A LOAD BALANCE AWARE XY ROUTING METHODOLOGY FOR NOC ARCHITECTURES

Siddhartha S[1], Venkateswara Rao M[2], Sai Prathyusha K[3], Rama Krishna T V[4]
Department of ECE, KLUniversity,Vaddeswaram, Guntur, India, 522502.
Siddu.sudineni@gmail.com, venkatvlsi@kluniversity.in, Kopanati.saiprathyusha@gmail.com,
tottempudi@kluniversity.in

Abstract— The routing algorithm performs a critical function inside the overall performance of the community on chip. Dynamic routing is attractive in view of its considerable change in communication bandwidth and keen adjustment to flawed connections and congested traffic. XY-Routing in the mesh topology creates congestion at the central part of the network which increases the latency and leads to decreased performance. Congestion in a network generally increases at the central part of the network due to increased traffic on the same nodes periodically by the neighbor nodes. Furthermore, congestion due to the neighboring nodes will certainly diminish the performance of the system and will have an adverse effect on the nodes. Then we strive to minimize the local latency due to congestion using address distinct nearby place size, based on Divide and Conquer method for routing strain. It minimizes latency in each local vicinity with the aid of decreasing the routing pressure of each local node. In this paper we have implemented the popular mesh topology along with LBAR algorithm and the results are compared with conventional XY-Routing. It is observed that Load balancing and latency is improved in case of LBAR as compared to Normal XY and Total Network Congestion is reduced for LBAR as compared to traditional XY.

Keywords— Dynamic routing, LBAR, Congestion, Divide and Conquer, conventional XY.

1. Introduction

Current System on chip contrast from wide region networks in their neighbourhood proximity and on the grounds that they show less nondeterminism. Neighbourhood, elite networks such as those produced for large scale multi-processors will be having comparative necessities as well as imperatives. A few distinct attributes, for example, energy requirements furthermore, design time specialization, are particular in SoC networks [1] [2]. The effectiveness of interconnection and meeting information exchange prerequisite are more imperative for NoC frameworks, and also network on chip (NoC) have turned out as an adaptable, versatile, and likewise reusable for the issues concerning these kind [3]. In a conventional wired NoC, the interchanges among embedded cores are for the most part by means of numerous switches and wired connections. This multi hop correspondence gets to be a noteworthy bottleneck in framework execution, which offers ascend to high dormancy and vitality dissemination. To defeat this execution restriction we propose a new architectural models roused by complex system hypothesis in conjunction with deliberately set on-chip connections to make high performance and low-latency NoCs [4]. Many center outlines, for example, the Intel Integrated structures are moving towards multicore Xeon Phi[5] [6] , as a way to accomplish productive and higher execution with low latency. In this paper we are proposing a performance model that minimizes the load in network-on-chip based system. Generally, engineers design a performance model, later analyse forthcoming technologies based on performance model. Regarding this point, first architectural model and applications are established individually. Then the applications are developed by the performance models and architectures are used to evaluate the selected application-architecture combination [7] [8]. Meanwhile there exists multiple directions from current node to targeted node, network on chip must carry out steering algorithms to route data packet for target destination. It influence the throughput and latency that experiences the traffic.

A large number of proposals have been made for NoC and are implemented by network topologies and the routing algorithms are utilized for the on-chip communication networks. Routing algorithms can be cascaded into two specific strategies in accordance to the network type which it firmly fits. If the routing path of the data packet is fixed in advance then this sort of routing is known as source routing. A part from that if the path of the data packet is resolved hop by hop then this type of routing is called source routing [5] **[9].** Such that NoC's are usually implemented by dimension ordered routing (DOR) which primarily routes the packets initially in the horizontal direction (X dimension) and later in the vertical direction(Y dimension) towards the receiver [10]. Although this algorithm reduces the latency, they generally perform very low because there is a heavy load amidst the system which causes traffic in whole network [11]. Load balance XY routing algorithm exploits the network topology into different co-ordinates this helps in reducing the traffic in network, thereby it provide efficient performance over oblivious routing i.e., dimension ordered routing (DOR). This algorithm

contains shortest path routing technique, advanced implementation and a light weight.

2. Related Work

A. NoC Overview

Switching approach, topology and routing algorithm are essential element within the layout of network on-chip. The topology represents network interconnection. In mesh topology the connections in system have comparative limits which makes basic physical design also the range becomes direct to the quantity of nodes. Size in the present topology is measured in terms of lines and segments [12]. Constant lengths is desired by so many research companies because of its appropriate electrical properties, layout capability and ease in resource on chip address **[13].** There is preference in work with this topology has got its individual specific manner of steering known as source directing. This element in this topology delivers a proficient encoding of way data with just few bits [12]. While a packet header arrives in between the nodes, the switching process resolves by what means the transfer is set, i.e. the injection channel is attached for the ejection channel [14] [12].

