PARTIALLY ADAPTIVE DEADLOCK-FREE ROUTING ALGORITHM FOR RECTANGULAR 2-D MESH NETWORK

Kamalesh Chandra Maurya[1] and Dr. Pankaj Kumar[2]

[1]Research scholar, Dr. A.P.J. Abdul Kalam Technical University, Lucknow, kamalesh1947@gmail.coom

[2] Associate Professor, Department of Computer Science & Engineering Shri Ramswaroop Memorial College of Engineering & Management, Lucknow, pk79jan@gmail.com

ABSTRACT:

Freedom from deadlock is one of key researches for interconnected processor network. Its significance and impact on the network's performance is appropriately cardinal. Deadlock-free, fault-tolerant, low latency and load-balance are the alluring properties of a routing algorithm for interconnected processor network. The current research on interconnection networks of multiprocessor mainly consider on wormhole switching, virtual channel flow control and routing algorithms. This structural characteristic tries to increase system efficiency by reducing network latency, which enhances network performance. In this paper, we proposed a simple, efficient and deadlock-free routing algorithm for interconnected 2-dimensionl mesh network. In our proposed method deterministic and adaptive both routing algorithm applied simultaneously in any interconnected network. The selection of deterministic routing depends on the dimension of the network. In comparison to prior algorithms, this novel routing approach makes it possible to create networks with low latency and higher performance. Our technique may be applicable to any network topology and routing methods. Simulation is carried out on different size of network and results show low network latency.

Keywords: *Deadlock-free, wormhole switching, multiprocessor, network latency, routing algorithm.*

I. INTRODUCTION

The communication latency between a source and a destination node inside the interconnected processor network determines the performance of a multiprocessor system. Minimizing latency between such pairs of communication nodes can improve the performance of interconnection networks.

This objective can be achieved by maximizing the utilization of network bandwidth and minimizing the latency in transmission of massage between any two nodes. Wormhole switching is a method used by modern multiprocessor systems to reduce network latency [16, 20]. A message is transferred from source to destination in the network using the wormhole switching approach by splitting each message into packets, which is further divided into fixed-size data units called flits. There is a special flit in each packet called header flit which contains the routing information and the remaining flits contain massage part called data flits and follow the header flit through the network to reach destination.

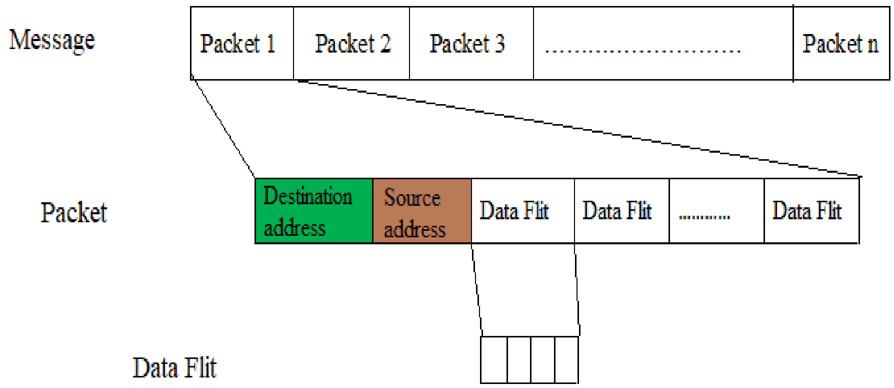

Figure 1: Message format

To increase the performance of multiprocessors systems, wormhole switching technique is used in the modern multiprocessors, because message flits are forwarded as quickly as feasible. The delay in massage delivery is almost independent to the position of the source node and destination node in the absence of link contention [20]. Wormhole switching also has the advantage of low memory requirements in intermediate nodes through which message are routed. The main disadvantage of wormhole switching is it suffers from deadlock situations.

The main advantages of wormhole switching include:

1. **Low Memory Requirements:** Wormhole switching requires less memory in routers compared to other switching techniques. This is because wormhole switching only requires a small buffer to hold a portion of a message (typically a few flits or flow control units) before it can be forwarded.
2. **Pipelined Data Movement:** In wormhole switching, data can be pipelined or sent in small units called flits (flow control units). This allows for faster data movement through the network, as flits can be forwarded as soon as they are available, without waiting for the entire message to arrive.
3. **Absence of Contention:** Wormhole switching can handle data movement in the absence of contention, meaning that multiple messages can be sent simultaneously without causing delays or blocking.

