Indonesian Journal of Electrical Engineering and Computer Science
Vol. 28, No. 2, November 2022, pp. 962~972
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v28.i2.pp962-972

Congestion aware and game based odd even adaptive routing in network on chip many-core architecture

Radha Doraisamy[1], Minal Moharir[2], Rajakumar Arul[3]
[1]Department of Computer Science and Engineering, Amrita School of Engineering, Bengaluru, Amrita Vishwa Vidyapeetham, India
[2]Department of Computer Science and Engineering, RV College of Engineering, Bengaluru, India
[3]Centre for Smart Grid Techonologies/School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology Chennai, Chennai, India

Article Info

Article history:

Received Apr 15, 2022
Revised Jul 14, 2022
Accepted Aug 12, 2022

Keywords:

Congestion management
Game theory
Network on chip
Odd-even routing
Stag hunt

ABSTRACT

The era of single processors had almost reached a saturation state, and the industry had moved to multi-core processors for the newer generation of many-core architecture. Interconnections between multiple cores with network on chip (NoC) surpass traditional bus architecture for its quality of service (QoS) and other additional services. Seamless communication among the cores is more significant for better performance and the proper utilization of the cores. The rise in the cores count in a semiconductor chip adds the complexity of the communication among cores. Cache misses request and packet transmission's traffic possibly will reduce the performance of the architecture. A theoretical game-based methodology is proposed to improvise the performance and communication by routing the request packets in the NoC of the many core architectures and the throughput is maximized with reduced latency by using the stag-hunt game (SHG) model. The proposed communication algorithm routes the packets in an adaptive way by detecting the congestion in routers. The SHG based odd-even routing algorithm is adaptive and can divert the packets towards less congested routers using the information gathered about congestion in the system, so that the overall performance of the system in terms of latency and throughput is improved.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Rajakumar Arul
Centre for Smart Grid Techonologies/School of Computer Science and Engineering (SCOPE)
Vellore Institute of Technology Chennai
Chennai, Tamil Nadu, India
Email: rajakumar.arul@vit.ac.in

1. INTRODUCTION

The advancements in the very large-scale integration (VLSI) technology have provisioned the system to embed multiple circuits over a single chip. The tiny single chip has complex circuitry which is managed and supported by the technology. The innovations over the years include internet of things (IoT), 5G technology, artificial intelligence, High resolution images and videos, block chain and so on. These innovations require a high computational power that in turn need powerful processor. But the current technology computation requirements need more processors which can perform tasks in a parallel manner. To ensure faster rate operations and communications, multiple cores and many cores are introduced in the market for parallel processing of chunks of data. The generation of complementary metal-oxide-semiconductor (CMOS) technology enables the creation of larger and more advanced systems on an integrated circuit (IC). Dennard scaling and factors mentioned have resulted in a paradigm shift towards the multi-core processors by the semiconductor business community. Instead of a single powerful processor, many simple cores are placed on a similar die and tasks are parallelized using these cores to improve the performance for real time systems.

Nowadays multi-core processors normally have four-eight general purpose cores and the recent styles such as Intel Xeon is with seventy-two cores, that are commercialized dual and quad core processors are the very minimum requirements of the current versions of mobile and tablet processors. In a multi-core architecture, as each core processes multiple applications in parallel which results in increase in the chip's overall processing capability [1]. Many cores architecture consists of many lightweight cores that use of the hardware resources effectively [2]. Smaller the feature size of the semiconductor for the current demand of the performance, leading towards increase in the number of cores in terms of hundreds and thousands in a chip for the future processors. But the design part is complex than expected. Apart from the complexity of arrangement in the large number of cores, the efficient communications between the cores are also challenging. Regular arrays of processors and cache banks in tiled chip multiprocessors (CMPs) and heterogeneous resources in system on-chip (SoC) [3]. Network on chip (NoC) is a reliable solution introduced by CMOS technology for meeting the communication demands that is incurred as a bottleneck in the many cores design. The increase in number of cores will increase the performance, but with a tradeoff of communication cost. Communication between the cores is to access the data from the cores than accessing it from memory which costs lesser. Many researchers are tuned towards intra-core communication for better utilization of cores.

Buses or crossbar switches may not be a suitable solution for the many-core architecture with a greater number of cores, bandwidth requirements and frequent communications. Shared bus technology is suitable for computer networks but not for the many-core architecture as the rate of packet transfer may go low with congestion in the network. And also, efficiency of shared bus may reduce if many cores use the shared bus. Thus, a reliable infrastructure like NoC is required for satisfying scalability, latency and bandwidth constraints. NoC architecture consists of interconnecting devices liked Routers or switches, cores and the networkinterface. The cores communicate with other cores using routers with the help of network interface. Depending upon the topology, each router is connected to one or more cores. The packet from each core is transmitted through one or more routers to reach the destination core. By controlling the data packet flowas in computer network [4], it can improve quality of service (QoS) in NoC. A sample layout of NoC based architecture of 9 cores is shown in the Figure 1 [5].

A tiled many cores architecture consists of a grid of cores/tiles, each core connected to a router has a private L1 cache, a shared L2 cache and a network interface that connects the tile to the NoC router. The routers are connected to each other and the memory controllers by a mesh network. So far, most NoCs have been designed for either common/average case application behavior or the worst-case application behavior. Common case application behavior is similar to application specific scenarios that apply to a certain set of application domains or types. Here the architecture is in such a way that it is supposed to work in all the applications in the same way. For the worst-case application behavior, NoC resources are over provisioned than needed, that is, maximizing bandwidth and minimizing latency even during very low traffic scenarios. But the application requirements and the traffic patterns for the chip multicores are not fixed, and it varies over the time and is dependent on the application and the environment. That is, applications may require less bandwidth or the entire bandwidth. And some other applications may need low latency and high bandwidth together, and so on. The average case design will not provide better performance for the applications as it requires more than the supported bandwidth or that benefit from the lower latency. NoC designs (mostly the over provisioned design) are power and energy inefficient for applications that don't require the quality-of-service parameters such as high bandwidth or low latency.