Wormhole routing is broadly utilized for switching approach on account of its less buffer prerequisites and high noteworthy critically, as it generates the packet conveyance time about unbiased of the separation amongst source and destination hubs. In wormhole routing, a packet can be divide into a progression of settled length entities, called flits. The header flit (which contain the routing measurements) sets up a course through the system in the meantime as a definitive body flit pursue it in a pipelined design.

Consecutively to surge the general execution of the system, each channel of the group can be convenient multiplexed between a few supports, in particular virtual channels. By assigning distinct packets to each of these supports, flits of data from different packets might be sent in an interleaved way over each physical channel. This improves both throughput and latency by permitting blocked data packets that are to be neglected [15]. The data path of the data packet includes buffers and also crossbar switch. The steering module and the VC allocator decides the following hop and the following virtual channel, and switch allocator is in charge of figuring out which flits are chosen to navigate into the crossbar switch. At the point when a data packet is blocked by the reason of the fact that there is no accessible support space in the required switch, it will occupy the buffer assets that are on forth its path. Hence, message directing in wormhole switch based systems is inclined to stop [14] [16]. The exchanging point of the node delineated in this figure comprises of 5 channels including the local interrupt to communicate with distinct direction connecting hubs, and the nearby interfere for the local interrupt as shown.

During the course, flit of the header touches base at the information channel of an exchange (including both a nearby nodes or the neighbourhood handling component associated with the indistinguishable switch) the switch decides packet take-off course e.g., in four different directions or the local interrupt channel i.e. ejection channel, later the switch arranges the crossbar switch through sending relevant data to it. Crossbar exchange associates the flit's approaching channel to the chose active one. Inside the instance of life of a free virtual channel, the header flit may be exchanged for the ensuing hub and the edge flits take after the header flit. Take some other case, the header flit needs to hold up until any of the virtual channel which was being used by other channel is set free to occupy and use the channel.

B. Routing Algorithm

Routing algorithm is an important component that influences effectiveness with correspondence to NOC .The routing calculation characterizes the direction chosen by the data packet starting from source directed towards the destination, is a primary errand in NOC to construct a network layer **[17].** As indicated to locate routing choices are seized, it is conceivable to analyse source and distributed routing. In case of the source routing, entire data path has to be chosen by the source node, although in distributed routing every switch gets the data packet as well as chooses effective path to address data. As indicated by in what way the path will be characterized to send packets, data packet transfer can be analysed by deterministic or else versatile. Likewise it will be characterized depend upon their versatility, the fault-tolerant capacity, concentrated controller governs the flow of data in a framework and bank on their destination quantity. Still generally named versatile, deterministic and oblivious. Deterministic routing algorithm designate similar path between a couples of hobs, load balance is exceptionally impoverished for already stated situation, and however they are ordinarily utilized because of simple usage. In Oblivious routing algorithm packet course left out considering system's state. The Adaptive routing algorithm utilize data regarding the system's state (e.g. limit of lines for assets, and so on.) to compose routing decisions. A fully designed router has to give versatile routing directed towards program a data packet over the less congested channel. In adaptive routing, every switch has clog data of its encompassing neighbourhood. The channel clog metric can be founded on free virtual channels quantity, the demand for router output, the number of free buffers are blend of these parameters [17] [21]. In regards of congestion data, the switch programs the data packet for its goals with less congested channels [18]. A few analysis are conceivable like Fault Tolerance routing Algorithm in which powerfully distinguishes the broken segments at the time of routing data packets. What's more, Routing

through the Reconfiguration without revolution form throughout the broken connection for utilizing the advanced one of a kind path rather than the broken path. The flow chart of the algorithm is shown in the fig.
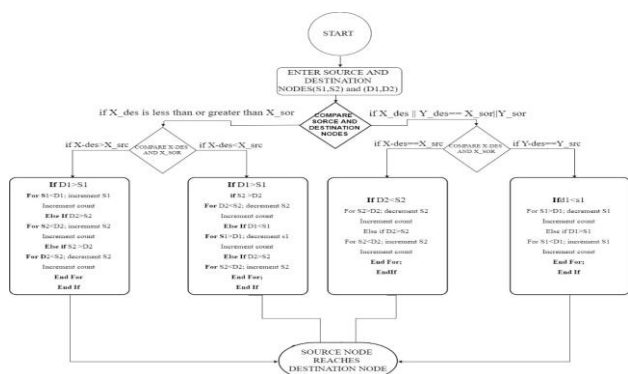

Figure 1.

