

Parametric Evaluation of Routing Algorithms in Network on Chip Architecture

Mohammad Behrouzian Nejad*

Department of Computer Engineering, Shadegan Branch, Islamic Azad University, Shadegan, Iran

Considering that routing algorithms for the Network on Chip (NoC) architecture is one of the key issues that determine its ultimate performance, several things have to be considered for developing new routing algorithms. This includes examining the strengths, capabilities, and weaknesses of the commonly proposed algorithms as a starting point for developing new ones.

Because most of the algorithms presented are based on the well-known algorithms that are studied and evaluated in this research. Finally, according to the results produced under different conditions, better decisions can be made when using the aforementioned algorithms as well as when presenting new routing algorithms. In this research, we first describe the existing algorithms include: XY, YX, Odd- Even and DyAD. We then evaluate each of the routing algorithms which naturally have their own strengths and weaknesses under different conditions. In the first scenario, based on the criteria of average latency, average throughput and average energy consumption in determining the final performance of the network on the chip, we show the algorithms in terms of their performance by deterministic and adaptive routing algorithms. In the second scenario, we evaluate the algorithms based on the network size and the number of cores on the chip. As a result, these algorithms can make better decisions when using these algorithms as well as when presenting new routing algorithms, considering the results produced under different condition.

Keywords: Network on Chip, Routing Algorithm, Latency, Throughput, Energy Consumption

1. INTRODUCTION

The idea of NoC is a sequence of Systems on Chip (SoC) in which the cores are connected to each other on the basis of a communication infrastructure including switches or routers through communication links called the interconnection network. Compared to traditional chip-based communications in the on-chip system, the on-chip networking solution can increase scalability, reliability, and bandwidth availability [1–7]. One of the major challenges in chip design is the routing problem and how to map the cores in a network interface. In the problem of mapping, latency and bandwidth constraints and communication, the goal is to optimize and save energy. This is complicated because there are many solutions and, in most cases, not all the research to find the optimal solutions, where many heuristic algorithms are provided to find the

optimization solutions. In order to increase the performance of the system, allocation algorithms are proposed to minimize communication overhead and power consumption. For example, the Fixed Center algorithm in [8], the best neighborhood search algorithm in [9], and the simulated fusion algorithm in [10]. In order to minimize the data transfer delay between processing elements, NOC architecture with central shared memory has been proposed to extend data transfer directly from / to shared memory [11]. NoC has been suggested as a good solution for achieving better performance and higher system performance over today's complex chips. Compared to traditional chip-based communications in the SoC, the NoC solution can increase scalability, reliability and available bandwidth. In this scheme, the intellectual property cores in the chip are connected by a network structure and the communication between these cores is done by sending a message. Because this network is located on a chip and the resource level on the chip is very limited and communication in this network has to be done with very little

*E-mail: Behrouzian@iaushadegan.ac.ir

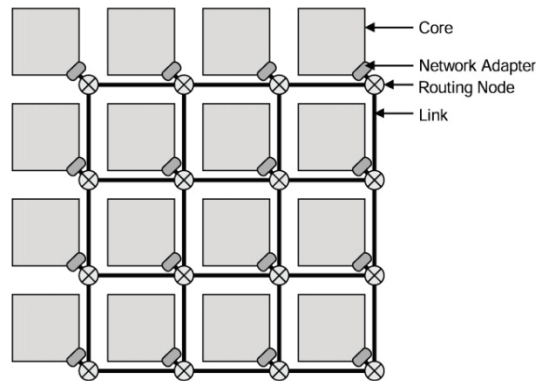


Figure 1 Two-dimensional mesh-based NoC [15].

delay, so in designing communication protocols for the NoC due to the existing constraints and constraints, Conventional communication protocols for computer networks cannot be used simply [1–6]. How the nodes connect to each other in a NoC determines its topology. Network connectivity can be regular or irregular. Much architecture has been proposed for the NoC [12–14]. Fig.1 shows a mesh based NOC, which consists of a grid of 16 cores. Each core is connected to a switch by a network interface. Cores communicate with each other by sending packets via a path consisting of a series of switches and inter-switch links [15]. The NOC contains the following fundamental components. a) Network adapters implement the interface by which cores (IP blocks) connect to the NOC. Their function is to decouple computation (the cores) from communication (the network). b) Routing nodes route the data according to chosen protocols. They implement the routing strategy. c) Links connect the nodes, providing the raw bandwidth. They may consist of one or more logical or physical channels.

In recent years, various algorithms have been proposed for the chip network. Among them are Algorithms DINRA [16], Dn-FTR [17], E-XY [18], PDA-FTR [19] and HFTRA [20].

The important thing is that many of these algorithms have been based on basic algorithms and developed from them. In this research, we first describe the well-known algorithms in the NoC. We evaluate each of the routing algorithms under different conditions. In the first scenario, we evaluate the algorithms in terms of deterministic and adaptive routing algorithms based on three important criteria: average delay, average throughput and average energy consumption. In the second scenario, we evaluate the algorithms based on the network size and the number of cores on the chip. As a result, these algorithms can make better decisions when using these algorithms as well as when presenting new routing algorithms, considering the results produced under different conditions. The contents of this paper are organized as follows: in the section 2 routing problem definition and routing algorithms are presented. In the section 3 Evaluation and its details are defined. After these the results obtained from the experiments are described and discussed; finally, the summarized conclusion is given.