The SoC application specific integrated chip (ASIC) uses customized NoCs that are specifically optimized towards the traffic requirements of a single application at the design time. This fixed traffic patterns allows designers to analyze their task graphs to implement NoCs to match an applications' traffic requirements. Studies have shown that these on-chip network contribute between 10% to 30% of the total chip and also it can account for a large number of cycles. This is the reason for NoC as a substantial part of multicore/many coredesign and hence needs to be studied carefully. Various topologies are available for arranging the cores and are suitable for diverse scenarios. Routing algorithms to route the packets are available for various topologies for efficient routing [6], [7]. Some algorithms use fixed paths, and some may use paths depending on the congestion in the network. Few may end up with non-minimal path but will reach in a shorter time.

Game theory has perceived in lot of parallel and distributed systems in regulating the behavior of core nodes, intermediate nodes and routing techniques. The reduction in energy drain, node behavior, reduced latency is greatly improved. Thus, the approaching research directions utilize the state of art of game theory applications effectively for the network performance and to achieve high QoS [8].

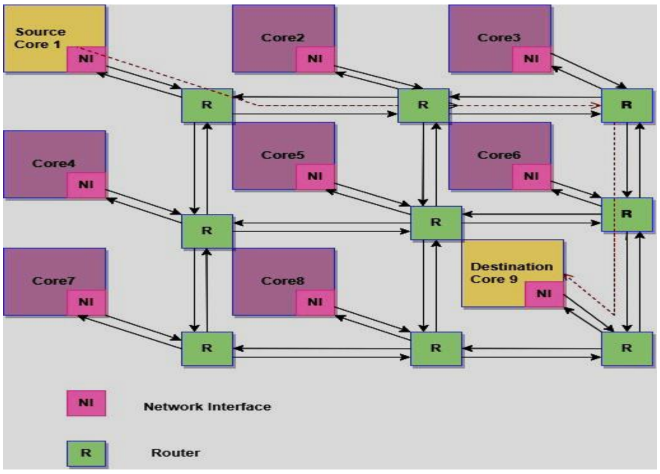

Figure 1. Layout of 9 cores architecture

The rest of the paper is organized as follows. Section 2 describes the literature survey of the work from which we incorporated elements to develop the algorithm. Section 3 outlines the system model and describes the components of the proposed model in detail. Section 4 details the simulation setup and results of the proposed work. Section 5 concludes the work and its future directions.

2. RELATED WORKS

Much advancement has been made in routing algorithms and routing techniques in the field of NoC architecture in-order to ensure a guaranteed delivery of packets. The routing algorithms could be classified as centralized versus distributed and static versus adaptive [9]. However, the increasing demands for high computing system performance has accelerated the demands for adaptiveness in routing, which can effectively provide an enhanced communication experience for the packets. The possible solutions such as reducing the packet size, increased use of large sized buffers, or using a smaller number of packets per message, rerouting the packets towards a less congested possible route, for managing the contention according to the traffic condition. In 2005, [10] proposed a router for on chip that interconnects and supports adaptively. They have proposed a router architecture that uses virtual channel with direction mapped. This architecture uses the congestion information of the neighboring routers. Here the destination channel is selected in such a way that it does not cause a deadlock and the path selection is made to route through less congested routers. Here, each router needs to maintain a record of the credits for the incoming and outgoing traffic. Once the allocator allocates a virtual channel for the flit, the outgoing virtual channel for the flit is decided initially by a hardware module that gives the possible virtual channels that can be used for avoiding possible deadlock scenarios, and in the next stage, the virtual channel which is having the lowest credits is used as the virtual channel to the destination. In 2010, Ramanujam and Lin [11] developed a new routing strategy where the adaptiveness lies in the packet destination. Here instead of collecting congestion information from a region of nodes or neighboring nodes, every node will have a record of delays to the other nodes. In other words, instead of calculating how populated the entire router is, the congestion is determined based on which all paths from the router are congested. This information is used to redirect the packet through the same router, in the less congested path, towards the destination. Since here the global congestion information is not required, it is able to meet the delay, area and power constraints for a NoC interconnection network. This gives a more precision in results compared to the regional congestion aware adaptive algorithms.

Raj *et al.* [12] proposed a study of hotspot formation where the cores experience more traffic in mesh NoCs. It de-routed the packets away from hotspot using deflection routing method. The aim is to de-route on the way to destination core to avoid more congestion. Hotspots are the non-uniform high traffic location of cores. A novel deflection routing technique is proposed to mitigate the effects of the destination hotspots. Ambient computing plays a vital role and acts as a framework to control several sets of services, resources to utilize it efficiently. In 2017, Debnath *et al.* [13] , authors suggested congestion management, using edge and in network throttling. Load balancing and network traffic throttling are the main purpose of the work. Ambient computing techniques uses the resources more effectively [14]. The injection rate is controlled to improve the throughput and latency with minimum logic and low cost for traffic throttling at the routers. Throttling signal generation hardware module is used inside every router for throttling purpose. Credit counters within each router are used to indicate the current buffer utilization. This buffer details are used for the traffic throttling signal decision. In 2010 Blagodurov *et al.* [15] proposed a method that predicts the contention in multicore system. It is done as follows. When a thread requests a cache line which is already full, the requested cache line replaces some of the existing shared cache lines. And these shared cache lines may belong to other threads which may cause resource contention. The contention is modelled by two different ways and tested. Multihop routing algorithm and uneven clustering algorithm are the few ways to minimize the hotpot problem [16]. Game [17]–[19] presents the stag hunt game that works effectively in the strategic situations where it just considers random amount of samples, the samples are individuals who are playing a series of stag hunt games tomark their selections in the strategic situation. Game theory acquire confident knowledge from the available data samples [20]. Considering the relevant studies and survey, the proposed work of this paper has the following contributions that includes congestion identification, congestion management using odd and even routing and the game theoretical model using stag hunt game (SHG) odd-even routing algorithm for maximizing the performance of the network in terms of QoS such as maximized throughput and reduced latency.

