An Efficient Deterministic Edge Traffic Distribution Network-on-chip Routing Algorithm Design

Cameron Lane, Eric Muller, Pedro Valencia, Yefa Mai, Nan Wang
Department of Electrical and Computer Engineering
California State University, Fresno
Fresno, USA
camlane, ericmuller, peterv91, maiyf1991119@mail.fresnostate.edu, nwang@csufresno.edu

***Abstract*— In recent decades, computer systems have been advanced from relatively simple single-core CISC and RISC architectures to more complex multi-core system-on-chip designs with higher inter-core communication requirements. Network-on-chip architectures emerged as promising solutions for future system-on-chip communication architecture designs. However, the switching and routing algorithm design of network-on-chip communication architectures are still facing great challenges. To address the deficiencies of the existing routing algorithms, a new deterministic network-on-chip routing algorithm-- Edge Traffic Distribution routing and its specifics are proposed in this paper. In the proposed algorithm, the traffic which involves the Edge Routers will be directed to go through the edges first instead of entering into the center of the network. It will greatly alleviate the congestion around the center of the network so as to help improve the overall system performance. Simulation of the proposed Edge Traffic Distribution algorithm and three existing routing algorithms has been carried out using NIRGAM network-on-chip simulator. Simulation results illustrate that the performance of the Edge Traffic Distribution routing algorithm exceeds the performance of the existing routing algorithms in term of average latency and network power consumption.**

Keywords—network-on-chip; routing algorithm; power consumption; communication latency

I. INTRODUCTION

Through recent years, as the technology scales toward deeper submicron, more and more IP components can be integrated into a single chip. Intel has started to manufacture and ship out 15-core Xeon server chips integrated with 4.31 billion transistors and running at clock speeds varying from 1.4GHz to 3.8 GHz[1]. This has promoted the development of high performance embedded platforms that can support the computation and communication requirements of recent complex applications, which could not be handled in traditional single processor architectures. The on-chip interconnection between the computing resources, such as microprocessors, DSPs, SRAMs and other functional cores, becomes a very challenging issue [2].

Shared bus architectures can no long meet the increasing communication demands. Network-on-chip (NOC) architecture has been proposed as a viable solution for the increasing complexity of on-chip communication problems due to following reasons: energy efficiency, reliability, scalability of bandwidth, reusability and distributed routing decisions compared to shared-bus based architecture. NOC architecture employs an on-chip packet switching micro network to support the communication between the master components as depicted in Fig. 1.

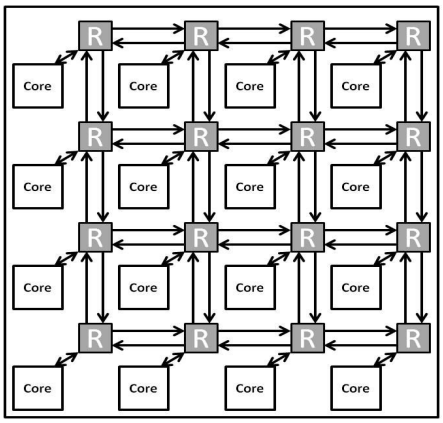

Fig. 1. 4 X 4 2-dimensional mesh NOC.

NOC architecture has the scalable and modular features of general networks, however, the high design complexity and complex configuration features of NOC architecture make it very difficult to implement the communication schemes of general networking as the on-chip communication methodology [3,4].

A NOC system is comprised of IP cores, routers as well as wires that connect the routers together. All the IP cores, such as microprocessors, memories and other cores are connected to the routers. The routers are responsible for all the communication tasks while the IP cores focus on the more important computations and other operations. NOC design consists of: topology design, routing algorithm design, flow control, switching algorithm design and switch architecture design. Switching algorithms decide how to get data from an input channel to the proper output channel inside the routers. Routing algorithms define the route for a packet to be delivered from source to destination routers on the network [5].

Routing algorithm is a key element for efficient NOC communication system design. There are two major types of NOC routing algorithms, one is called deterministic and another one is called adaptive routing algorithm. Deterministic routing always selects the same path between any two nodes. The route taken is determined before the packet is sent out, regardless of traffic conditions. Adaptive routing algorithm, on the other hand, chooses the path according to the current network status, such as the depth of the queues and network loads. The deterministic routing algorithms work fine under

low network loads, and are generally simple and low power. The adaptive routing algorithms perform better under high network loads, but consume more power than the deterministic ones. It is equally important to develop more efficient deterministic and adaptive routing algorithms.