However, there is a main disadvantage associated with wormhole switching:

1. **Channel Congestion:** In wormhole switching, if a message gets blocked or encounters congestion along its path, It does not give up the communication channels it has already acquired. This can lead to channel congestion and potentially impact the performance of other messages in the network.
2. **Deadlock:** Deadlock occurred in a system when there exist a cycle of blocked messages in the network and every blocked message is waiting for a communication link belonging to the next blocked message in the cycle. When the deadlock occurs the messages are not

progresses to their destination, causing a system-wide halt and rendering the network non-functional. In a wormhole-switched network, the routing algorithm needs to be free of deadlock in order to guarantee that every message is eventually delivered. Therefore, it is crucial to design routing algorithms that avoid deadlock and ensure the progress of all messages in the network.

Wormhole routing is used by several modern generation multiprocessor computers, such as the Cray T3D, T3E, Intel Touchstone, and MIT J-machine [24, 25, 26, 27], where quick switching is an important issue in the designing of interconnected network.

II. PREVIOUS WORK

The necessary and sufficient conditions for deadlock-free routing in an interconnection network were presented by Dally and Seitz. They simplified the design of deadlock free routing algorithms for wormhole routing by proving that channel dependency graph [20] which is acyclic ensured the absence of deadlock.

Glass and Ni both suggested a technique for creating deadlock-free routing algorithms. This technique requires the use of channel dependency graph. Glass and Ni were proposed a technique for analyzing routing algorithms based on the allowed and denied dependencies [18] between channels. Boura and Das [17] propose a method for establishing deadlocking free network by splitting the channels into two acyclic sets and forcing messages to transit entirely through the one set before utilizing communication links from the second set. J. Duato suggested a criterion that is both necessary and sufficient for adaptive deadlock free routing in 2-dimensional mesh networks [15]. In his paper titled "A Necessary and Sufficient Condition for Deadlock-Free Adaptive Routing in Wormhole Networks," Duato presented a comprehensive analysis of routing in wormhole-switched networks which are deadlock-free. He introduced a condition called "turn model" and proved that it is both necessary and sufficient for deadlock avoidance in adaptive routing algorithms.

Kawano develop a new HiRy-based routing technique that increases the number of permissible route and hence improves network performance [21]. A heuristic method is developed to support all source-and-destination pairs and reduce the distance between them. G. M. Chiu introduced the odd-even turn model and a deadlock-free adaptive routing algorithm for mesh networks [13]. The odd-even turn model is a routing technique that uses a combination of odd and even turns to avoid deadlocks in 2D mesh networks. It guarantees deadlock freedom by ensuring that packets always move in a direction that eliminates potential cyclic dependencies between channels. Canwen Xiao construct a new theory of deadlock-free flow control for k-ary, n-cube mesh networks that allows the adoption of any minimal-path adaptive routing algorithms while avoiding deadlock[5].

Jindun Dai [3] presented a novel deadlock-free adaptive routing algorithm for 3-dimensional mesh Network on Chip (NoC) interconnections. The routing method of XY routing and YX routing are relaxed and used for intra-layer routing. Freek Verbeek proposed a technique that automatically demonstrates deadlock-free routing functions or generates a minimum counterexample describing the cause of the deadlock [6]. Yuan Cai [1] proposed a new 3-dimensional Network-on-Chip routing algorithms aim to efficiently utilize the vertical connections in a vertically stacked 2-dimensional mesh topology to improve communication performance and reduce congestion. These algorithms take advantage of the additional routing paths provided by the vertical connections. Wei Luo [8] discussed one commonly used adaptive deadlock-free routing algorithm for torus networks is the Dimension-Ordered Routing (DOR) algorithm. DOR ensures deadlock avoidance by carefully selecting the next hop for each packet based on the current and destination coordinates in the torus network.