3. SYSTEM MODEL

While considering congestion or network traffic, the algorithms prefer shortest path and also the less congested model. Let us consider a network represented as Ns that has p edges E = $\{e_1, e_2, e_3 \ldots e_p\}$ and nodes denoted as N. The nodes are further classified as Core Nodes that has c nodes such as CN = $\{CN_1, CN_2 \ldots CN_c\}$, and the Network Interface Ni = $\{Nci_1, Nci_2\ Ncnc\}$ and the Router Nodes $Nrp_v = \{Nr_1, Nr_2, Nr_3 \ldots. Nrc\}$. The Router nodes will have physical as well as the virtual interfaces. Using the game strategy, the players would choose the path to traverse among the multicores. With the NoC, the user has to update the required parameters and also access the performance of the system. The main tradeoff here is that there should a maximum throughput with the reduced latency. The proposed approach utilizes two modules to pay off the utility, that also attains Nash equilibrium by maximizing the throughput or with a reduced latency.

3.1. Congestion identification module and congestion management

The implementation phase of the NoC has multiple cores in the architecture. To avoid the many-core traffic the following procedure is incorporated. The initial parameters that are pondered for network configuration are confirmed. With the chosen input parameters, the network is simulated. After the simulation the flow mechanism i.e., the path is investigated. The flits that are sent over the route helps to identify the whole delay being experienced at each router and its interface until the lifetime of flits. At each router the queuing and processing delays experienced by the flits are monitored. After each episode/iteration the router delay is updated with the current value. So, this helps to identify the exact delay that takes place at each router. To calculate the accurate statistical data, the number of times the delay value being updated are also maintained. Thus, the count and the delay metrics aid in finding the traffic congestion in a network and helps the flits to utilize the accurate route to maximize its performance by reducing the delays. Considering the parameters and the network configuration, it is vital to prove that the obtained utility in the proposed game is based on congestion identification and routing model are in nash equilibrium and with greater performance.

3.2. Game model for the routing

Considering the well-known Prisoner's Dilemma game, the game is played in such a way that there exists conflict with the mutual benefit. In the stag hunt game, the game is played to increase the payoff of one player depending on the belief of the other player’s choice. Figure 2 shows the cooperative stag hunt game in which each player has the utilities and play a co-operative game in order to increase their utilities. In the game of perfect information, the players lead to the strategies not only with the Nash equilibria but also with the sub games of the network. Based on the calculated metrics it can be believed as the router with maximum delay value is said to a highly congested router and the router with least value is said to be the low congested router [21]. After calculating this metric, the interface of the router can be chosen according to the algorithm and the game strategy so the decision of path from source to destination can be calculated to provide high performance of the flits by using a very less delay routers.

3.3. Congestion management model

Figure 3 shows the congestion management model. The proposed model SHG odd-even routing algorithm uses the game theoretical odd even turn routing algorithm. The adaptiveness is introduced to this by incorporating the router delay information that is obtained from the congestion detection in the model. Once the flit enters the network from the source node, the route towards the next router is found using the odd even adaptive routing. From the available paths, the path towards the router having the lowest delay has been selected. Thus, this adds to the adaptiveness in the SHG odd even turn model. The SHG odd even turn model rejects 2 out of 8 possible turns to avoid the formation a cycle and hence avoiding the formation of live lock or deadlock. The adaptiveness adds to the performance of the system, by incorporating the network status and the congestion information in the network. Figure 4 shows the restricted turns in odd and even columns of odd even routing. The two can be explained by two theorems.

- The packets are forbidden to take the paths East-North or East-South, if the current router lies on an even column.
- The packets are forbidden to take the paths South-East or North-East, if the current router lies on an odd column.

Thus, for the packets entering from west direction, west-north and west-south are permitted, only if the current router lies in even column. Similarly, for packets entering from east direction, east-north and east-south are permitted, only if the current router lies on odd column