In this paper, several existing NOC routing algorithms and their limitations will be briefly discussed. Then the proposed deterministic NOC routing algorithm-- Edge Traffic Distribution (ETD) routing algorithm along with description of the simulation and analysis of the results will be presented.

The routers on a NOC can be categorized into two types, one is Edge Routers which are located on the edges of the network, and the other type is Internal Routers. In existing routing algorithms, most of the communication traffic are passed through the Internal Routers, especially the center routers. This creates congestion and sometimes deadlock in the center of the network. It is important to evenly distribute the traffic all over the network. The primary concept of the ETD routing is to direct the traffic involving the Edge Routers to go through the edges first instead of entering the inside of the network immediately. A detailed description of the proposed ETD routing algorithms is presented in Section III.

The organization of this paper is as follows. Section II discusses the existing popular routing algorithms, XY, Odd-Even and DyAD routings as well as some other approaches. Section III presents the proposed deterministic ETD routing algorithm and its specifics. The simulation results and analysis are presented in Section IV. Finally, Section V concludes the paper.

II. Related work

XY, Odd-Even [5], and DyAD[6] are among the most popular traditional routing algorithms. TA [7], NoP [8], RCA [9] and DAR [10] routing algorithms are other recent proposed methods.

A. XY Routing

XY is one of the popular deterministic routing algorithms. In XY algorithm, packets will be transported along X direction to get to the destination column, and then along Y direction to reach their final destination.

To implement XY algorithm, a router (X,Y) always compares its current X location to the packet X-destination. If it is greater or lesser than, the packet will be moved in the west or east direction. If it is a match, the router will then compare its current Y location to the packet Y-destination. The packet will be moved to north or south direction based on the result. The packet will be delivered to the local IP core only if both of the X and Y locations match the packet X and Y destination.

The XY algorithm is simple and easy to implement. On the other hand, it suffers from deadlock and central congestion problems.

B. Odd-Even Routing

Odd-Even routing is claimed to be an adaptive routing algorithm. For the routers in odd columns, the turn model north-to-west and south-to-west turns are banned and for the routers in even columns, east-to-south and east-to-north turns are banned. Between the directions where the router can send packets, the direction in which the neighboring router has less empty slots in its input buffer is selected. In this algorithm each router keeps track of the packet depth of the input buffer of its adjacent neighbors. The routers will send a message to the upstream neighbors to inform the packet depth of its corresponding input buffers [11].

Odd-Even routing is said to be deadlock-free, but has overheads in message exchanging between routers and buffer status tracking.

C. DyAD Routing

DyAD routing dynamically employs a deterministic (generally XY) or an adaptive routing (Odd-Even) based on different network congestion situations.

The depth of the four input buffers of all the routers will be monitored. For any router, if one of input buffers reaches a predefined congestion threshold, a congestion mode flag will be set to inform the neighboring routers about the situation. All routers will check the mode flags of the neighboring routers to decide which of the deterministic routing or adaptive routing will be used to forward the packets. One congestion mode flag will be enough to trigger the switching of the routing algorithms [12].

DyAD routing algorithm is more complicated but efficient compared to XY and OE algorithms. It suffers from overheads for the routers to check mode flags of the neighbors.

D. Other Routing Algorithms

In TA (Traffic Allocation) routing, a traffic allocation register is added for each of the four outgoing directions. Instead of checking depth of the adjacent incoming buffers, a router only needs to monitor the traffic load of its four outgoing directions. The direction with lower value in its traffic allocation register will be chosen to forward a packet.

In NoP (Neighbors-on-Path) routing, dedicated wires are added to the architecture in order to exchange status information. It focuses on allocating the channels that will allow the packets to be routed to their destination along a path that is as free of congested nodes as possible. RCA (Regional Congestion Awareness) routing informs the routing policy of congestion in parts of the network beyond adjacent routers. The congestion information is across the network in scalable manner. In DAR (Destination-Based Adaptive) routing, every node maintains per-destination congestion state in the form of average delays to all other nodes through the candidate output ports permitted by minimal adaptive routing.

III. Proposed Edge Traffic Distribution Routing Algorithm

To address the limitations of the existing routing algorithms, a new deterministic routing algorithm, Edge Traffic Distribution (ETD) routing is proposed.