III. MATHEMATICAL BASIS FOR THE DESIGN OF A DEADLOCK-FREE ROUTING SYSTEM

Designing routing algorithms that avoid deadlocks for arbitrary topologies is a complex task, but there are several theoretical approaches that can be utilized. One popular idea is based on the concept of network flow, which entails modeling the routing issue as a flow network and using algorithms to assure deadlock freedom. Here's a theoretical framework for designing deadlock-free routing for arbitrary topologies:

1. Deadlock free routing algorithm:

A deadlock-free routing algorithm is a routing algorithm used in computer networks to ensure that deadlocks do not occur during the transmission of data packets. Deadlocks exist in multiprocessor system when more than one data packets are waiting for each other to give up communication links, causing a circular dependency and effectively halting the network.

There are several deadlock-free routing algorithms that have been developed to prevent deadlocks in different network architectures. Here we discuss two of the main deadlock-free routing algorithm:

a) **DOR (Dimension-Order Routing)** is a deadlock-free routing algorithm commonly used in parallel computer systems, such as multiprocessor systems or computer clusters. It is designed to efficiently route messages between nodes in a network while avoiding deadlocks.

In DOR routing, the network is organized as a multi-dimensional mesh or torus, where each node is connected to its neighboring nodes in each dimension. The dimensions can be thought of as the coordinates of the network nodes.

The basic idea behind DOR routing is to route messages in a deterministic and ordered manner by following a specific dimension order. Messages are routed along each dimension one at a time until they reach their destination. The order in which the dimensions are traversed is predetermined and consistent across all nodes in the network. **XY routing** is the best known example of DOR routing.

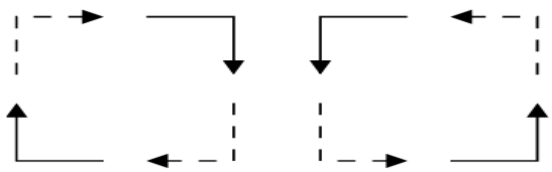

Figure 2: Turn allowed in XY routing

XY Routing Algorithm:
The XY routing algorithm is a type of static routing strategy. This routing technique is applicable to both regular and irregular mesh topologies. It adheres to the principle of minimum turns routing. In this routing technique, each node or router is expressed by its (x, y) co-ordinates in the 2-dimensional mesh network. In XY routing algorithm, the data packet will traverse first in the X-direction after that in the Y-direction. This technique prevents packets from moving in the Y direction before moving in the X direction. So there are certain constraints on its route or turning. As a result, there are no deadlocks. XY routing algorithm will be efficiently worked when the number of rows is less or equal to the number of columns in the mesh network.

b) **Odd-Even Routing algorithm:** The odd-even routing algorithm is a type of adaptive routing technique. It depends on the odd-even turn model strategy. It imposes different route constraints or turning limits; as a result, the interconnection network is deadlock free. Every node in the 2-dimensionl mesh network is expressed by its (x, y) coordinate. The turn model says that a column is considered an odd column if its x dimension element is odd, and that a column is considered an even column if its x co-ordinate is an even number. A change in the direction of moving packet by 90^{o} is called as a turn. There are eight types of turns possible in a 2-dimensional mesh based on travel direction. An NW turn is one that has a 90^{o} change in direction from north to west. An EN turn can be defined as a turn that has a 90^{o} change in direction from east to north. In the same way, we may define additional six types of turns as WS, WN, SE, SW, NE, and ES turns, where N, S, E and W stand for North, South, East, and West respectively.
The following theorems are required by the OE routing algorithm to avoid deadlocks.

Theorem1: If packets available on odd column it cannot take SW turn and NW turn.

Theorem2: If packets available on even column it cannot take ES turn and EN turn.

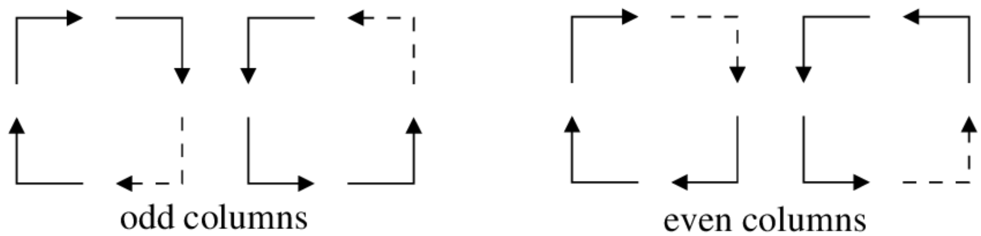

Figure 3: Turn allowed in OE routing

It's important to note that the specific implementation details of the routing algorithm may vary depending on the dimension of network and the characteristics of the network topology. However, following this theoretical framework can provide a solid foundation for designing deadlock-free routing in arbitrary topologies.