C. Methodology

The majority of the NoC node architecture are utilized in a 2D mesh-based topology. In general effective addresses of resources and routers can be easily defined in mesh topology. Every node in this topology has a location in the form of X and Y Coordinates [20] [17] [19]. X- Represents its position in the X dimension which in horizontal direction, y-represents its position in the Y dimension which in vertical direction as shown in fig 1. For implementation of conventional algorithm source address (S_x,S_y) is contrast and the destination address (D_x, D_y) of the packet depends on the correlation output of the routing calculation switch courses the packets on the off chance that $(D_x < S_x)$ head flit turns to west else it take east direction up to (D_x, S_x) get to be distinctly equivalent this passion is known as horizontal alignment. Now (D_y, S_y) experiences same method, but in this event that it is found that $(D_y > S_y)$ then packet's header flit moves north else south up to $(D_y= =S_y)$ [17] [19]. The conventional routing is a dimensional order routing algorithm, with free of deadlock, live lock algorithm and less reliability of data transfer. Although it has some drawbacks like traffic is not distributed over the network, more load is distributed at the center of the network, No reliability in case of node breakage and more latency. Our proposed algorithm is known as Load balance XY routing. In consideration of subsidiary the conventional routing, later it will stay deadlock free. Our routing algorithm which will be working as a deterministic or versatile routing relies on upon system stack position Packets are directed with conventional routing until there is less traffic in network. At the point when clog turns out to be high, our change tries to route packets through less congested way.
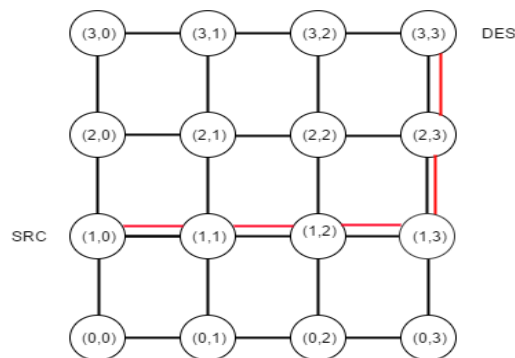

Figure 2.

The basic thought for LBR algorithm is to diminish the load on the entire network by utilizing address distinct nearby place estimate, based on Divide-Conquer strategy and routing strain. It maintains a strategic distance from congestion in each local region with the guide of continuing routing weight to minimize every local area as shown in figure[ ] according to Gratz *et* al local congestion aware directing [17][18] which abstracts the local traffic level into four single numbers to figure out whichever path will probably have low congestion. LBR is improvised version of the XY-routing, LBR algorithm also has improved latency when compared to traditional XY that is to calculate total time has been taken by the data packet to enter into the node and come out of the node as an output and bound to transfer data in case of plane mesh. To reduce congestion over the network based on some combinational parameters like quantity of free virtual channels, the number of free buffers, demand for router output. To implement proposed algorithm we consider current nodes as (s1, s2) and target nodes as (d1, d2) of the data packet. when X , Y destination nodes is greater than X, Y source nodes (d1,d2>s1,s2) header flit moves to east direction or else move to south direction until source and destination are to be equal or otherwise X coordinates are equal (d1= =s1) then they are two condition to check that is Y destination is less than Y source node(d2<s2) take south direction or else north direction till Y coordinates are to equal or otherwise if Y coordinates are equivalent (d2= =s2) then two conditions are there to check that is X destination node is less than X source node (d1<s1) then it move towards west or else data packet take east direction this algorithm is improved based on traditional XY routing algorithm but when congestion become high, it divide into four coordinates to distribute congestion over network and it counts how many nodes it crosses referring to the number of nodes involved in path and overall compilation time has calculated. The latency is minimum in load balancing routing algorithm, reliability is less in case of blockages compared to XY routing. Experimental trails shows average latency in network along with proposed routing with load balance. Load balance routing algorithm not only distribute the load over network but also minimize

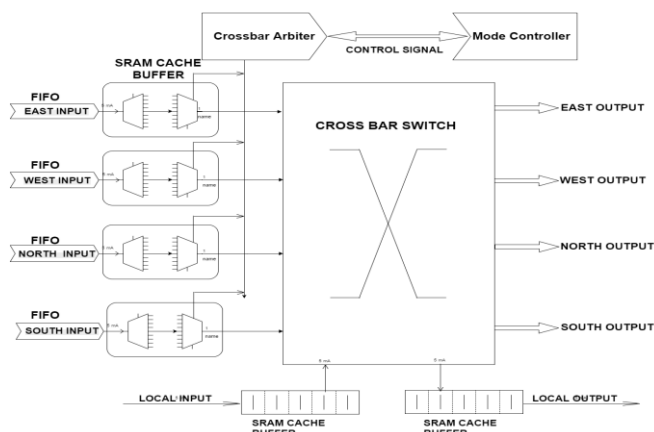the latency of data packet. The architecture and mathematical model for the proposed system is shown in fig: 3