2. ROUTING ALGORITHM

Routing algorithms generally can be divided into two deterministic and adaptive groups. Deterministic algorithms always set a

particular path between a pair of source and destination nodes, that is, all paths are initially sent from the source node to the specified node and then packets are sent to the destination node in the same specified way [21]. In these algorithms, no attention is paid to network traffic conditions when determining the route for packets.

In contrast to deterministic routing algorithms, there are adaptive routing algorithms. In adaptive algorithms, routing decisions are made according to the network traffic conditions, and these algorithms usually try to redirect packets to other destinations with less congestion if there is congestion in the network [22,23]. Depending on network traffic and the packet is sent in the same direction.

2.1 XY Deterministic Routing Algorithm

As stated in the previous section, in a deterministic routing, the path of packets between source and destination is definite and does not change. This method determines the routing path before sending packets to the network. Many networks are compatible with this routing because it is easy and inexpensive to implement. The switches are easy to implement in this routing. An example of definitive routing is XY routing. The XY routing strategy can be applied to a regular two-dimensional MESH correlation. The position of MESH nodes and network components will be explained by their coordinates. The X coordinates are for the horizontal direction and the Y coordinates are for the vertical direction. In selecting the XY output, packets first go to the X dimension and are then routed to the Y dimension. In other words, in this routing algorithm, packets first travel the X path until the $X_{Source} = X_{Destination}$ condition is established, then move in the Y direction until the $Y_{Source} = Y_{destination}$ condition is established. Figure 2 shows the some paths propagated by the XY routing algorithm [22,24–26].

2.2 Odd-Even Routing Algorithm

The Odd-Even (OE) algorithm is another adaptive algorithm based on the rotational model [27]. This algorithm is more versatile and has little overhead compared to other adaptive routing algorithms that do not use the virtual channel. Like other proposed algorithms, this algorithm imposes a series of

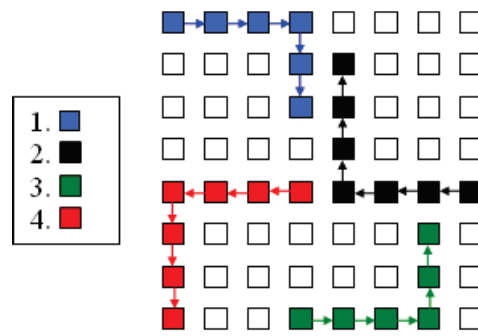


Figure 2 Some paths propagated by the XY routing algorithm.

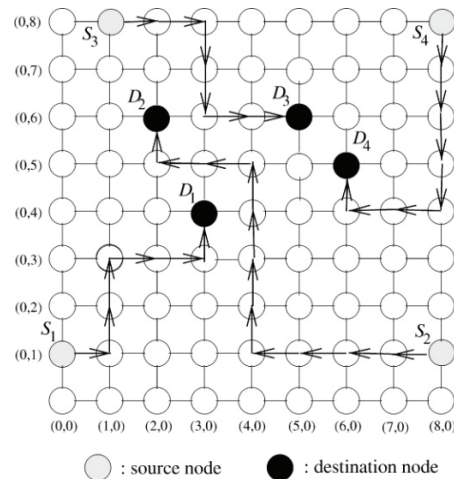


Figure 3 shows some possible routing paths.

constraints on rotation to avoid impingement. In the OE, the impedance wheels are expected to be restricted to places where some rotations can occur, thereby avoiding deadlock. Therefore, no rotations are eliminated in this model. This makes the degree of adaptability of this algorithm much higher than the previous algorithms. It also creates new routing algorithms in combination with previous algorithms and techniques, which we will see below as an example. To give this algorithm some expressions are necessary: In a two-dimensional mesh with dimensions $K_0 \times K_1$, each node X is known by its coordinates (X_0, X_1) , where X_0 is the zero coordinate and X_1 is the first coordinate. There are four directions to the north, south, east and west on the 2D Mesh. All nodes that have the same dimension as zero are in a column and all nodes that have the same dimension are in a row. In this model it is called an even column if its zero dimensions are even. Similarly, an individual column is called odd if its zero dimensions are odd. Fig. 3 shows some possible routing paths for four packets in a mesh topology. There are generally two main rules in OE:

Rule One: No packets are allowed to rotate from east to north in any node located in an even column. Also, in any node located in an odd column no packet will be allowed to rotate north to west. Rule Two: No packet is allowed to rotate from east to south in any node located in an even column. Also, in any node located in an odd column, no packet is allowed to rotate south to west. It is proven that any routing algorithm that uses OE rules will be deadlock free.

For example Fig. 3 show some possible routing paths for packets in a mesh NoC. S_i represent the source and D_i represent

the destination nodes of packet p_i . At node $(2, 3)$ p_1 can only move east as an EN turn is not allowed at the column. Consider p_2 , which is a westbound packet. It cannot turn north at node $(7, 1)$ or node $(5, 1)$ since it is prohibited from taking an NW turn, which is required later for it to reach the destination, in odd columns.