2. **Network Topology:** The network topology refers to the interconnection scheme used to connect multiple processors or processing units within the system. The choice of network topology for a multiprocessor system plays a crucial role in determining the communication efficiency, scalability, fault tolerance, and overall performance of the system. Mesh topology are most commonly used topology for interconnection network. In a mesh topology, each processor is connected directly to its neighboring processors, forming a grid-like structure. Data is routed through multiple links and intermediate processors to reach the destination. Mesh topologies can be two-dimensional or three-dimensional, depending on the arrangement of processors. They offer high scalability and fault tolerance but may require a large number of connections, resulting in increased wiring complexity. Our definition of a rectangular mesh topology is identical to the full mesh, but the shape of network is not necessary squire (the number of nodes in row are not equal to number of nodes in column) (Figure 4).

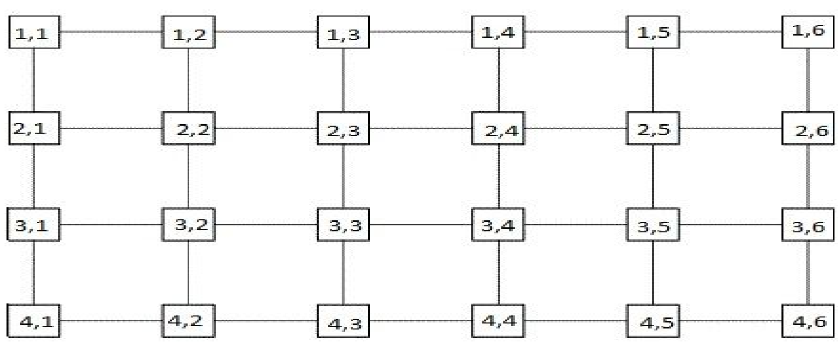

Figure 4: Rectangular mesh network

For choosing any pair of source and destination node:

The probability of flowing packet in row (x-direction) $P(x) = \frac{Nx-1}{Nx+Ny-2}$

The probability of flowing packet in column (y-direction) $P(y) = \frac{Ny-1}{Nx+Ny-2}$

Where N_x is number of nodes in a row and N_y is number of nodes in a column.

A simple XY routing algorithm cannot be employed efficiently in rectangular shape mesh network as the number of nodes in a row is not equal to the number nodes in column. In XY routing algorithm if $N_x > N_y \Rightarrow P(x)>P(y)$, initially most of the packets flowing in x-direction after that packets flow in y-direction. Initially there are no congestion occurred, because all the packets flow in X^+ or X^- direction (may be in pipeline fashion) after that packet may turn in y direction. If their exist congestion on any node, packet routed through OE routing, in which there are many columns (as $N_x > N_y$) and in OE routing there are three possible turns allowed to route the packet to the next node. Therefore there are more paths available to route the packet to their destination. In such case the congestion can be minimized and latency will be reduced. Similarly in YX routing algorithm if $N_y > N_x$ initially most of the packets flowing in y-direction after that packets flow in x-direction. When congestion occurred, then OE routing algorithm is used to route the packets to destination node. In such a way the network latency can be minimized and throughput can be maximized in a rectangular 2-D mesh network.

IV. PROPOSED ROUTING ALGORITHM

In the proposed routing technique we use the advantages of both the adaptive and non-adaptive routing algorithms. As non-adaptive (deterministic) routing algorithms have minimum communication delay in less congested networks and simple router design. The adaptive routing method provides the benefits of avoiding traffic in the network and having better throughput. At lower levels of congestion, the performance of adaptive routing algorithm is inadequate. Because of these extra logic tasks, it will have a large latency overhead. Based on current network congestion, this routing method transforms from adaptive to non-adaptive and vice versa. In this routing technique, each router in the NoC regularly checks the network situation. Traffic levels are computed for all neighboring nodes and verdict are taken on the basis of real-time data. When deadlock-free adaptive and non-adaptive routing modes are used in the same NoC concurrently the deadlock never occurs in system.