A NOC system consists of routers and wires. All the routers are identical except the router IDs which consists of their X and Y locations. The existing routing algorithms suffer from various problems, such as head-of-line, centric

congestion, deadlock, and etc. To alleviate the problems, the ETD routing is presented to distribute the traffic away from the Internal Routers to the Edge Routers. The routers in an NOC network can be categorized into two types, Edge Routers and Internal Routers as shown in Fig.2.

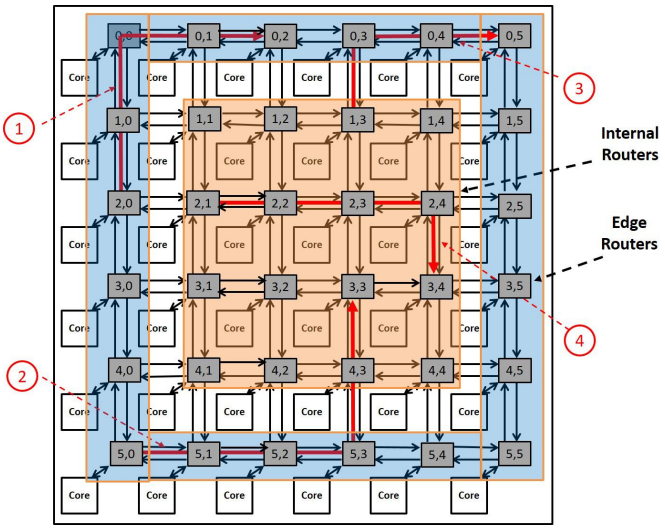

Fig. 2. Traffic patterns in the ETD routing algorithm

The primary rule is all the packet transportations have to take the shortest path. Based on this condition, four traffic patterns are taken into consideration of the ETD routing algorithm design as described as follows:

- **Case 1. Edge Router to Edge Router:** the packets travel along the edges only to the destination if the destination router is on the adjacent edges to the source router. If the destination router locates on the across edge, the packets move on the edge first if necessary, then go to the destination router directly through the Internal Routers. For example: (2,0) to (0,2), the packets travel on the west edge to (0,0), then on the north edge to the destination router (0,2).
- **Case 2. Edge Router to Internal Router:** the packets move on the edge first if necessary, then go to the destination router directly through the Internal Routers. If the source router happens to be a corner router, the packets need to move on the longer edge. Y-direction edge wins the tiebreak. For example: (5,0) to (3.3), the packets move on the south edge first instead of the west edge because the packets can travel longer (3 hops) on the south edge than on the west edge (2 hops); then through the Internal Routers (4,3) to the destination router (3,3).
- **Case 3. Internal Router to Edge Router:** the packets are first delivered to the closest edge through the Internal Routers, then to the destination. Y-direction edge wins the tiebreak. For example: (1,3) to (0,5), the packets move to the closer north edge, then to the destination router (0,5).
- **Case 4. Internal Router to Internal Router:** the packets are moved to the destination router using XY routing. For example: (2,1) to (3,4), the packets travel to X-direction, then Y-direction to the destination router (3,4).

As shown in Fig.2, packets with the different traffic patterns follow their designated paths with reduced interference. A great deal of traffic has been distributed from the center to the perimeter of the network. In Case 4, we choose XY routing algorithm when the packets are moved from an Internal router to another because of its characteristics of low power and low complexity.

IV. Simulation Result and Analysis

The simulation is carried out based on 2-dimension 6X6 mesh NOC architecture on NIRGAM 2.1 simulator. NIRGAM is an open source discrete event simulator for simulating different operation cases on NOC. It is a SystemC/C++ based simulator[13]. The proposed ETD routing algorithm is added to the simulator and tested along with the existing XY, OE and DyAD routing algorithm under the same NOC topology and traffic configuration.

A. Simulation Set Up

The NIRGAM simulation parameters can be configured in three files. In "nirgam.config", you can define NOC topology type, size, routing algorithm, clock frequency, flit size, number of warm up clock cycles and total simulation clock cycles. In "application.config", the role of the tiles can be defined. In our simulations, the tiles were attached to a Constant Bit Rate (CBR) traffic generator. Packet size, network load, and destination information can be defined in the tile-n files found in $NIRGAM/config/traffic [13]. The simulation result can be found in sim_result and graphic result file.