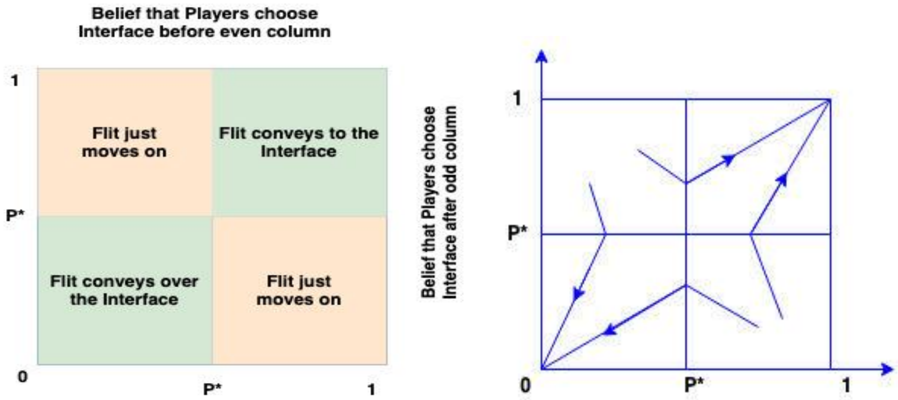

Figure 2. Probability of routing using stag hunt game

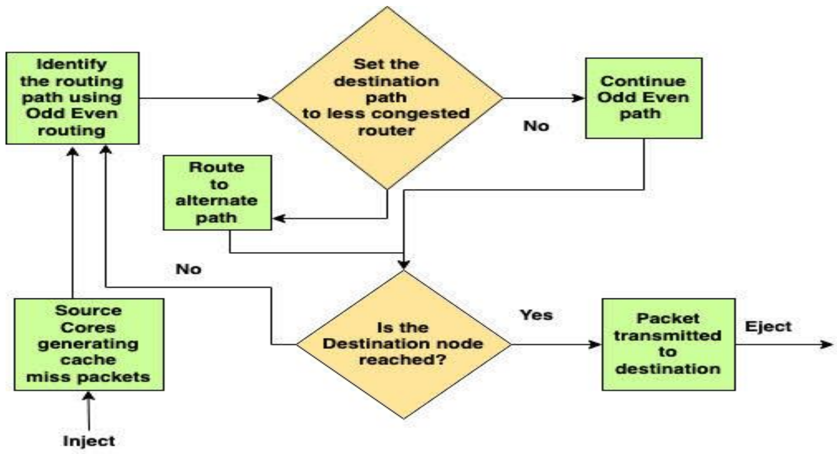

Figure 3. Congestion management model

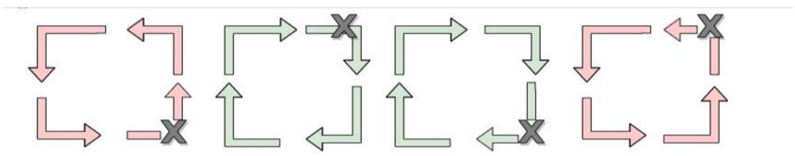

Figure 4. Restricted turns in odd even routing

3.4. Interactions between agents

Stag hunt is a pure strategy Nash equilibrium played among the routers' port position such as East-North/East-South for the even column, South-East/North-East for the odd column so that the flit traverse through the network in a reduced latency. So, the strategy is, if the previous route of the flit is in even column, it can choose any path other than East-North/East South interfaces to get the higher payoff. Suppose if there is any uncertainty about the other player's action maintaining of histories will support to provide a higher payoff. The more uncertainty actions being performed the more likely they will choose the strategy equivalent to it. The players prefer one over the other and achieve pareto optimal. At a time, t, the node i of core node say CNi will interact with another node j CNj. The flit has to pass through Ncx interface and the router Ncry. The selected interface and router adhere as in Algorithm 1 and each of the agents will choose its action based on its strategy. The strategy of node i with respect to j is chosen with the single time dependent probability P(t) of the existing interfaces to use the interface for the flit or choose alternate interface/delay the flit by rerouting it.

$$P(t+1) = \{p(t) + \lambda\beta(t)[1- p(t)]\} \quad (1)$$

where the flit choses the interface Ncx ∈ {Nci1, Nci2…. Ncnc} and if the flit choses the strategy other than Ncx ∈ {Nci1, Nci2…. Ncnc} then the probability is given by:

$$P(t+1) = \{p(t) - \lambda \beta(t)p(t)\} \quad (2)$$

where the flit choses the interface ⌐Ncx ∈ {Nci1, Nci2…. Ncnc}. β(t) is the output obtained and has a maximum payoff at time t with the learning rate λ.

The nash equilibrium is an action profile a∗ with the property that interface Ncx of the set {Nci1, Nci2... Ncnc} where Ncx ∈ {Nci1, Nci2…. Ncnc} can do better by choosing an action different from a ∗ x so that every other player say k adheres to a ∗ k.

However instead of taking the all the possible interfaces there exists a strategic correlation to attain the overall performance and reduced latency in the network. Algorithm 1 states that based on the coordinates of source and destination, flit and network interface of the SHG odd-even routing ejects the packets from forbidden route based on the algorithm.

The adaptiveness to the SHG odd even routing has been added, by selecting the best route from the available paths with the best available strategies. This is attained by selecting the path to the router that is having the less delay value and has higher payoff utility compared to the route to the other router that is having more delay values. The delay module finds the delays of the routers for every cycle. Thus, dynamic information is used for selecting the best path to transfer the packet to the next node and it is tested.

4. SIMULATION AND RESULTS

The performance of a many cores' processor depends immensely on the interconnection network. For evaluating the performance and for the user to be able to analyze the performance of the network, a simulator is used. The simulator used is BookSim2 manycore interconnect simulator. This simulator is a well accurate and flexible manycore interconnection simulator that supports many different topologies, routing algorithms and synthetic traffic patterns. Another add-on advantage is that unlike many other simulators, this simulator supports inter-router delays. For getting an optimized value for the NoC or the interconnection network used, the user has to change certain parameters and access the performance of the system. The simulator produces an output which is based on the current network topology, the routing algorithm, the synthetic pattern, the injection rate of the packets, the traffic patterns, the number of nodes in the network, the channel size, the buffer length. There are various other dependencies, which the simulator output can have. But for evaluation of network in majority of the simulations and the test cases, these inputs are sufficed.

The many cores interconnector network is tested with various configurations. Initially the configuration of the network that has to be built onto a SoC chip is decided. Here the topology is initialized first. Here the network is a mesh topology. The values for the number of dimensions denoted by 'n' and number of routers per dimension denoted by 'k' has to be provided. Once the topology is finalized, the router channel size indicated by 'vc_buf_size' and the number of virtual channels indicated by 'num_vcs' has to be provided. Here the vc_buf_size is selected as 8 and num_vcs as 8. The performance can be improved using queues like integrated queues or weighted fair queue [22]. Then the traffic pattern is selected. Here the transpose traffic pattern and uniform random traffic pattern is used. The uniform random traffic is a synthetic traffic pattern to imitate the real time scenario of many core processors. Once the traffic pattern is selected the network is tested with various injection rates in terms of number of flits injected per second by a core towards a channel. The value of injection rate ranges from 0.0 to 1.0. The network is simulated with the above configurations, tested and the results are analyzed.

The latest flit and the slowest flit id as well can also be analyzed. Once the slowest flit id is analyzed, the path of the flit has been traced out from the trace-out file. The trace-out file will be able to display the processes that a flit had undergone from the point the flit got injected into the network, till the flit gets ejected out from the network. The delay of each router for processing the flit can be analyzed and noted down and the maximum of those delay values will be able to provide the router id that has incurred most of the delay to that flit. The network is again simulated by including a code for finding the router delay of each flit. The maximum delay incurred by each router is analyzed and checked whether the delay has been found present in the above noted down value.'The main aim of SHG odd-even routing is to deroute the packets in such a way as described by the SHG odd-even routing algorithm. The SHG odd-even routing algorithm ensures that, at a point of time during routing, there will be atleast more than one path available for making a routing decision for a router. This routing possibilities are produced in such a way that, it will ensure a deadlock free routing for the packet, if the packets are transferred though any of the available routes. Each router is analyzed for how much time, the flit is being held for each router, to get transferred to the next. The max delay shows the maximum value of the delay incurred by a router to the flit. This value is assigned in such a way that; it is the maximum of the delays of each router incurred to whatever flits that are passed through the respective routers.