Figure 3. NoC architecture

Proposed Mathematical solution for overall latency in Architecture

- Time consumed in the processing of the data is calculated using the mathematical equation as mentioned below:

- L=

$$T_{is} + \sum_{i=1}^{n}(T_{in} + T_{ac} + T_{e_{cs}} + T_{out} + T_{int\_loc}(Tiexe) + T_{fout})$$

  Execution time latency is termed as:

- $E_l = T_{ac} + T_{e\_cs} + \prod_{j=1}^{n}(T_{j,int_{loc}}(L_{iexe}))$

- Where, **L**=total latency

    T_{is}=initial waiting time
    T_{in}=time taken to inject into node
    T_{ac}=time taken by arbiter and controller
    $T_{(e\_cs)}$ =execution time at crossbar switch
    T_{out}=time taken to leave node
    $T_{(int\_loc)}$ =time taken by local interrupt
    T_{iexe}=interrupt execution time
    T_{fout}=final out time

  Processing time also depends on the type of topology used , So based upon the merits and demerits best suitable topology is selected like the Mesh or Torus. The bandwidth problem can be reduced by using the high speed transmission.

- Bandwidth is measured as $L = \frac{Ds}{B}$

    Ds=Data packet size
    B=speed of transmission.

- Deadlock is avoided by using efficient algorithms like the Round robin method etc.

3. Simulator Model

The simulations are done using the online gcc compiler to calculate average delay for different network sizes like 4x4, 8x8 and 16x16 in mesh topology. The above mentioned network sizes are simulated using conventional XY Routing and proposed Load balance aware XY routing in this paper. Average time delay of the respective networks are taken into consideration with different nodes. A C++ based cycle accurate on chip network simulator is used to evaluate the performance of the network size 16x16 is chosen with different PIR rates under the subsequent parameters. The simulation is executed for buffer depth 4 and flit size 64 bit with DYAD threshold of 0.6. Each simulation is first run for 1000 cycles and then 11000 cycles were executed for each simulation with Flit transmission delay of 1000ps/cycle.

4. Results

1. Performance evaluation of LBAR over XY-Routing

The results obtained by executing the XY routing algorithm the central part of the network has more amount of congestion and load, which drastically decreases the overall performance of the system. The load in the network is represented with dark colour. Greater the load on the node, darker the colour is. Distribution of the load throughout the network will result in much better results in terms of latency and congestion. The congestion on the nodes are represented by darkening the node areas in the figures [4] [5] [6].Figures 4, 5, 6 depict the load and its distribution on the network.
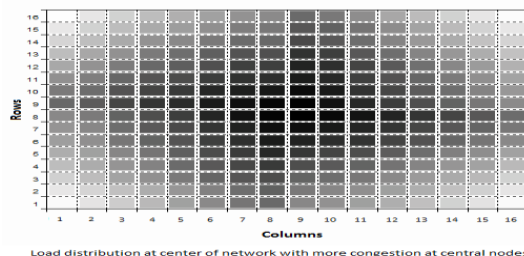

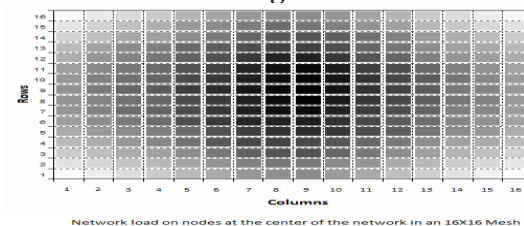Load distribution at center of network with more congestion at central nodes.

Figure 4.

Network load on nodes at the center of the network in an 16X16 Mesh

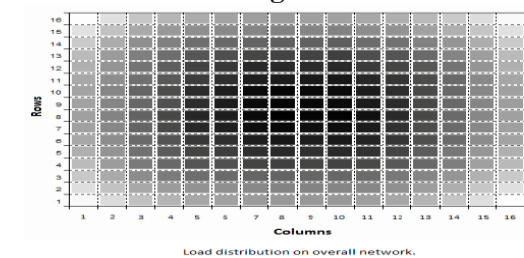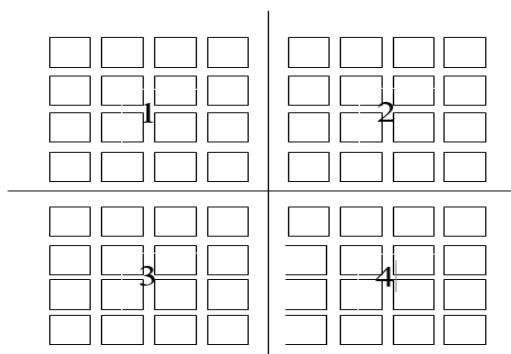**Figure 5.**

Load distribution on overall network.