2.3 DyAD Routing Algorithm

The idea of the DyAD routing algorithm is to work in a deterministic mode when the traffic is low to minimize delay and whenever the congestion of the network goes above a certain limit; the routing should be in adaptive mode [28]. According to the latest reports, OE adaptive algorithm performs better than other adaptive algorithms presented. Another reason for using this algorithm is that all algorithms developed based on OE are deadlock free and have a much higher degree of compatibility than other algorithms. For example, if in OE adaptive mode a packet with the specified source and destination address can be routed to both p_1 and p_2 routes, in DyAD mode this packet will always be routed only to P_1 . For this reason, this method is used in a deterministic, which is primarily OE-based and deadlock free. Secondly, because of its compatibility with the OE to implement its router, there is no need to use any additional equipment, since this algorithm is based on the same OE rules. And all the equipment needed to support it has already been used for OE, thus increasing network performance both during low congestion and high congestion.

3. EVALUATION

The important thing is that many of algorithms have been developed based on basic algorithms which evaluated in this paper. Basic methods include Algorithms XY, YX and Odd-Even. Many of the new algorithms introduced use the logic of these algorithms such as: DyAD, West First, Negative Last, Surrounding XY, FTXY, SNWE, E-XY and etc. The results produced under different conditions can make better decisions by researcher when presenting ideas for developing new routing algorithms. Three very important issues in the NoC are the amount of latency, throughput and power consumption, all three criteria are considered for evaluating algorithms. These three are key criteria for evaluation.

There are two issues that can be raised in NoC: output selection and input selection. Due to the fact that input selection and output selection are separate from each other, in this paper we have focused on output selection in evaluation. The reason for this is that most studies try to reduce latency, throughput and reduce the power consumption of network, and this is possible by providing appropriate methods for output selection. In this paper we use from FCFS input selection method.

To evaluate the performance of the proposed algorithm we have developed a C++ based simulator. Noxim simulation software is used to implement and simulate the algorithms. Unlike other network simulators, this simulator was developed specifically for NoC simulation. It is an open source simulator and also has the ability to define new routing algorithms. It is based on the C++ language and runs under the Linux operating system. In the Noxim simulator, two delay and throughput parameters are considered as performance parameters. Throughput can be defined in different forms based on the implementation features, the throughput defined in the Noxim structure is as relation 1.

$$T = \frac{\text{Tptal received flits}}{\text{Number of nodes} * \text{Total cycles}} \quad (1)$$

In relation 1, Total Received Flits refers to the number of all flits that reach their destination node, Number of Nodes, the number of network generating nodes, and Total Cycles to the number of clock cycles that elapse between the first message generation event and the last message received. Therefore, the message's throughput is determined as a fraction of the maximum load that the network is able to physically support. The next performance parameter, as we said, is the delay parameter, which is defined here as the average delay form based on relation 2.

$$D = \frac{1}{K} \sum_{i=1}^K D_i \quad (2)$$

In relation 2, K denotes the total number of messages that will reach their destination nodes, and D_i denotes the delay message i . In fact, the delay parameter will be the time interval (in clock cycle) that elapses between the two events of the header flit being fed into the network at the source node and receiving the sequence flit at the destination node. The power consumed on the chip network consists of two parts: the power consumed in the routers and the power consumed by the links as relation 3.

$$P_{NoC} = P_{routers} + P_{links} \quad (3)$$

As $P_{routers}$ and P_{links} are dependent on the total capacity and activity of the switch signal and each part of the connection, respectively.

3.1 Deterministic Routing Algorithms Evaluation

In this simulation, XY and YX deterministic routing algorithms are implemented and evaluated separately. The performance of the mentioned algorithms has been evaluated by the graphs of average delay, throughput and power consumption. In this simulation a 2D mesh NOC with 5 * 5 cores is used and the above algorithms use wormhole switching technique. Each source generates packets and injects them into the network at specific intervals selected by an exponential distribution. The length of each packet is considered to be 5 flits, each flite is 32 bits and the width of each channel is 32 bits. The size of each input buffer is also considered to be 5 flits. For each specific packet injection rate (the number of packets injected into the network per cycle) the average packet delay was measured. Each cycle is one millisecond. Packet delay is considered from the time of its first flit creation to the source node until it receives its last flit at destination. Elementary cycle information has not been applied to computing until the network reaches an almost stable state. All of these adjustments and choices have been made to be as consistent as possible with previous work to have a good benchmark.

Figure 4 shows the average delay of packets. In this figure, the XY and YX algorithms are compared. The horizontal axis is the rate of packet injection rate into the network. The vertical axis indicates the rate of delay per cycle.

As can be seen in Figure 4, the YX algorithm performs better than the XY routing algorithms when the packet injection rate is low. The XY algorithm performs better than the YX algorithm when the packet injection rate is increased and the traffic is higher. In general, the figure shows the superiority of the XY method in reducing latency over the YX algorithm.

Figure 5 shows the data packet throughput of the XY and YX algorithms at different packet injection rates. The horizontal axis indicates the pack injection rate. The vertical axis indicates the flow rate in terms of flits/cycle/ip.

As can be seen in Figure 5, due to the definite nature of the XY and YX routing algorithms, they operate almost similar in different traffic conditions.

Since the amount of energy consumed is one of the most important parameters in determining the effectiveness of an algorithm, Figure 6 shows the energy consumption of the XY and YX deterministic algorithms at different pack injection rates. The horizontal axis indicates the packet injection rate to the network. The vertical axis represents the amount of energy consumed by the algorithms.