Algorithm 1. SHG odd-even routing algorithm

Input:

Flit FP, Source Coordinates (Cx, Cy), Destination Coordinates (Dx, Dy) Core Nodes CN = {CN1, CN2, … CNc}, Network Interface Ni = {Nci1, Nci2, … Ncnc} and Router Nodes Nrpv = {Nr1, Nr2, Nr3, … Nrc}.

Output:

Best possible interface Ncx of the set {Nci1, Nci2…, Ncnc} and Expected edges E, Coordinates (Ex, Ey)

Pseudocode:

```
Calculate Ex=Dx-Cx, Ey=Dy-Cy (Cx, Cy) are the current node coordinated.
If Ex == 0 && Ey == 0 then Eject the port interface, End if
If Ey == 0 then
        If Ex < 0 then, Set North port interface available
          Else Set South Port interface available, End if
End if
If Ey > 0 then
    If Ex = 0 then Set East port interface available, End if
If Cy%2! = 0 || Cy == Sy then Set North port interface available Else Set South Port interface available
If Dy%2! = 0 || Ey != 1    Set East port interface available
Else Set West Port Interface available
If Cy %2 == 0 && Ex < 0      Set North port interface available
Else    Set South port interface available
End
```

To give more clarity to the above data, the count of how many times each router is being causing the highest delay to each and every flit, is also found. The results of the adaptive_xy_yx routing model with credit based adaptive ness is compared with the adaptive xy_yx model with router delay-based adaptiveness for various injection rates. The results have been calculated for uniform and transpose traffic as well as for 8x8 mesh and 16x16 mesh for detailed analysis. Each router is analyzed for how much time, the flit is being held for each router, to get transferred to the next. The max delay shows the maximum value of the delay incurred by a router to the flit. This value is assigned in such a way that, it is the maximum of the delays of each router incurred to whatever flits that are passed through the respective routers. To give more clarity to the above data, the count of how many times each router is being causing the highest delay to each and every flit, is also found. This obtained router delay is used for adaptiveness in the routing decision.

Figure 5 shows the comparative analysis for an 8x8 mesh network for Adaptive xy_yx with credit adaptiveness and Adaptive xy_yx with router delay adaptiveness for the uniform and transpose traffic respectively. Figure 6 shows the comparative analysis for a 16x16 mesh network for Adaptive xy_yx with credit adaptiveness and Adaptive xy_yx with router delay adaptiveness for the uniform and transpose traffic respectively. The Figures 5 and 6 show that the Adaptive xy_yx with router delay adaptiveness tends to give almost the same efficiency as of the Adaptive xy_yx with credit adaptiveness. But for increased no of nodes in the multicore processor, that is, 16 nodes, the Adaptive xy_yx with router delay adaptiveness has a comparable performance advantage compared to the Adaptive xy_yx with credit adaptiveness.

Figure 7 shows the comparative analysis for an 8x8 mesh network for odd even routing with router delay adaptiveness and odd even routing with probabilistic routing, for the uniform and transpose traffic respectively. The OE delay model has improved performance compared to the OE probabilistic model. Figure 8 shows the comparative analysis for a 16x16 mesh network for odd even routing with router delay adaptiveness and odd even routing with probabilistic routing, for the uniform and transpose traffic respectively. From the Figures 7 and 8 odd even routing with router delay adaptiveness has a greater performance advantage compared to the odd even probabilistic routing. The OE delay model has improved latency compared to the OE probabilistic model. The Figure 9 shows the delay avg. packet latency comparison of AD delay xy_yx and OE delay of 16x16 mesh network for uniform and transpose traffic respectively. From the above figures, it is evident that the odd even adaptive delay method has a improved performance over the adaptive xy_yx method. The OE delay model has comparable latency compared to the ad xy_yx routing for uniform traffic.

The following are the inferences from the latency calculations:

- The Adaptive odd even routing has improved performance over the adaptive xy_yx routing for larger sized NoCs over any traffic pattern
- The Adaptive odd even delay adaptive routing and the adaptive xy_yx routing has almost similar performance metrics over the adaptive xy_yx routing for smaller sized NoCs
- the odd even delay adaptive routing has improved performance characteristics compared to the odd even probabilistic model.
- The adaptive delay xy_yx model has improved performance over the adaptive xy_yx model.

Figure 10 represents the router delay characteristics for OE delay adaptive routing and adaptive xy_yx routing for 8x8 uniform mesh NoC.The delay overshoots are maximum for the adaptive xy_yx mesh NoCs and the odd even delay adaptive routing is able to distribute the delays for the routers more evenly compared to the ad xy_yx routing.

It is evident that the delays are more evenly distributed to the routers for the OE delay adaptive routing compared to the adaptive xy_yx routing. Thus, shows and higher percentage improvement in overall reduction of system latency for the packets in OE delay adaptive routing. Along with low latency, cluster-based energy efficiency routing algorithms can also be used [23]. A centralized scheme can be made as thread-based [24], and a thread can be used for this kind of utility function [25].