Figure 6.

DC (Divide and Conquer) methodology was proposed to minimize the overburden on the nodes and also minimizes the latency of the network without any drawbacks like deadlock and livelock provided without any node breakages. We have proposed the methodology to improve the latency and the congestion by dividing the whole network into four equal quadrants and routing the data packets using the XY algorithm i.e. routing the data packet first in the horizontal direction and later in the vertical direction as in figure.

Partitioning of network into 4 quadrants.

Figure 7.

Table 1: Latency analysis of 4x4 mesh

S.No	Communication between nodes	Avg.time delay of XY	Avg.time delay of LBAR	Minimized percentage
1.	0 - 8	0.5385	0.3269	39.29%
2.	3 - 10	0.5389	0.3272	39.28%
3.	7 - 12	0.5250	0.3129	40.4%
4.	4 - 15	0.5378	0.3261	39.36%

Every quadrant is made free from deadlocks and livelocks. The same XY routing algorithm is used in the LBAR with small dissimilarities in execution of the simulated network. Simulation results obtained by executing 4x4, 8x8 networks for both XY and LBAR are compared and found that Avg.Time delay is reduced by 40% using gcc compiler in figure[]. 16x16 2D mesh topology is executed in the noxim simulator and results depict that the Avg.time delay has been reduced to 25% on an average Delay.

Figure 8.
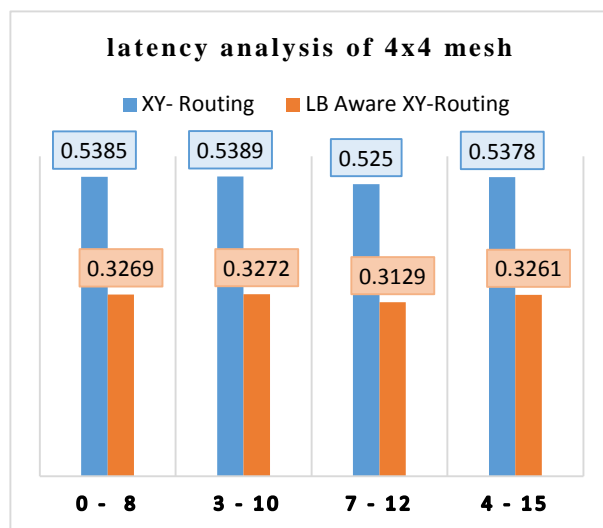
Table 2: Latency analysis of 8x8 mesh

S.no	Communication between nodes in 8X8 2D Mesh	Avg.time delay of XY	Avg.time delay of LBAR	Minimized percentage
1.	0 - 23	0.5981	0.3483	41.76%
2.	13 - 40	0.6348	0.3527	44.43%
3.	21 - 53	0.6537	0.3541	45.83%
4.	25 - 61	0.6983	0.3619	48.17%

Figure 9.

Table 3: Latency analysis of 16x16 mesh (for packet injection rate 0.01).

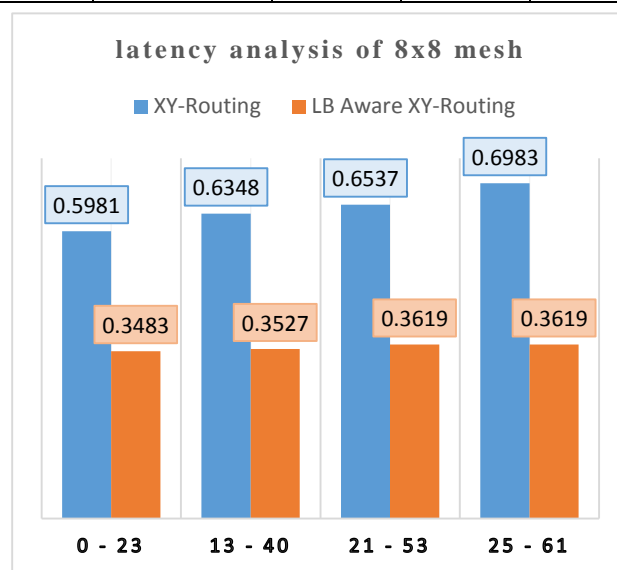| S.no	Node addresses	Avg.time delay of XY	Avg.time delay of LBAR	Minimized percentage
1.	10 – 99	3828	3013.5	21.27%
2.	91 – 193	3196	2855.5	10.65%
3.	167 – 221	5522	4330.5	21.57%
4.	120 – 252	3919	3231.5	17.54%

Figure 10.