As can be seen in Figure 6, the pack injection rate increases, the energy consumption increases. From the XY and YX algorithms it can be seen that the YX algorithm has lower power consumption than the XY algorithm. As shown in the figure, except for the lowest injection rate depending on the network, the YX algorithm had lower power consumption than the XY algorithm, and as a whole shows the superiority of the YX method over the XY in total power consumption.

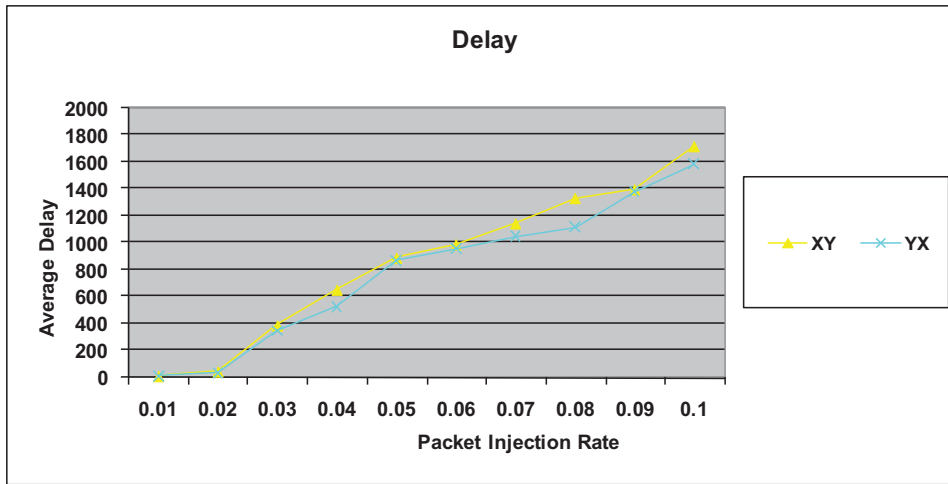


Figure 4 Average delay of receiving packets at different packet injection rate.

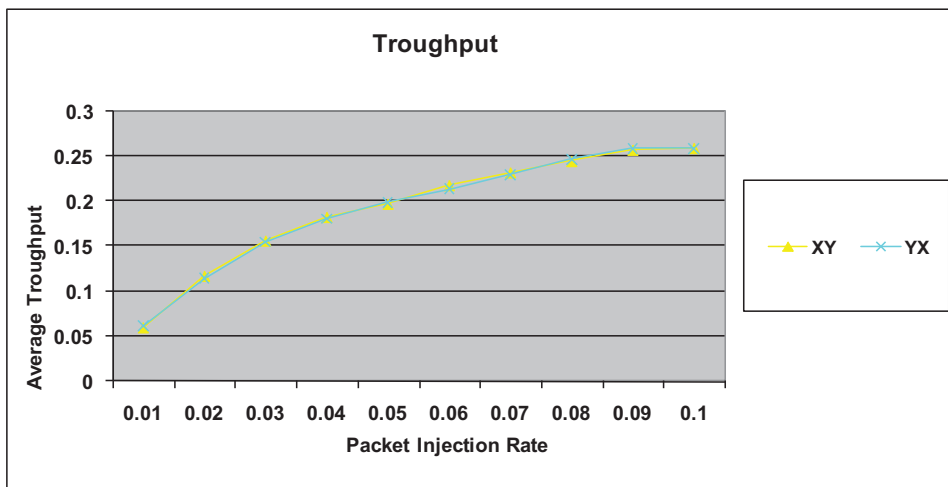


Figure 5 Average throughput of algorithms at different packet injection rate.

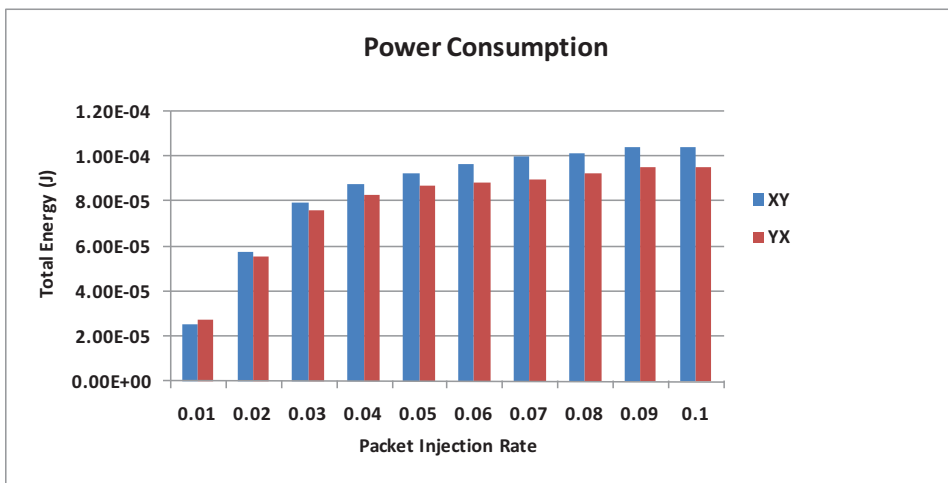


Figure 6 Average energy consumption of algorithms at different packet injection rate.