In the simulation, the packet size is set at 20 bytes with random destination mode. The simulation percentage loads from 10% to 90% in steps of 10%. The clock runs at 1GHz and the simulation runs for 50,000 clock cycles with warm up period of 800 clock cycles. The structure of NIRGAM simulator is shown in Fig.3.

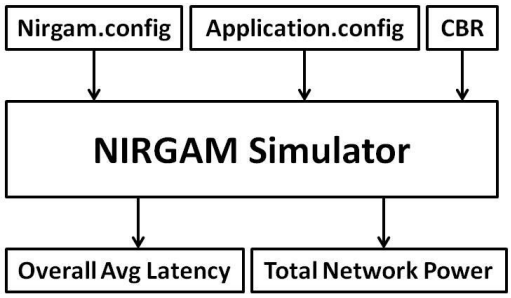

Fig. 3. NIRGAM simulator structure.

B. Simulation Results

To evaluate performance of the four routing algorithms, total network power consumption (milliWatts), the overall average latency in clock cycle per flit and the throughput in average flits per millisecond under all nine network loads in percentages from 10% to 90% are measured. To precisely evaluate performance of the routing algorithms under different network load levels, the average of total network power

consumption, communication latency in clock cycle per flit and throughput in flits per millisecond was calculated for network loads from 10% to 30%, 40% to 60% and 70% to 90% representing performance of the four routing algorithms under low, mid and high network load degrees respectively.

The results of Network Loads vs. Total Network Power Consumption for XY, OE, DyAD and the proposed ETD routing algorithms are shown in Table I and Fig.4.

TABLE I. TOTAL NETWORK POWER OF THE ROUTING ALGORITHMS

Loads	XY	OE	DyAD	ETD
10%	13.339	16.4339	13.3909	12.101
20%	25.2734	31.7	25.0217	23.72
30%	31.3019	39.637	31.1679	30.4723
40%	42.1943	52.3314	56.3301	40.5613
50%	62.3058	68.6231	62.2054	60.5228
60%	62.1858	62.1858	62.1903	60.5883
70%	62.535	67.7153	66.6585	60.2573
80%	62.6296	68.1021	67.3256	60.6435
90%	62.1195	67.7392	66.6569	59.8437

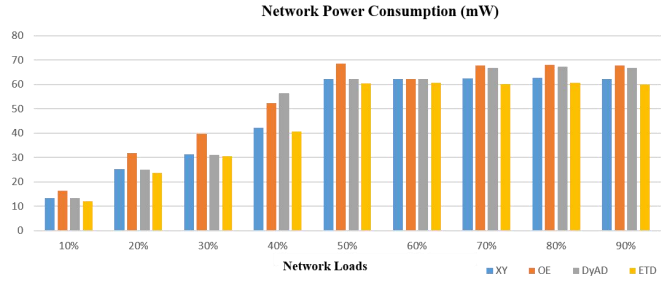

Fig.4. Total Network Power of the routing algorithms.

Simulation results of Network Loads vs. Average Latency in clock cycle per flit for XY, OE, DyAd and the proposed ETD routing algorithms are shown in Table II and Fig. 5.

TABLE II. OVERALL AVERAGE LATENCY UNDER NETWORK LOADS

Loads	XY	OE	DyAD	ETD
10%	43.2656	52.652	43.8445	41.1774
20%	42.3411	53.6972	42.2145	41.5537
30%	42.4873	56.1594	42.5541	42.5149
40%	43.8165	63.0111	43.7522	42.8236
50%	47.2198	106.528	75.6226	46.6093
60%	47.8017	111.9218	109.5462	46.8745
70%	47.4432	119.36	118.5612	46.3269
80%	48.6428	118.811	117.3476	46.4288
90%	49.3433	118.072	117.5734	46.0036

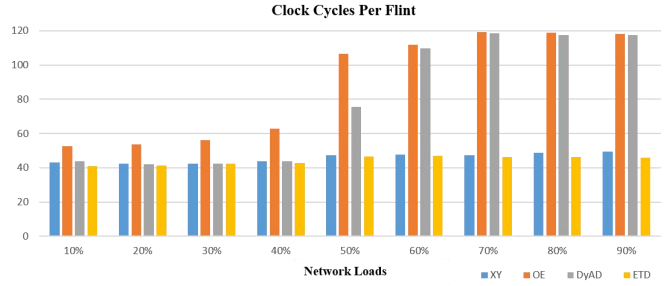

Fig. 5. Overall Average Latency in clock cycles/flit

Table III shows Average Network Power Consumption under low, mid and high network loads. Fig. 6 shows the graphical representation for data of Table III.