Table 4: Latency analysis of 16x16 mesh (for packet injection rate 0.02)

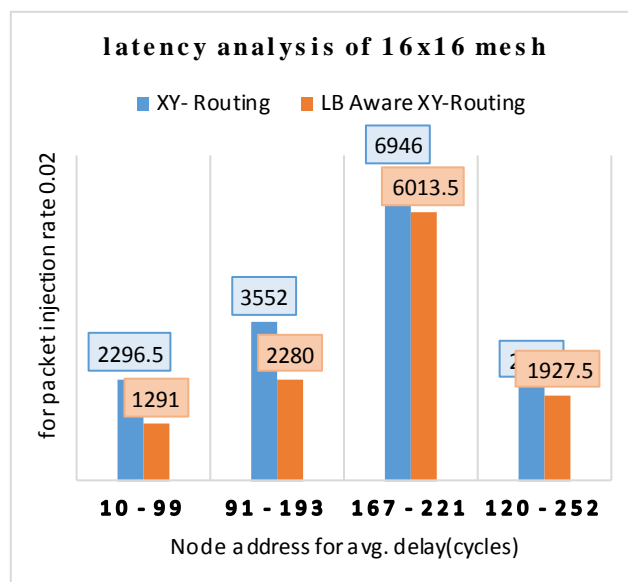| S.no	Node addresses	Avg.time delay of XY	Avg.time delay of LBAR	Minimized Percentage
1.	10 - 99	2296.5	1291	43.78%
2.	91 - 193	3552	2280	35.81%
3.	167 - 221	6946	6013.5	13.42%
4.	120 - 252	2147	1927.5	10.22%

Figure 11.

For example when latency of the delay with PIR 0.02 is considered for different nodes and the same are examined for both the methods. When the delay values for nodes 91 to 193 are taken the delay values are 3552ms and 2280ms for Traditional XY and LBAR respectively reducing the delay up to 35.81%. And the delay values are noted down for analysis. On an average the overall delay of the network has been reduced to 25% as mentioned. Also the Load at the central part of the network is distributed for the neighbouring nodes and network, which is also a main criteria in increasing the performance and throughput of the system. The load of the network is indicated by darkening the more congested area in the network. Increase in the traffic on the node increases the darkness of the network. Simulation results show that the load on the network is decreased and spread across the network.

Table 5: Latency analysis of 16x16 mesh (for packet injection rate 0.03)

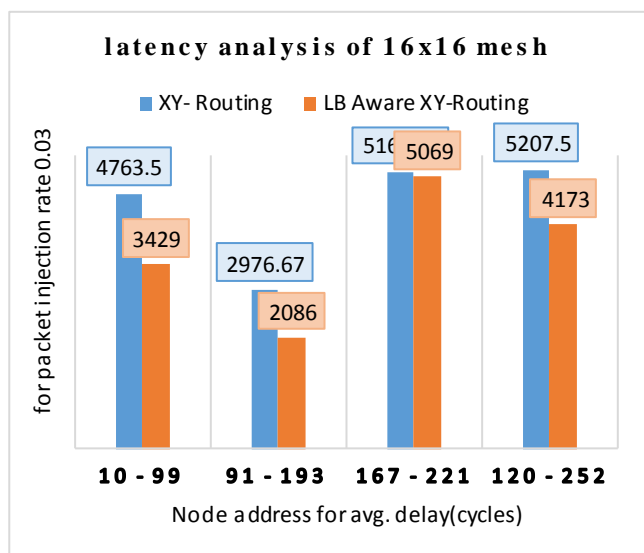| S.no	Node addresses	Avg.time delay of XY	Avg.time delay of LBAR	Minimized Percentage
1.	10 – 99	4763.5	3429	28.01%
2.	91 – 193	2976.67	2086	29.92%
3.	167 – 221	5164.64	5069	1.85%
4.	120 – 252	5207.5	4173	19.86%

Figure 12.