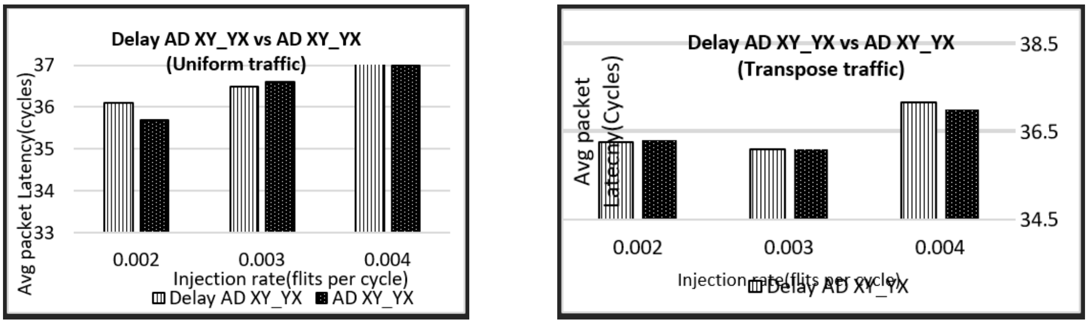

Figure 5. Uniform and transpose traffic on 8x8 adaptiveness delay

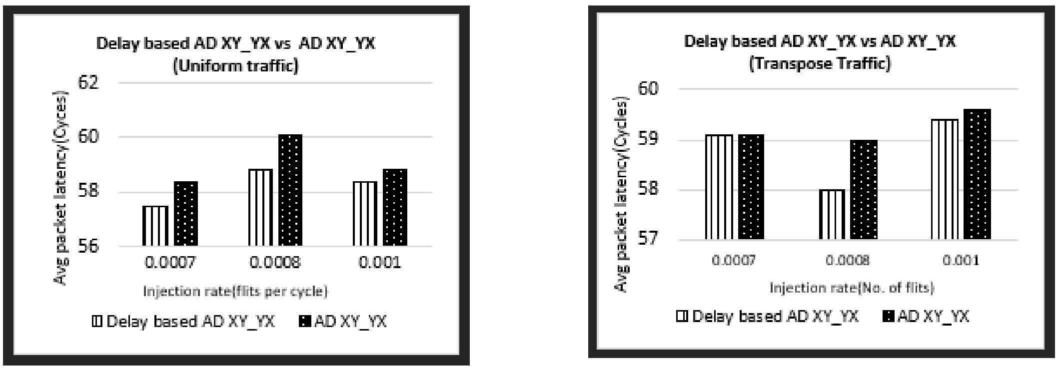

Figure 6. Uniform and transpose traffic on 16x16 adaptiveness delay

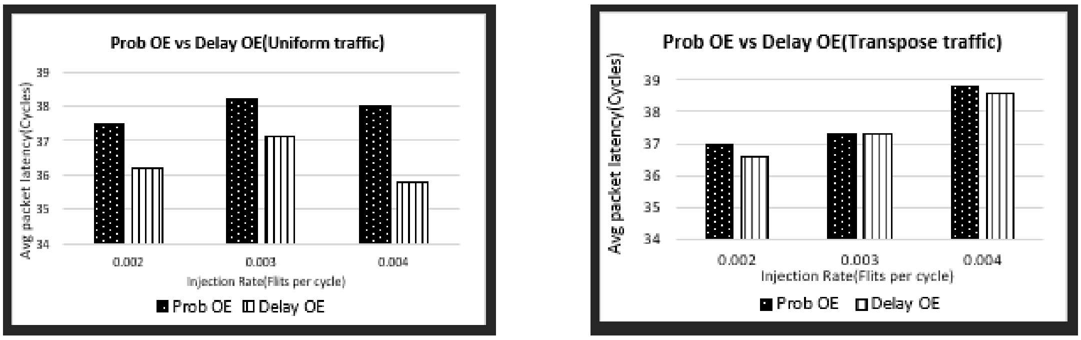

Figure 7. Uniform and transpose traffic analysis on 8x8 nodes

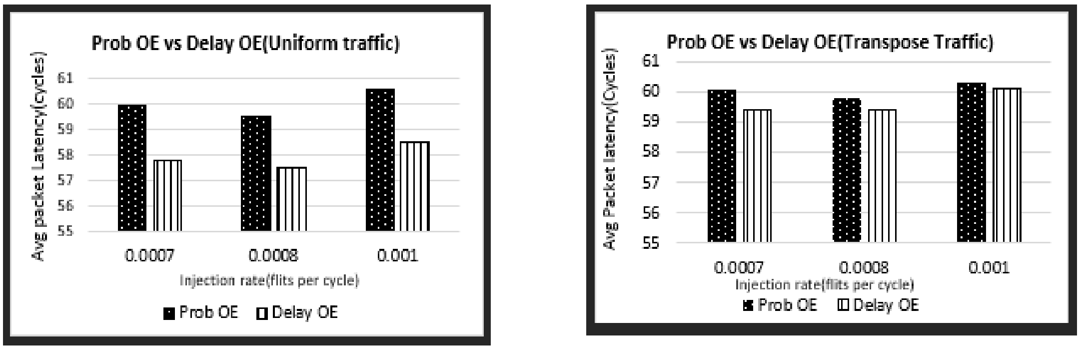

Figure 8. Uniform and transpose traffic analysis on 16x16 nodes

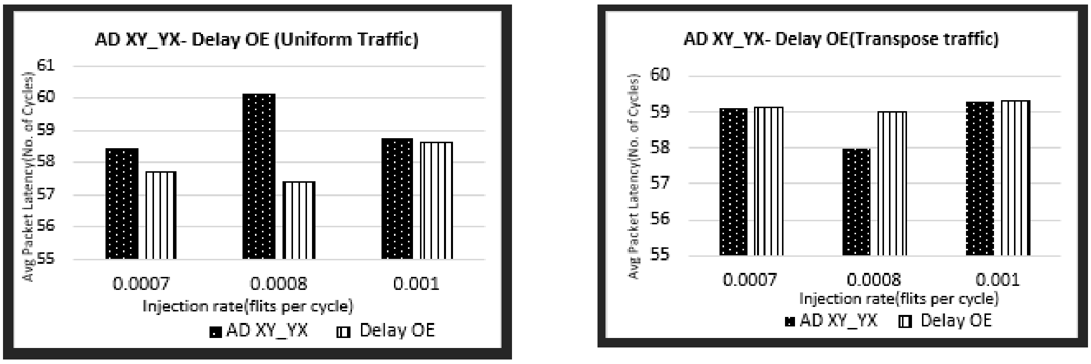

Figure 9. Uniform and transpose traffic analysis over latency calculation on 16x16 nodes