Proposed Routing Algorithm:

```
BEGIN
        If (Congestion = < Threshold_value) then
                If( Number_of_row > Number_of _column) then
                        Non-adaptive routing algorithm (XY Routing)
                else
                        Non-adaptive routing algorithm (YX Routing)
        else
```

```
                    Adaptive routing algorithm (OE Routing)
        END
```

XY Routing Algorithm:

/* Source node: (X_s, Y_s), Destination node: (X_d, Y_d), Current node: (X_c, Y_c), $\delta_x = X_d - X_c$, $\delta_y = Y_d - Y_c$ */

```
BEGIN
        if (δx < 0 ) then
                Send packet to WEST port;
        else
                if (δx > 0) then
                        Send packet to EAST port;
                else
                if (δx = 0) then
                {
                        if (δy > 0) then
                                Send packet to NORTH port;
                        else
                        if (δy <  0) then
                                Send packet to SOUTH port;
                        else
                        if (δy = 0) then
                                Current node is the Destination node;
                }
END
```

YX Routing Algorithm:

/* *YX routing algorithm similar to XY routing algorithm, while in YX routing algorithm the data will forwarded in Y-direction first and then in X-direction. This algorithm will be chooses in that condition in which the number of row is more than number of column in the mesh network.* */

/* Source node: (X_s, Y_s), Destination node: (X_d, Y_d), Current node: (X_c, Y_c), $\delta_x = X_d - X_c$, $\delta_y = Y_d - Y_c$ */

```
BEGIN
        if (δy < 0 ) then
                Send packet to SOUTH port;
        else
                if (δy > 0) then
                        Send packet to NORTH port;
                else
                if (δy = 0) then
                {
                        if (δx > 0) then
```

```
                    Send packet to EAST port;
                else
                if (δx < 0) then
                    Send packet to WEST port;
                else
                if (δx = 0) then
                    Current node is the Destination node;
            }
END
```

ODD-EVEN Routing Algorithm

/* Source node: (X_s, Y_s), Destination node: (X_d, Y_d), Current node: (X_c, Y_c), $\Delta_x = X_d$-X_c; $\Delta_y = Y_d$-Y_c; avail_paths is an array consisting of all possible routing directions */

BEGIN

1. Set avail_paths is empty.
2. If (Δ_x= 0 and Δ_y = 0), then Current node is the Destination node.
3. If (Δ_x= 0), it indicates packets are in destination column
4. If above condition is satisfied then check for Δ_y and compute either SOUTH or NORTH to be stored in avail_paths.
5. If (Δ_x >0) Check for Δ_y. If Δ_y = 0 then add EAST in avail_paths. Else check for X_c is odd or X_c = X_d
6. If the above condition is satisfied then check Δ_y accordingly add SOUTH or NORTH in avail_paths.
7. If X_d is odd or Δ_x! = 1, then add EAST to avail_paths.
8. If (Δ_x <=0) then add WEST to avail_paths and check for X_c. If X_c is even, then check for Δ_y
9. If Δ_y <0, add SOUTH to avail_paths or add NORTH to avail_paths.

END

V. SIMULATION AND EXPERIMENTAL RESULTS

We are using NIRGAM 2.1 simulator. NIRGAM is an extensible and modular SystemC based simulator (NIRGAM), which let the user plug-in and experiment with different applications and routing algorithms. It allows the user to analyze the performance (Average latency, throughput and total network power) of a NoC design for a user specified application and a user specified routing algorithm. At present, NIRGAM (NoC Interconnect RoutinG and Applications' Modeling) simulator supports mesh, torus, mesh with link failures and irregular topologies with wormhole switching mechanism.

The experimental setup for calculating the performance of various routing algorithms for different dimensions of mesh topology, with each node linked to a traffic generator which produces CBR (Constant Bit Rate) of the value 2 Gbps for every pair of source and destination. We take input channel FIFO buffer depth (number of buffers) is 8 and each physical channel has four virtual channels. The clock frequency of 1 GHz and packet size is 32 bytes, with a flit interval of 2 clock cycle.