Table 6: Latency analysis of 16x16 mesh (for packet injection rate 0.04)

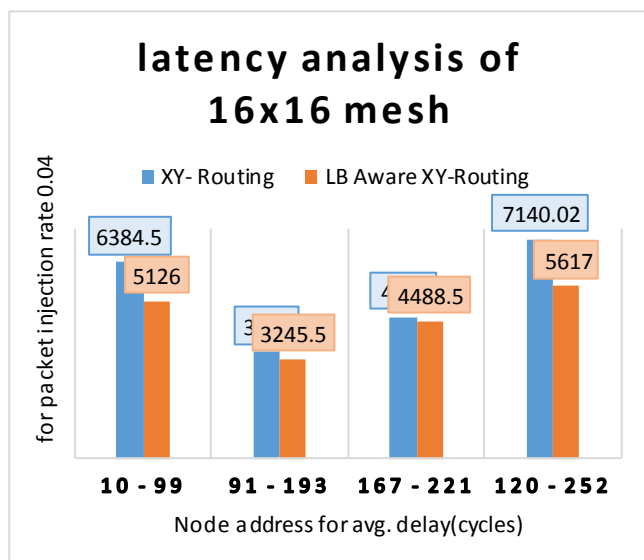| S.no	Node addresses	Avg.time delay of XY	Avg.time delay of LBAR	Minimized Percentage
1.	10 – 99	6384.5	5126	19.71%
2.	91 – 193	3470	3245.5	6.46%
3.	167 – 221	4607	4488.5	2.57%
4.	120 – 252	7140.02	5617	21.33%

Figure 13.

Table 7: Latency analysis of 16x16 mesh (for packet injection rate 0.05)

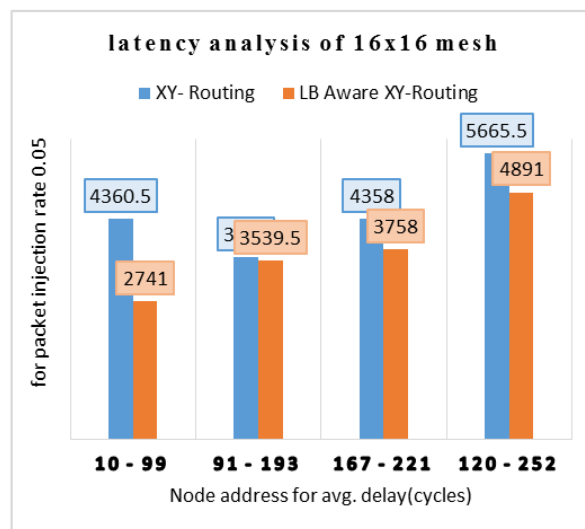| S.no	Node addresses	XY-Routing	LB Aware XY-Routing	Minimum percentage
1.	10 - 99	4360.5	2741	37.14%
2.	91 - 193	3610	3539.5	1.95%
3.	167 - 221	4358	3758	13.76%
4.	120 - 252	5665.5	4891	13.67%

Figure 14.

5. Conclusion

In this paper we examined the conventional XY and LBAR algorithm. We have estimated the trade-offs such as Avg.Delay, network size and load on the entire network of the NoC architecture. Our simulations showed us that XY routing has more load at center of the network. Also we observed that the delay and performance of the network is decreased in case of the traditional XY routing algorithm. The proposed LBAR algorithm shows good results in terms of Avg.Delay and overall performance of the system by using the divide and conquer method. The load at the center of the network is decreased by dividing the network into quadrants and distributing the bandwidth among the nodes.

Our future work includes the implementation of the algorithms using blockages in the nodes by implementing PAR (Path Aware Routing Algorithm) which eradicates the deadlocks and livelocks in case of node breakages. It shows a great amount of difference in terms of latency and load on the entire system.

References

[1]   Luca Benini University of Bologna, Giovanni De Micheli Stanford University, "Networks on Chips: A New SoC Paradigm" in 2002 IEEE.

[2]   Masood Dehyadgari, Mohsen Nickray, Ali Afzali-kusha, Zainalabein Navabi, "Evaluation of Pseudo

Adaptive XY Routing Using an Object Oriented Model for NOC" in 2005 IEEE.

[3] Jili Yan, "Enhanced global congestion awareness (EGCA) for load balance in networks-on-chip" in springer Science+Business Media New York 2015.

[4] kevin chang and sujay deb, amlan ganguly, xinmin yu, suman prasad sah, partha pratim pande, benjamin belzer, and deukhyoun heo, "Performance Evaluation and Design Trade-Offs for Wireless Network-on-Chip Architectures" in ACM Journal on Emerging Technologies in Computing Systems, Vol. 8, No. 3, Article 23, Pub. date: August 2012.

[5] Mukund Ramakrishna, Vamsi Krishna Kodati, Paul V. Gratz, Member, IEEE, and Alexander Sprintson, "GCA:Global Congestion Awareness for Load Balance in Networks-on-Chip", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 27, NO. 7, JULY 2016.