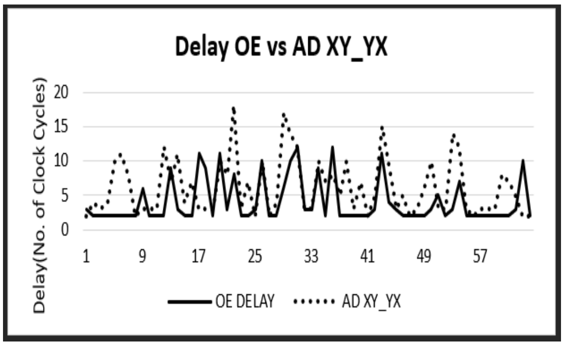

Figure 10. Router delay characteristics over 8x8 nodes

5. CONCLUSION AND FUTURE WORK

Most of the existing and current research are based on what happens after congestion. Also, it proceeds into an assumption that the routers in a manycore processor are prone to congestion and henceforth mitigate to some routing mechanisms so as to avoid congestion at the later stages. And often mentions that the congested nodes are those nodes, that injects more packets into the network at a time which is not at all a guaranteed assertion. The intend of doing this research was to deal with the problem statement, that is, to identify the congestion within a network using game theoretical method and to reroute the packets towards the least congested nodes. The work, first calculates the congestion in the nodes or routers, using the processing delay of each flit, and records the highest delay. The router delay experienced by the nodes is found to showcase the congestion status of the respective routers. The proposed work uses Odd even Routing Strategy with Delay Adaptiveness along with the game model approach, for routing the flits to the destination, and have found that this routing strategy has improved latency compared to the Deterministic routing, Probabilistic Odd even routing, Adaptive XY_YX routing, Probabilistic XY_YX routing. In addition to this, applying delay adaptiveness to the already existing routing methods, also has resulted in improved performance compared to the XY_YX based adaptiveness. On-chip algorithms can be designed based on game theory for better understanding and performance.

REFERENCES

[1] J. Marri, S. Manishankar, D. Radha, and M. Moharir, “Implementation and analysis of adaptive odd-even routing in booksim 2.0 simulator,” in *Proceedings of the 4th International Conference on Communication and Electronics Systems, ICCES 2019*, Jul. 2019, pp. 76–83. doi: 10.1109/ICCES45898.2019.9002391.

[2] M. N. V. S. Saiteja, K. S. S. Reddy, D. Radha, and M. Moharir, “Multi-core architecture and network on chip: Applications and challenges,” *Journal of Computational and Theoretical Nanoscience*, vol. 17, no. 1, pp. 239–245, Jan. 2020, doi: 10.1166/jctn.2020.8657.

[3] P. Gratz, B. Grot, and S. W. Keckler, “Regional congestion awareness for load balance in networks-on-chip,” in *Proceedings - International Symposium on High-Performance Computer Architecture*, Feb. 2008, pp. 203–214. doi: 10.1109/HPCA.2008.4658640.

[4] A. Khafidin, T. Andrasto, and Suryono, “Implementation flow control to improve quality of service on computer networks,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 3, pp. 1474–1481, Dec. 2019, doi: 10.11591/ijeecs.v16.i3.pp1474-1481.

[5] O. Surakhi, M. Khanafseh, and S. Sarhan, “A survey on parallel multicore computing: Performance & improvement,” *Advances in Science, Technology and Engineering Systems*, vol. 3, no. 3, pp. 152–160, Jun. 2018, doi: 10.25046/aj030321.

[6] R. Arul *et al.*, “Intelligent data analytics in energy optimization for the internet of underwater things,” *Soft Computing*, vol. 25, no. 18, pp. 12507–12519, Jul. 2021, doi: 10.1007/s00500-021-06002-x.

[7] R. Jayaraman, G. Raja, D. Ghosal, R. Arul, and S. Kumar, “A compatibility vector technique for cooperative scheduling and channel assignment algorithm in broadband wireless networks,” *Mobile Networks and Applications*, vol. 22, no. 4, pp. 730–742, Mar. 2017, doi: 10.1007/s11036-017-0841-x.

[8] D. E. Charilas and A. D. Panagopoulos, “A survey on game theory applications in wireless networks,” *Computer Networks*, vol. 54, no. 18, pp. 3421–3430, Dec. 2010, doi: 10.1016/j.comnet.2010.06.020.

[9] M. Daneshtalab, A. Sobhani, A. Afzali-Kusha, O. Fatemi, and Z. Navabi, “NoC hot spot minimization using AntNet dynamic routing algorithm,” in *Proceedings of the International Conference on Application-Specific Systems, Architectures and Processors*, 2006, pp. 33–36. doi: 10.1109/ASAP.2006.49.

[10] J. Kim, D. Park, T. Theocharides, N. Vijaykrishnan, and C. R. Das, “A low latency router supporting adaptivity for on-chip interconnects,” in *Proceedings. 42nd Design Automation Conference, 2005.*, 2008, pp. 559–564. doi: 10.1109/dac.2005.193873.

[11] R. S. Ramanujam and B. Lin, “Destination-based adaptive routing on 2D mesh networks,” 2010. doi: 10.1145/1872007.1872030.

[12] R. S. R. Raj, A. Das, and J. Jose, “Implementation and analysis of hotspot mitigation in mesh NoCs by cost-effective deflection routing technique,” Oct. 2017. doi: 10.1109/VLSI-SoC.2017.8203461.

[13] M. Debnath, D. Konstantinou, C. Nicopoulos, G. Dimitrakopoulos, W. M. Lin, and J. Lee, “Low-cost congestion management in networks-on-chip using edge and in-network traffic throttling,” in *ACM International Conference Proceeding Series*, Jan. 2017, vol. Part F128361, pp. 8–11. doi: 10.1145/3073763.3073764.