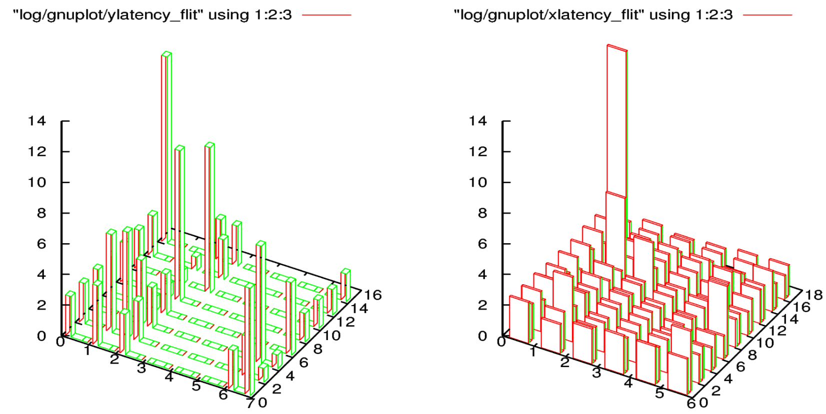

***Figure5**. Overall average channel latency (in clock cycles per flit)*

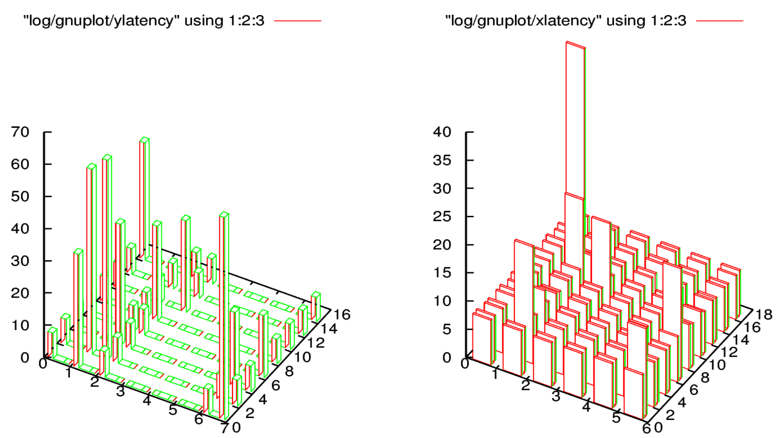

***Figure6.** Overall Average Channel Latency (in clock cycles per Packet)*

The overall average latency per channel, measured in clock cycles per flit, refers to the average time it takes for a flit to traverse a communication channel within a network, including all processing and transmission delays, measured in clock cycles. Figure 5 shows a screenshot of the overall average channel latency (in clock cycles per flit) generated by NIRGAM simulator. The overall average latency per channel, measured in clock cycles per packet, refers to the average time it takes for a packet to traverse a communication channel within a network. Figure 6 shows the overall average channel latency (in clock cycles per *Packet*) for a particular network

The simulation results are evaluated for overall average channel latency (clock cycle/packet) for different size of network in uniform random traffic pattern.

Table1. Overall Average Channel Latency (clock cycle/packet)

Network size	Overall Average Channel Latency (clock cycle/packet)			
	XY	*OE*	*DYAD*	*Proposed algo.*
3x5	84.6953	97.473	87.4666	84.5847
3x6	84.9026	87.115	85.6061	83.806
5x7	92.7272	95.4428	94.3086	92.033
5x9	98.5079	118.291	100.593	97.4156
9x7	95.5349	122.057	91.3552	85.6383
9x5	75.3954	87.9591	72.9357	72.0258