[6] Alexander Heinecke, Karthikeyan Vaidyanathan, Mikhail Smelyanskiy, Alexander Kobotov, Roman Dubtsov, Greg Henry, Aniruddha G Shet , George Chrysos, Pradeep Dubey " Design and Implementation of the Linpack Benchmark for Single and Multi-Node Systems Based on IntelR Xeon PhiTM Coprocessor" in 2013 IEEE 27th International Symposium on Parallel & Distributed Processing.

[7] Abbas Eslami Kiasari, Zhonghai Lu and Axel Jantsch, " An Analytical Latency Model for Networks-on-Chip" in January 2013 IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 21, NO. 1, JA.

[8] John D. Owens University of California, Davis William J. Dally Stanford University, Ron Ho Sun Microsystems, D.N. (Jay) Jayasimha Intel Corporation, Stephen W. Keckler University of Texas at Austin, Li-Shiuan Peh Princeton University "RESEARCH CHALLENGES FOR ON-CHIP INTERCONNECTION NETWORKS" in 2007 Published by the IEEE Computer Society.

[9] Yongfeng Xu, Jianyang Zhou, Shunkui Liu Department of electronic engineering, "Research and Analysis of Routing Algorithms for NoC", in 2011 IEEE.

[10] Jose Miguel Montañana, Michihiro Koibuchi, Hiroki Matsutani, Hideharu Amano, "Balanced Dimension-Order Routing for k-ary n-cubes" in 2009 International Conference on Parallel Processing Workshops.

[11] Garba Adamu, Mr. Pankaj Chejara, Dr. Ahmed Baita Garko, "REVIEW OF DETERMINISTIC ROUTING ALGORITHM FOR NETWORK-ON-CHIP", in September 2015 International Journal Of Advance Research In Science And Engineering Vol.No.4, Special Issue (01).

[12] M. Venkateswara Rao, T. V. Rama Krishna, S. Raaga Sai Sruthi, S. Akhila, Y. Gopi and L. Bhavani Krishna, "An Effective on-Chip Network Topology for

Network on Chip (Noc) Trade-Offs" in Indian Journal of Science and Technology, Vol 9(17), May 2016.

[13] Ahmad Patooghy, Seyed Ghassem Miremadi Department of Computer Engineering, Sharif University of Technology, Tehran, Iran, "Microprocessors and Microsystems" in 2010 Elsevier B.V. All rights reserved.

[14] William J. Dally and Brian Towles Computer Systems Laboratory Stanford University Stanford, CA 94305, "Route Packets, Not Wires: On-Chip Interconnection Networks" in International Conference on Design Automation, pages 684–689, 2001.

[15] Robert Mullins, Andrew West and Simon Moore, "Low-Latency Virtual-Channel Routers for On-Chip Networks" Proceedings of the 31st Annual International Symposium on Computer Architecture (ISCA'04) 1063-6897/04 $ 20.00 © 2004 IEEE.

[16] Pengju Ren, Member, IEEE, Xiaowei Ren, Sudhanshu Sane, Michel Kinsy Member, IEEE and Nanning Zheng, Fellow, IEEE, "A Deadlock-Free and Connectivity-Guaranteed Methodology for Achieving Fault-tolerance in On-chip Networks" in , IEEE Transactions on Computers.

[17] Shubhangi D Chawade, Mahendra A Gaikwad, Rajendra M Patrikar, "Review of XY Routing Algorithm for Network-on-Chip Architecture" in International Journal of Computer Applications (0975 – 8887) Volume 43– No.21, April 2012.

[18] Mohsen Nickray, Masood Dehyadgari, Ali Afzali-kusha, "Adaptive Routing Using Context-Aware Agents for Networks on Chips" in 2009 IEEE.

[19] Wang Zhang, Ligang Hou, Jinhui Wang, Shuqin Geng, Wuchen Wu, "Comparison Research between XY and Odd-Even Routing Algorithm of a 2-Dimension 3X3 Mesh Topology Network-on-Chip" in 2009 IEEE computer society.

[20] Lalit Kishore Arora, Raj Kumar, "Alternatives of XY-Routing for Mesh" in Special Issue of International Journal of Computer Applications (0975 – 8887) on Issues and Challenges in Networking, Intelligence and Computing Technologies – ICNICT 2012, November 2012.

[21] Paul Gratz, Boris Grot, Stephen W. Keckler, "Regional Congestion Awareness for Load Balance in Networks-on-Chip" in 2008 IEEE.

[22] T. Padmapriya and V. Saminadan, "Priority based fair resource allocation and Admission Control Technique for Multi-user Multi-class downlink Traffic in LTE-Advanced Networks", International Journal of Advanced Research, vol.5, no.1, pp.1633-1641, January 2017.