3.2 Adaptive Routing Algorithms Evaluation

Figure 7 shows the average delay of packets. In this form, the adaptive routing algorithms of odd even and DyAD are compared. The horizontal axis is the rate of packet injection

into the network. The vertical axis indicates the rate of delay per cycle.

As can be seen in Fig. 7, the OE routing algorithms and DyAD operate almost similarly in conditions where packet injection rate is low on the network and by increasing the packet injection

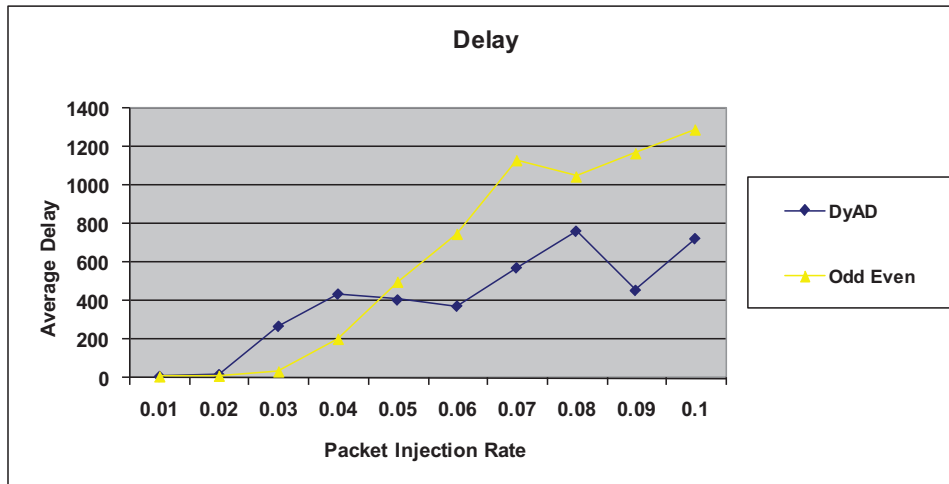


Figure 7 Average delay of odd even and DyAD Algorithms.

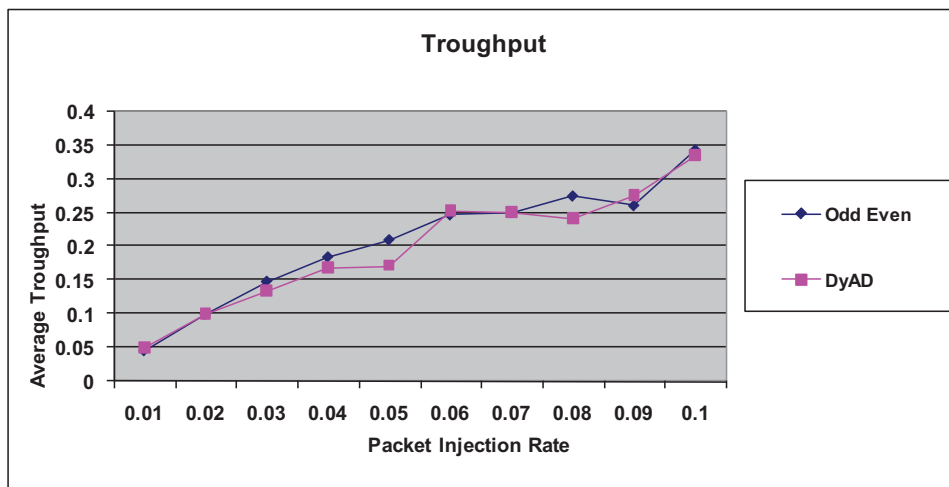


Figure 8 Average throughput of odd even and DyAD Algorithms.

rate on the network, the OE routing algorithm performs better than the DyAD algorithm. By increasing the packet injection rate on the network, DyAD algorithm performs better than OE on average. In general, the figure shows the superiority of the DyAD method in reducing latency over the OE algorithm.

Figure 8 shows the packet throughput of OE and DyAD algorithms at different packet injection rate. The horizontal axis indicates the packet injection rate to the network. The vertical axis indicates the flow rate in terms of flits/cycle/ip.

As can be seen in Fig. 8, the OE and DyAD algorithms perform almost the same when packet injection rate is low, and the OE algorithm has better throughput when packet injection rate is increased on the network. In general, the OE and DyAD algorithms are similar in some conditions, but on average the OE algorithm performs better than the DyAD algorithm.

Figure 9 shows the energy consumption of the OE and DyAD at different pack injection rates. The horizontal axis indicates the packet injection rate to the network. The vertical axis represents the amount of energy consumed by the algorithms.

As can be seen in Fig. 9, the energy consumption of OE and DyAD algorithms operate at different rates at low relative load conditions and with increasing injection rates naturally have high energy consumption. It goes, but as the packet injection

rate increases on the network, it can be seen that the DyAD algorithm has lower power consumption than the OE algorithm. As the figure shows, the OE method performs poorly in the power consumption.