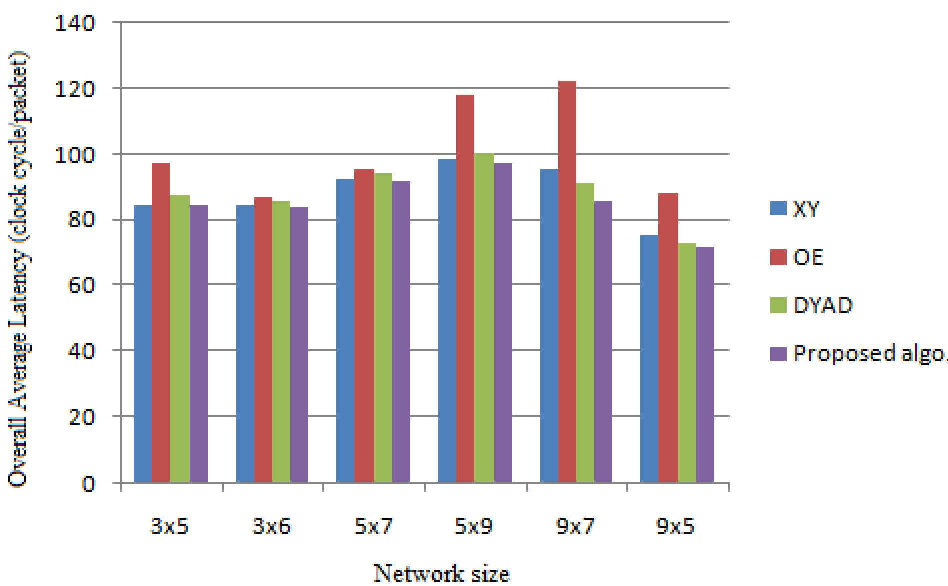

Figure 7: Performance analysis of various routing algorithm

Figure 7 shows the overall average latency of XY, OE, DYAD and proposed routing algorithms. The OE algorithm has the highest average latency while proposed routing algorithm has lowest average latency among the four algorithms. From the above result we see that for any dimension of network the proposed routing algorithm give the better result. OE algorithm performs better on high traffic of network. If there are average or low traffic XY routing algorithm perform better than the OE algorithm. Proposed routing algorithm give better result on any traffic (low or high traffic) in the network. The overall average latency of DyAD algorithm is slightly higher than the proposed algorithm. The DyAD routing algorithm perform better on regular (squire shape) mesh network, while proposed routing algorithm perform better on both rectangular and squire shape mesh network.

VI. CONCLUSION

In this paper we proposed an efficient adaptive routing algorithm. In contrast to other adaptive routing algorithms, the proposed routing algorithm avoids congestion in any 2-Dimensionl regular or irregular mesh network. Performance evaluation of XY, OE, DYAD and proposed routing algorithms is carried out on the NIRGAM simulator. Results of performance measurement show effectiveness of proposed routing algorithm compared to static XY routing, odd-even routing, and adaptive routing algorithms. The proposed routing algorithm chooses the optimal path between source node and destination node. The proposed routing algorithm also worked for 2-dimensional mesh as well as Torus network. The algorithm can be extended to work for 3-dimensional regular or irregular network.

REFERENCES:

1. Yuan Cai , Dong Xiang and Xiang Ji, "Deadlock-Free Adaptive Routing Based on the Repetitive Turn Model for 3D Network-on-Chip", *IEEE Intl Conf on Parallel & Distributed Processing with Applications, Dec. 2018*
2. Yoon Seok Yang , Hrishikesh Deshpande , Gwan Choi and Paul V. Gratz "SDPR: Improving Latency and Bandwidth in On-Chip Interconnect Through Simultaneous Dual-Path Routing", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol: 37 , No. 3 , pp. 545 – 558, March 2018*.
3. Jindun Dai, Xin Jiang, Renjie Li and Takahiro Watanabe "An Efficient Deadlock-Free Adaptive Routing Algorithm for 3D Network-on-Chips", *IEEE 11th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC), pp. 29 – 36, 2017*
4. Vahid Janfaza and Elaheh Baharlouei "A new fault-tolerant deadlock-free fully adaptive routing in NOC", *IEEE East-West Design & Test Symposium (EWDTS), pp 1-6, 2017*
5. Canwen Xiao, Yue Yang, and Jianwen Zhu "A Sufficient Condition for Deadlock-Free Adaptive Routing in Mesh Networks", *IEEE Computer Architecture Letters, Vol. 14, No. 2, July-December 2015*
6. Freek Verbeek and Julien Schmaltz" A Decision Procedure for Deadlock-Free Routing in Wormhole Networks," *IEEE Transactions On Parallel And Distributed Systems Vol. 25, No. 8, August 2014.*
7. Mark Karpovsky, Lev Levitin and Mehmet Mustafa "Optimal Turn Prohibition for Deadlock Prevention in Networks with Regular Topologies," *Transactions On Control Of Network Systems, Vol. 1, No. 1, March 2014*
8. Wei Luo and Dong Xiang, "An Efficient Adaptive Deadlock-Free Routing Algorithm for Torus Networks" IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, NO. 5, MAY 2012