[14] M. Auxilia, K. Raja, and K. Kannan, “Cloud-based access control framework for effective role provisioning in business application,” *International Journal of System Dynamics Applications*, vol. 9, no. 1, pp. 63–80, Jan. 2020, doi: 10.4018/ijsda.2020010104.

[15] S. Blagodurov, S. Zhuravlev, and A. Fedorova, “Contention-aware scheduling on multicore systems,” *ACM Transactions on Computer Systems*, vol. 28, no. 4, pp. 1–45, Dec. 2010, doi: 10.1145/1880018.1880019.

[16] A. A. Alkadhmawee, M. A. Altaha, and W. M. Lafta, “Unequal clustering algorithm with IDA∗ multi-hop routing to prevent hot spot problem in WSNs,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 1, pp. 445–453, Oct. 2020, doi: 10.11591/ijeecs.v20.i1.pp445-453.

[17] M. Gansterer and R. F. Hartl, “The Prisoners’ Dilemma in collaborative carriers’ request selection,” *Central European Journal of Operations Research*, vol. 29, no. 1, pp. 73–87, Jan. 2021, doi: 10.1007/s10100-020-00717-2.

[18] M. Belloc, E. Bilancini, L. Boncinelli, and S. D’Alessandro, “Intuition and deliberation in the stag hunt game,” *Scientific Reports*, vol. 9, no. 1, Oct. 2019, doi: 10.1038/s41598-019-50556-8.

[19] S. Anbalagan, D. Kumar, G. Raja, and A. Balaji, “SDN assisted stackelberg game model for LTE-WiFi offloading in 5G networks,” *Digital Communications and Networks*, vol. 5, no. 4, pp. 268–275, Nov. 2019, doi: 10.1016/j.dcan.2019.10.006.

[20] K. Kaliyan and R. Kothandaraman, “Secure decision-making approach to improve knowledge management based on online samples,” *International Journal of Intelligent Engineering and Systems*, vol. 11, no. 1, pp. 50–61, Feb. 2018, doi: 10.22266/ijies2018.0228.06.

[21] N. J. C. Abraham and D. Radha, “Detection and analysis of congestion of nodes in many-core processor,” in *Advances in Intelligent Systems and Computing*, vol. 1045, Springer Singapore, 2020, p. 755. doi: 10.1007/978-981-15-0029-9_59.

[22] T. A. Assegie and H. D. Bizuneh, "Improving network performance with an integrated priority queue and weighted fair queue scheduling," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 1, pp. 241–247, Jul. 2020, doi: 10.11591/ijeecs.v19.i1.pp241-247.
[23] G. N. Basavaraj and C. D. Jaidhar, "Low latency and energy efficient cluster based routing design for wireless sensor network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 13, no. 2, pp. 615–625, Feb. 2019, doi: 10.11591/ijeecs.v13.i2.pp615-625.
[24] G. C. Chasparis, M. Rossbory, V. Janjic, and K. Hammond, "Learning-based dynamic pinning of parallelized applications in many-core systems," in *2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, Feb. 2019, pp. 1–8. doi: 10.1109/EMPDP.2019.8671569.
[25] J. Wu, B. Jiang, and H. Xu, "Design of on-chip bus arbitration algorithm based on game theory," in *2021 13th International Conference on Communication Software and Networks (ICCSN)*, Jun. 2021, pp. 34–38. doi: 10.1109/ICCSN52437.2021.9463653.

BIOGRAPHIES OF AUTHORS

Radha Doraisamy is currently working as an Assistant Professor, Department of CSE at Amrita School of Engineering, Amrita Vishwa Vidyapeetham. She is pursuing her doctoral sudies in Computer Science and Engineering, VTU. She completed M. Tech (CSE) from Dr. MGR University, and B.E (CSE) from Madurai Kamaraj University, India. She has 20+ years of teaching and research experience. Her areas of interest are Data Structures, Cryptography, High performance computing, Multicore architecture, Image processing and web technology. She has published around 25+ technical papers in National/International Journals and conferences. She can be contacted at email: d_radha@blr.amrita.edu.

Prof. Minal Moharir is presently working as a Professor, Department of CSE at R V College of Engineering. Her Academic Qualification includes Ph.D. in Information and Network Security (Avinashilingam University, Coimbatore), M. Tech (VTU) in Computer Network Engineering, and B.E in Computer Science and Engineering (GECA). Her 18-years of experience includes teaching, training and research. The professional expertise includes Computer Networks, Data Communication, Information Security, Wireless sensor Networks, IoTs and High-Performance Computing & GPU Computing. She is a research supervisor of Six scholars, who are pursuing doctoral program in Computer Networks and High-Performance Computing. She has executed/executing projects and consultancy works funded by private funded agencies. She has published and presented more than 40 technical papers in National and International Journals/conferences. She can be contacted at email minalmoharir@rvce.edu.in.

Rajakumar Arul is currently associated with the Centre for Smart Grid Techonologys/ School of Computer Science and Engineering (SCOPE), VIT Chennai. He pursued his Bachelor and Master's in Computer Science and Engineering from Anna University, Chennai. He completed his Doctor of Philosophy under the Faculty of Information and Communication, Department of Computer Technology, Anna University - MIT Campus. He is a recipient of Anna Centenary Research Fellowship for his doctoral studies. His research works has been published in the journals of IEEE transactions of Industrial Informatics, Springer Wireless personal Communication, Springer Mobile Networks and Applications, Wiley Transactions on Emerging Telecommunication Technologies etc. He has received Amrita Chancellor's Research award for his research contributions in 2021. His research interests include BlockChain, Security in Broadband Wireless Networks, LTE, Robust resource allocation schemes in Mobile Communication Networks, Smart Grid and Cryptography. He is a senior member of IEEE. He can be contacted at email: rajakumararul@ieee.org.