3.3 Evaluation of XY, OE and DyAD Algorithms

Considering the proper performance of XY, Odd even and DyAD algorithms, in this section we evaluate these algorithms in different conditions of network size and number of cores in the network. The performance of the mentioned algorithms has been evaluated by means of average delay, throughput and power consumption. In this simulation, 2D mesh NOC with 5 * 5 cores is used, and the above algorithms use the wormhole switching technique. Each source generates packets and injects them into the network at specific intervals selected by an exponential distribution. The length of each packet is considered to be 5 flits, each flite is 32 bits and the width of each channel is 32 bits. The size of each input buffer is also considered to be 5 flits. Packet injection rate into network (the number of packets injected into the network per cycle) is also a constant rate of 0.1. Each cycle

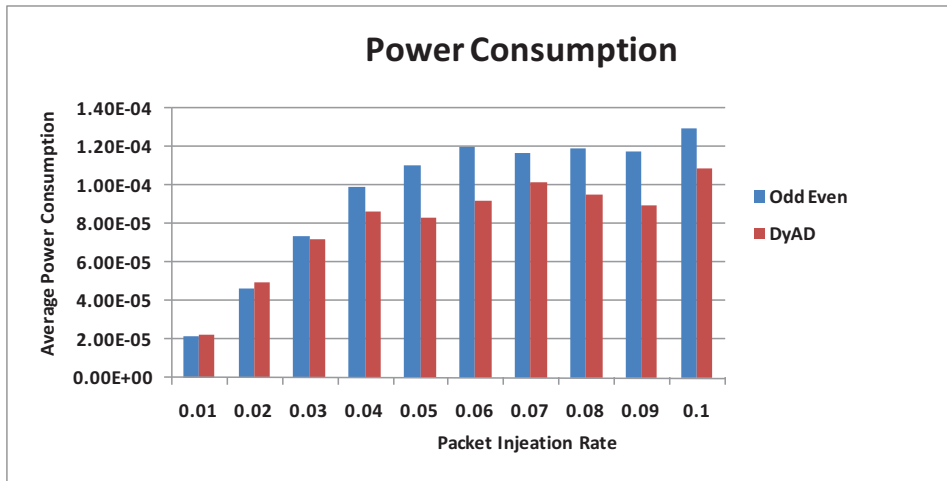


Figure 9 Average energy consumption of odd even and DyAD Algorithms.

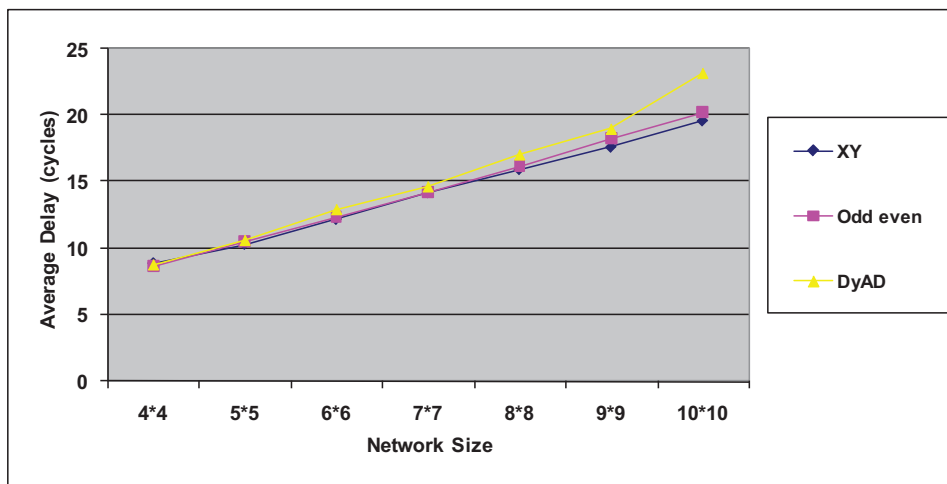


Figure 10 Average delay of packets in different network size.

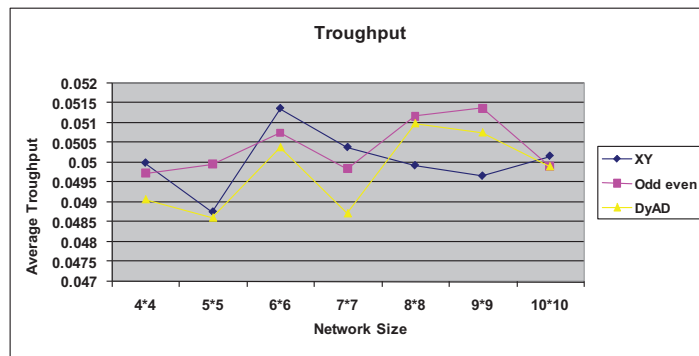


Figure 11 Average throughput of packets in different network size.

is one millisecond. Packet delay is considered from the time of its first flit creation to the source node until it receives its last flit at destination. Elementary cycle information has not been applied to computing until the network reaches an almost stable state. All of these adjustments and choices have been made to be as consistent as possible with previous work to have a good benchmark.

Figure 10 shows the average delay of packets. In this form, the XY, odd even and DyAD routing algorithms are compared.

The horizontal axis is the size of the network and the number of cores. The vertical axis indicates the rate of delay per cycle.

As can be seen in Fig. 10, the XY, odd even and DyAD routing algorithms operate at almost the same when the network size is low and by increasing the network size, the odd even and XY routing algorithms perform better than the DyAD algorithm. By increasing the network size, on average, the XY algorithm performs better than the odd even and DyAD algorithms.