9. D. Xiang, "Deadlock-free Adaptive Routing in Meshes with Fault-Tolerance Ability Based on Channel Overlapping," *IEEE Trans. Dependable and Secure Computing, vol. 8, no. 1, pp. 74-88, Jan./Feb. 2011.*

10. D. Xiang, Y. Pan, Q. Wang, and Z. Chen, "Deadlock-Free Fully Adaptive Routing in 2-Dimensional Tori Based on New Virtual Network Partitioning Scheme," *Proc. 28th Int'l Conf. Distributed Computing Systems, pp. 454-461, June 2008.*

11. Singh, W.J. Dally, A.K. Gupta, and B. Towles, "Goal: A Load-Balanced Adaptive routing Algorithm for Torus Networks," *SIGARCH Computer Architecture News, vol. 31, no. 2, pp. 194-205, 2003.*

12. J. Duato and T.M. Pinkston, "A General Theory for Deadlock-Free Adaptive Routing Using a Mixed Set of Resources," *IEEE Trans. Parallel Distributed Systems, vol. 12, no. 12, pp. 1219-1235, Dec. 2001.*

13. G M. Chiu, "The Odd-Even Turn Model for Adaptive Routing," *IEEE Trans. Parallel and Distributed Systems, vol. 11, no. 7, pp. 729-738, July 2000.*

14. Rajendra V. Boppana and Suresh Chalasani "A Framework for Designing Deadlock-Free Wormhole Routing Algorithms" *IEEE Transactions On Parallel And Distributed Systems, Vol. 7, No. 2, February 1996.*

15. J. Duato, "A Necessary and Sufficient Condition for Deadlock-Free Adaptive Routing in Wormhole Networks," *IEEE Trans. Parallel Distributed Systems, vol. 6, no. 10, pp. 1055-1067, Oct. 1995.*

16. L.M. Ni and P.K. McKinley, "A Survey of Wormhole Routing Techniques in Direct Networks," IEEE Computer, Vol. 26, pp. 62-76, 1993

17. Y.M. Boura and C.R. Das, "A Class of Partially Adaptive Routing Algorithms for n-dimensional Meshes," *Proc. Int'l Conf. Parallel Processing, vol. III, pp. 175-182, Aug. 1993.*

18. C. Glass and L.M. Ni, "The Turn Model for Adaptive Routing," *Proc. 19th Ann. Int'l Symp. Computer Architecture, pp. 278-287, May 1992.*

19. J. Duato, "On the Design of Deadlock-Free Adaptive Routing Algorithms for Multicomputers: Design Methodologies," *Proc. Parallel Architectures and Languages Europe 91, pp. 390-405, June 1991.*

20. W.J. Dally and C.L. Seitz, "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks," IEEE Trans. Computers, vol. C-36, no. 5, pp. 547-553, May 1987.

21. Kawano et al."A Generalized Theory Based On The Turn Model For Deadlock-Free Irregular Networks", IEICE TRANS. INF. & SYST., Vol.E103–D, No.1 January 2020

22. P.R. Nuth and W.J. Dally, "The J-Machine Network," Proc. IEEE Int'l Conf. Computer Design: VLSI in Computer and Processors, pp. 420-423, 1992.

23. R.E. Kessler and J.L. Schwazmeier, "CRAY T3D: A New Dimension for Cray Research," Proc. COMPCON '93, pp. 176-182, 1993

24. Cray T3E Scalable Parallel Processing System. Cray Research Inc., 1995.

25. A Touchstone DELTA System Description. Intel Corp., 1990

26. P. Kermani and L. Kleinrock, ªVirtual Cut-Through: A New Computer Communication Switching Technique, Computer Networks, vol. 3, no. 4, pp. 267-286, Sept. 1979.