TABLE III. AVERAGE TOTAL NETWORK POWER CONSUMPTION UNDER LOW, MID AND HIGH NETWORK LOADS

Loads	XY	OE	DyAD	ETD
Low	23.30477	29.25697	23.1935	22.09777
Mid	55.56197	61.04677	60.24193	53.8908
High	62.428	67.8522	66.88033	60.24817

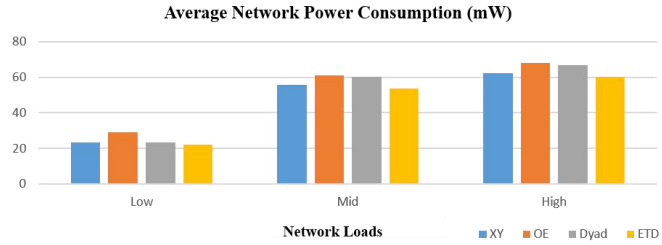

Fig.6. Average Total Network Power under Loads.

Table IV shows Average Network Power under low, mid and high network loads. Fig.7 shows the graphical representation for data of Table IV.

TABLE IV. AVERAGE OVERALL LATENCY IN CLOCK CYCLE PER FLIT UNDER LOW, MID AND HIGH NETWORK LOADS

Loads	XY	OE	DyAD	ETD
Low	42.698	54.16953	42.8710	41.7487
Mid	46.27933	93.8203	76.3041	45.4358
High	48.4764	118.7477	117.8233	46.2531

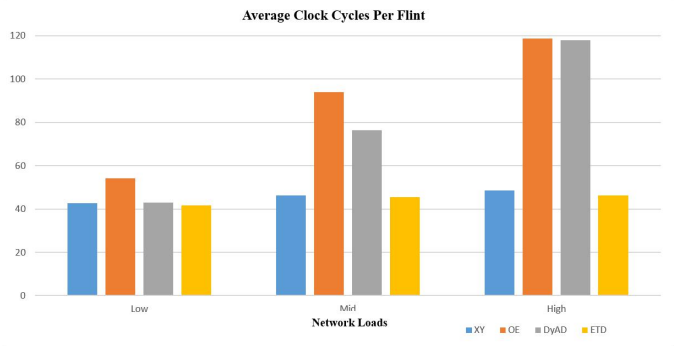

Fig.7. Average latency in clock cycle per flit

Table V and Fig.8 show Throughput in average flits per millisecond under low, mid and high network loads.

TABLE V. THROUGHPUT IN AVERAGE FLITS PER MILLISECOND

Loads	XY	OE	DyAD	ETD
Low	23.4203	18.4606	23.3258	23.9529
Mid	21.6079	10.6587	13.1055	22.0091
High	20.6286	8.4212	8.4873	21.6202

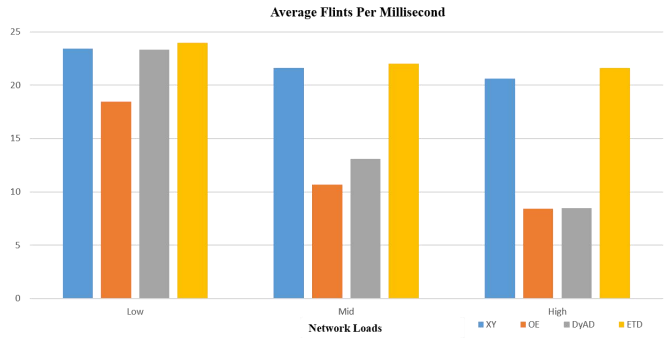

Fig.8. Throughput in average flits per millisecond

C. *Results Justification*

All four tables and graphs display the advantages of the ETD routing over three of the most popular routing algorithms.