Figure 11 shows the throughput of the XY, odd even and DyAD algorithms at different network sizes. The horizontal

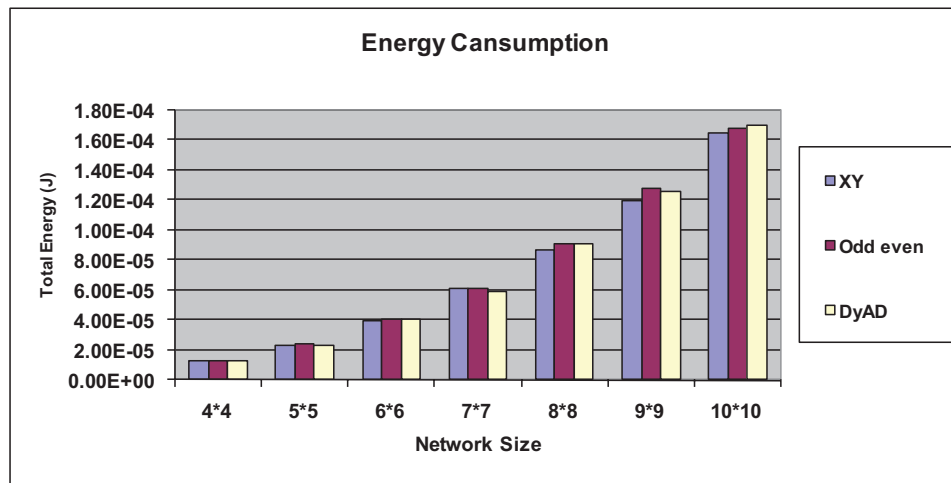


Figure 12 Average energy consumption of algorithms in different network size.

axis represents the network size. The vertical axis indicates the flow rate in terms of flits/ cycle/ ip.

As can be seen in Fig. 11, the DyAD algorithm performs poorly in the low network size than the XY and OE algorithms. As the network size increases, the XY algorithm performs different functions. As the size of the network increases, it can be seen that the odd even and DyAD algorithms have similar performance. As the figure shows, in general the XY method has a stronger performance.

Figure 12 shows the energy consumption of XY, odd even and DyAD algorithms at different network sizes. The horizontal axis represents the network size. The vertical axis represents the amount of energy consumed by the algorithms.

As can be seen in Fig. 12, the energy consumption of the XY, odd even and DyAD algorithms performs similarly at the lower network size. As the network size increases, the XY algorithm performs better.

4. CONCLUSIONS

As technology advances and the need for heavy and parallel computing increases, chip-based systems have become a viable solution for processing massive data by simultaneously assigning tasks through data partitioning and partitioning techniques. Compared to single-processor and single-core systems, multi-core systems have higher computational capabilities and can be used to increase communication efficiency and parallelization. One of the most important challenges in determining the overall performance of the NoC is selecting the appropriate routing algorithm. The routing algorithm determines the route of packet forwarding between the sender and the receiver. Since routing is one of the key issues that determine the ultimate network performance on the chip, since it is one of the most useful ways to evaluate the capabilities and weaknesses of the commonly proposed methods; we implemented and evaluated. Because most of the algorithms presented are based on the algorithms proposed in this research. In this study, we first describe the existing algorithms. We then evaluated the algorithms under different conditions. The evaluations are based on the criteria of average packet delay, average throughput and average power

consumption. We also present the results of the evaluation for better utilization in the form of graphs in terms of their performance.

It is suggested that researchers interested in NoC considering the results produced under different conditions can make better decisions when presenting ideas for developing new routing algorithms considering the performance of the algorithms in different conditions. There are two issues that can be raised in NoC: output selection and input selection. Input selection methods such as FCFS, RR and CAIS have reasonable performance but may not work well on large networks. But in the field of input selection, newer methods can be proposed to improve NoC performance so that it can be used in larger networks. It is possible to use hybrid methods with different priorities such as residual path length, delay rate for future research. For output selection, heuristic methods can be used to find the best path to information packets.

5. ACKNOWLEDGMENT

This study was supported by a grant number (117318) from Islamic Azad University, Shadegan Branch, Iran. The grant entitled: Performance parametric evaluation of routing algorithms in Network on Chip architecture.

REFERENCES

1. M.S. Nehnoui, "A New Fault Tolerant Routing Algorithm for Networks on Chip", *International Journal of Embedded and Real-Time Communication Systems*, Vol. 10, No. 3, pp 68–85, 2019.
2. D. Sinha, A. Roy, K. V. Kumar, P. Kulkarni, J. Soumya, "Dn-FTR: Fault-Tolerant Routing Algorithm for Mesh based Network-on-Chip", In *Proceedings of the 4th International Conference of Recent Advances in Information Technology, IEEE*, 2018.
3. A. Punhani, P. Kumar, "E-XY: an entropy based XY routing algorithm", *International Journal of Grid and Utility Computing*, Vol. 10, No. 2, pp. 179–186, 2019.
4. R. Rajaei, F. A. Kakhki, B. Golvarzade, "Evaluation of Routing Algorithms and Introducing a New Semi-Adaptive Algorithm for Network-On-Chip", In *Proceedings of the 13th Iranian*

