing with message propagation of congestion information. *Microprocessors and Microsystems*, 49, 117-126.
- [25] Tang, M., Lin, X., & Palesi, M. (2016). Local Congestion Avoidance in Network-on-Chip. *IEEE transactions on parallel and distributed systems*, 27(7), 2062-2073.
- [26] Wang, L., Wang, X., & Mak, T. (2016). Adaptive routing algorithms for lifetime reliability optimization in Network-on-Chip. *IEEE Transactions on Computers*, 65(9), 2896-2902.

- [27] Ahmed, A. B., & Abdallah, A. B. (2014). Graceful deadlock-free fault-tolerant routing algorithm for 3D Network-on-Chip architectures. *Journal of Parallel and Distributed Computing*, 74(4), 2229-2240.
- [28] Babu, Y. A., Prasad, G. M. V., & Solomon, J. B. (2018). FPGA Implementation of Buffer-Less NoC Router for SDM-Based Network-on-Chip. In *Progress in Advanced Computing and Intelligent Engineering* (pp. 561-567). Springer, Singapore
- [29] Chen, Y. Y., Chang, E. J., Hsin, H. K., Chen, K. C. J., & Wu, A. Y. A. (2017). Path-Diversity-Aware Fault-Tolerant Routing Algorithm for Network-on-Chip Systems. *IEEE Transactions on Parallel and Distributed Systems*, 28(3), 838-849.
- [30] Catania, V., Mineo, A., Monteleone, S., Palesi, M., & Patti, D. (2015, July). Noxim: An open, extensible and cycle-accurate network on chip simulator. In *Application-specific Systems, Architectures and Processors (ASAP), 2015 IEEE 26th International Conference on* (pp. 162-163). IEEE.
- [31] Zerbo, B., Sevaux, M., Rossi, A., & Créput, J. C. (2017). Optimizing the Cyclic K-conflict-free Shortest Path Problem in a Network-on-chip. *International Journal of Computer & Software Engineering*, 2(1), 115-1.
- [32] Behrouzian Nejad, M. " Parametric Evaluation of Routing Algorithms in Network on Chip Architecture." *Comput Syst Sci & Eng* (2020) 5: 367–375
- [33] Zong, W., Agyemen, M. O., Wang, X., & Maky, T. (2015, September). Unbiased regional congestion aware selection function for nocs. In *Proceedings of the 9th International Symposium on Networks-on-Chip* (pp. 1-8).
- [34] Chang, E. J., Hsin, H. K., Lin, S. Y., & Wu, A. Y. (2013). Path-congestion-aware adaptive routing with a contention prediction scheme for network-on-chip systems. *IEEE Transactions on computer-aided design of Integrated circuits and systems*, 33(1), 113-126.
- [35] Mehranzadeh, A., Khademzadeh, A., Bagherzadeh, N., & Reshadi, M. (2019). DICA: destination intensity and congestion-aware output selection strategy for network-on-chip systems. *IET Computers & Digital Techniques*, 13(4), 335-347
- [36] Gabis, A. B., Bomel, P., & Sevaux, M. (2018). Bi-objective cost function for adaptive routing in network-on-chip. *IEEE Transactions on Multi-Scale Computing Systems*, 4(2), 177-187.
- [37] Mohammadi Ghanatghestani, M., & Mohammadi Ghanatghestani, F. (2021). A Full Adder Cell Based on MOSFET Technology to apply in Arithmetic circuits. *Journal of Advances in Computer Research*, 12(2), 1-15.
- [38] Mokhlesi Ghanevati, D., Khorami, E., Boukani, B., & Trik, M. (2020). Improve Replica Placement in Content Distribution Networks with Hybrid Technique. *Journal of Advances in Computer Research*, 11(1), 87-99.
- [39] Mohammadi Ghanatghestani, M., & Bagherizadeh, M. (2019). A Low Power Full Adder Cell based on Carbon Nanotube FET for Arithmetic Units. *Journal of Advances in Computer Research*, 10(3), 1-12.