- From Table I and Fig.4 it can be seen that the proposed ETD routing algorithm consumes less total network power under almost all 10% - 90% network load levels.
- Table II and Fig.5 both portray the effectiveness of using the ETD routing algorithm. The ETD showed better results, consistently throughout all of the varying loads when compared against XY, OE, and DyAD routing algorithms.
- From Table III and Fig.6, it can be seen that the proposed ETD routing algorithm consumes less power than XY, OE, and DyAD for all low, mid, and high network load. Compared to the existing XY, OE and DyAD routings, the proposed ETD routing consumes 5.46%, 32.40%, and 4.96% less power at low network load, 3.10%, 13.28%, and 11.79% less power consumption under mid network load, and 3.62%, 12.62%, and 11.01% less power under high network load, respectively.
- Both Table IV and Fig.7 demonstrate ETD's advantages over the other 3 routing algorithms for all of the low, medium and high traffic loads, especially when compared to the OE and DyAD routing algorithms. Compared to the existing XY, OE and DyAD routings, the proposed ETD routing reduces latency by 2.27%, 29.75%, 2.69% under low network load, by 1.86%, 106.49%, and 67.94% under mid network load, and by 4.81%, 156.73%, and 154.74% under high network load, respectively.
- Table V and Fig.8 both depict that the propose ETD routing yields higher throughput under all of the low, mid and high network loads over the existing routing algorithms, especially when compared to the OE and Dyad routings.

V. CONCLUSION

New deterministic NOC routing algorithm, Edge Traffic Distribution routing algorithm, simulation results and analysis are presented in this paper. The comparison of the four routing algorithms, XY, OE, DyAD and the ETD routings algorithms in term of average total network power consumption and average communication latency in clock cycle per flit is also presented. Test results demonstrate that the new ETD routing algorithm is capable of improving the system performance under different network load levels. We hope to show in the future that many real-world NOC applications can benefit from our proposed ETD routing algorithm design.

REFERENCES

[1] I. Cutress, "Intel Reading 15-core Xeon E7 V2," AnandTech, 2014. http://www.anandtech.com/show/7753/intel-readying-15core-xeon-e7-v2

[2] J.A. Pande, P. Sharma and A.Singh, "Performance Optimization for System-on-chip Using Network-on-chip and Data Compression," IJIRAE Journal, vol 2(1), 2015, pp228-234.

[3] N. Wang, A.Sanusi, P.Y.Zhao, M. Elgamel and M.A. Bayoumi, "PMCNOC: A Pipelining Multi-channel Central Caching Network-on-chip Communication Architecture Design," Springer Journal of Signal Processing, vol.60,2010, pp 315-331.

[4] L. Benini and G.D. Micheli, "Network-on-chip: A New SOC Paradigm," IEEE Trans. Computers, vol 35(1),2002, pp70-78.

[5] P. Parandkar, J.K. Dalal and S. Katival, "Performance Comparison of XY, OE and DyAD Routing Algorithm by Load Variation Analysis of 2-Dimensional Mesh Topology Based Network-on-chip," BIJIT Journal, vol 4(1), 2012, pp391-396.

[6] W. Zhang, L. Hou, J. Wang, S. Geng and W. Wu, "Comparison Research Between XY and Odd-Even Routing Algorithm of a 2-dimension 3X3 Mesh Topology Network-on-chip," IEEE Proceedings of GSIS'09. Nanjing, China. 2009, pp329-333.

[7] N. Wang and P. Valencia, "Traffic Allocation: An Efficient Adaptive Network-on-chip Routing Algorithm Design", IEEE 2nd International Conference on Computer and Communications (ICCC2016), Chengdu, China, October 2016.

[8] G. Ascia, V. Catania, M. Palesi, and D. Patti, "Implementation and Analysis of a New Selection Strategy for Adaptive Routing in Networks-on-Chip," IEEE Trans. Computers, vol. 57(6), 2008, pp 809-820

[9] P. Gratz, B. Grot and S.W.Keckler, "Regional congestion awareness for load balance in networks-on-chip," IEEE Symp. HPCA, 2008, pp 203-214.

[10] R. S. Ramanujam and B. Lin, "Destination-based adaptive routing on 2D mesh networks," IEEE Symp. ANCS, 2010, pp 1-12.

[11] G.M. Chiu, "The odd-even turn model for adaptive Routing," IEEE Trans. Parallel and Distributed Systems, vol 11(7), 2000, pp 729-738.

[12] J. Hu, and R. Marculescu, "DyAD-Smart Routing for Network-on-chip," IEEE proceedings of 41st DAC . San Diego. 2004, pp 260-263.

[13] jGyan,com. NIRGAM Simulator Basics. 2016. http://www.jgyan.com/nirgam/simulator%20basic.php