- Student Conference of Electrical Engineering. Tarbiat Modarres University, Iran, 2010.
5. D. Wu, M. B. Al-Hashimi, T. M. Schmitz, "Improving Routing Efficiency for Network-on-Chip through Contention-Aware Input Selection", In Proceedings of the Asia and South Pacific Design Automation Conference, 2006.
 6. S. Kumar, A. Jantsch, J. P. Soininen, M. Forsell, M. Millberg, J. Oberg, "A network on chip architecture and design methodology", ISVLSI, pp. 117–24, 2002.
 7. O. Lehtoranta, E. Salminen, A. Kulmala, "A parallel MPEG-4 encoder for FPGA based multiprocessor SoC", In Proceedings of the 15th International Conference on Field Programmable Logic and Application, p. 380, 2005.
 8. C. Chou, U. Ogras and R. Marculescu, "Energy- and performance-aware incremental mapping for networks on chip with multiple voltage levels", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 27, No. 10, p. 1866–1879, 2008.
 9. E. Carvalho, F. Moraes, "Congestion-aware task mapping in heterogeneous MpSoCs", In Proceedings of the International Symposium on System-on-Chip, 2008.
 10. H. Orsila, T. Kangasm, E. Salminen, "Automated memory-aware application distribution for Multi-processor System-on-Chips", Journal of Systems Architecture, Vol. 53, No. 11, pp.795–815, 2007.
 11. Ser-Hoon Lee, Yeo-Chan Yoon, Sun-Young Hwang, "Communication-aware task assignment algorithm for MPSoC using shared memory", Journal of Systems Architecture, Vol. 56, No. 7, pp. 233–241, 2010.
 12. N. Wu, Y. Mu, F. Zhou, F. Ge, "GA-MMAS: an Energy-aware Mapping Algorithm for 2D Network-on-Chip", In Proceedings of the World Congress on Engineering and Computer Science, USA, 2011.
 13. V. Rantala, T. Lehtonen, J. Plosila, "Network On Chip Routing Algorithms", Turko center for computer science, Joukahaisenkatu 3–5 B, 20520 Turku, Finland, 2006.
 14. W. J. Dally, B. Towles, "Principles and Practices of Interconnection Networks". Morgan Kaufmann, 2004.
 15. M. Behrouzian Nejad, A. Mehranzadeh, M. Hoodgar, "Performance of Input and Output Selection Techniques on Routing Efficiency in Network-On-Chip", International Journal of Computer Science and Information Security, Vol. 9, No. 9, pp 125–130, 2011.
 16. C. Nehnouch, M. Senouci, "A New Fault Tolerant Routing Algorithm for Networks on Chip", International Journal of Embedded and Real-Time Communication Systems, Vol. 10, No. 3, pp. 68–85, 2019.
 17. D. Sinha, A. Roy, K. V. Kumar, P. Kulkarn, J. Soumya, "Dn-FTR: Fault-Tolerant Routing Algorithm for Mesh based Network-on-Chip", In Proceedings of the 4th International Conference of Recent Advances in Information Technology, IEEE, 2018.
 18. A. Punhani, P. Kumar, "E-XY: an entropy based XY routing algorithm", International Journal of Grid and Utility Computing, Vol. 10, No. 2, pp. 179–186, 2019.
 19. Y. Y. Chen , E. J. Chang, H. K. Hsin, K.-C. J. Chen, A.-Y. A. Wu, "Path-diversity-aware fault-tolerant routing algorithm for network-on-chip systems", IEEE Transactions on Parallel and Distributed Systems, Vol. 28, No. 3, pp. 838–849, 2019.
 20. R. Bishnoi, "Hybrid fault tolerant routing algorithm in NoC", Perspectives in Science, Vol. 8, pp. 586–588, 2016.
 21. A. V. Mello, L. C. Ost, F. G. Moraes, N. L. V. Calazans, "Evaluation of Routing Algorithms on Mesh Based NoCs", Technical Report Series, PUCRS, No. 40, 2004.
 22. M. A. J. Jamali, H. M. Ghader, A. Khademzadeh, M. H. N. Ghavifekr, "Introducing New Adaptive Routing Algorithms to Reduce Number of Route Steps on Chip Networks (NoC) with Mesh and Torus Topology", In Proceedings of the 3rd Information Technology and Knowledge, Ferdowsi University, Iran, 2007.
 23. J. Duato, S. Yalamanchili, L. Ni, "Interconnection Networks: an Engineering Approach", IEEE Comp. Society Press, 1997.
 24. W. J. Dally, B. Towles, "Principles and practices of Interconnection network", morgan Kaufmann, 2004.
 25. S. Kumar, A. Jantsch, J. -P. Soininen, M. Forsell, M. Millberg, J. Oberg, "A network on chip architecture and design methodology", ISVLSI, pp. 24–117, USA, 2002.
 26. L. Benini, G. De Micheli, "Networks on chips: A new SoC paradigm, Computer", vol. 35, pp. 70–78, 2002.
 27. G. M. Chiu, "The odd-even turn model for adaptive routing", IEEE Transactions on Parallel and Distributed Systems, vol. 11, pp.729–38, 2000.
 28. J. Hu, R. Marculescu, "DyAD - Smart routing for networks-on-chip", DAC, pp. 260–263, USA, 2004.

