

Level Based Routing Using Dynamic Programming for 2D Mesh

Akash Punhani¹, Pardeep Kumar¹, Nitin Nitin²

¹Jaypee University of Information Technology, Solan, Himachal Pradesh 173234, India

²Jaypee Institute of Information Technology, Noida, Uttar Pradesh, India

E-mails: akashpunhani@gmail.com pardeepkumarkhokhar@gmail.com delnitn@gmail.com

Abstract: *The performance of the interconnection network doesn't only depend on the topology, but it also depends on the Routing algorithm used. The simplest Routing algorithm for the mesh topology in networks on chip is the XY Routing algorithm. The level based Routing algorithm has been proved to be more efficient than the XY Routing algorithm. In this paper, level based Routing algorithm using the dynamic programming has been proposed. The proposed Routing algorithm proves to be more efficient in the terms of the computation. The proposed Routing algorithm has achieved up to two times bigger speed.*

Keywords: *Dynamic programming, Routing algorithm, Mesh topology, Time complexity, Space complexity.*

1. Introduction

The interconnection network is the major component of the digital systems. In smaller systems, we have used the common bus architectures for the communication between the various devices. But with the thirst for getting more and more computational speed, the processors and memory devices are made faster; this leads to the requirement of dedicated lines between the various components of the system [1, 2]. In present scenario, the multiple processors can be used to achieve the speedup. These multiple processors have to communicate with each other and require some topology and Routing algorithm for routing the packets. Based on the topology there exist various Routing algorithms that may be suitable in particular cases [3-7]. The most popular and simplest topology is the mesh topology and the simplest Routing algorithm is the XY Routing algorithm. The Routing algorithm has the responsibility to route the packet from the source node to destination node with the key knowledge of the underlying hardware. The Routing algorithms are used in both types of networks, whether it is a regular network or irregular network [3-7].

The various variants of the mesh topologies [8-16] has been suggested in the past; that proves to be more efficient than Mesh topology and have used the routing derived from the *XY* Routing algorithm. This makes a point to study the *XY* routing in details because the study will not only support mesh topology, but it forms the bases for designing the Routing algorithm for the variants of the mesh topology. The various parameters have been used to analyse the performance of the Routing algorithm. As a Routing algorithm is a type of algorithm used for routing, so, the main objective in designing a new Routing algorithm should focus on the time complexity and space complexity. These two factors are indirectly affecting the various design parameters of the routers. The complex Routing algorithms will also require the large number of logical gates to perform the operations and thereby increasing the chip area. The execution of the complex Routing algorithm will also increase the decision time of the routing and requires the more power and dissipates more heat. The main objective of our paper is to study various existing *XY* routing and its alternatives, and design a faster and area-efficient algorithm for the network on chips.

This paper is organized as follows: Section 2 discusses the underlying architecture of the router and existing Routing algorithm. In Section 3, we propose the Level based Routing algorithm using dynamic programming. In Section 4, we analyse and compare the existing and proposed schemes in terms of execution time and memory required to represent the port address. Section 5 concludes this paper.

2. Underlying architecture and Routing algorithm

To study the detailed analysis of Routing algorithms the detailed understanding of the mesh along with the employed router architecture should be understood.

2.1. A two-dimensional mesh topology

A two-dimensional mesh topology has been described in Fig. 1. Fig. 1 primarily described two prime components, the processors are represented by the circles and squares are used to describe the routers. These hardwares are connected to each other by the links and the routers are connected to its adjacent neighbours both in the horizontal and vertical manner. The label on the nodes is based on the Routing algorithm employed in the Routing algorithm. There exist various Routing algorithms for Mesh topology [17-21] that may be simple or adaptive in nature. The most common type of Routing algorithm used in the networking makes the decision on the source, but they are not suitable for the network on chips as the routing tables are needed to maintain at each node. This will increase the area of the router which is a prime factor while designing the network on chip. In *XY* Routing algorithm the labels are based on the coordinates which will require to memory blocks that will represent the *X* and *Y* coordinate of the node (Router+ Processing element). In the case of level based Routing algorithm or index based Routing algorithm this label is a single integer value that begins from 0 up to $n - 1$ [21].

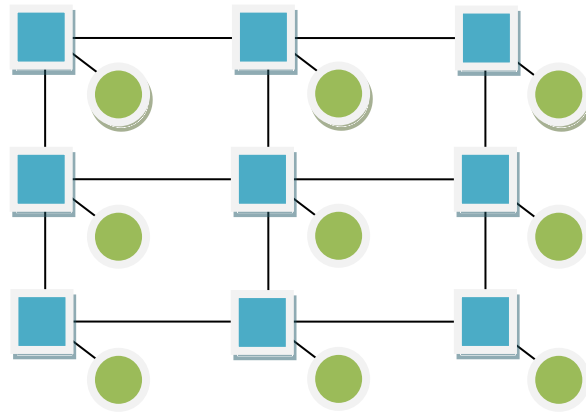


Fig. 1. Two-dimensional mesh topology

The delivery of messages from the source to destination will be dependent on the router and the Routing algorithms used. Before studying the Routing algorithm, the knowledge of the router architecture is essential. For designing a mesh topology, a five-port router is used. The detailed overview of the router architecture has been described in the Fig 2. In Fig. 2 the router has five ports, that is the four ports connected to other routers and one port is a local port that is connected to the processing element. By the processing element, it should not be interpreted as a simple processor but it is an element that may use the network for communication like Processor, memory element, or any other Intellectual Property (IP). The number of ports the router used in the mesh topology is limited to only 5 ports that is the next state of the system can be any one of the five ports east, west, north, and south or the local port. These ports can be labelled according to the various conventions either by the predefined labels for the ports or by the labels which are assigned to the node of the adjacent router to which a particular node is connected.

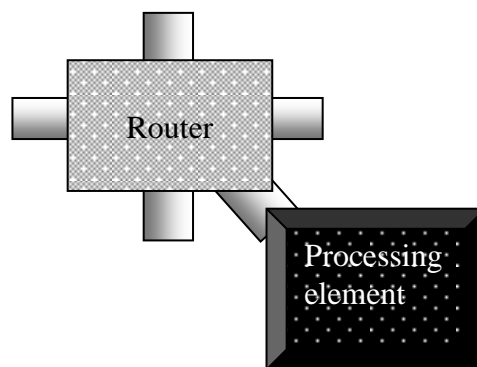


Fig. 2. Router architecture for the 2D mesh

2.2. XY Routing algorithm

The XY Routing algorithm have been efficiently described in various literatures, it is the combination of nested if else statements. In XY routing, first the comparison of the X coordinates values of the current router is done with destination X coordinate value. Based on the comparison the packet is transferred in the eastward or westward direction ports. Once X coordinate of the current router and destination address are same, then the comparison based on the Y coordinates of the router is done. This comparison will route the packet to north, south or to the local port of the router. For the comparison, the detail algorithm is described below with cost and time representations.

Algorithm for XY routing

<i>Input:</i> Source Coordinates and destination coordinates		
<i>Output:</i> Destination Port		
	Cost	Time
XY_Routing (Xcurrent, Ycurrent, Xdest, Ydest)		
1. Xoffset = Xdest – Xcurrent;	C_1	1
2. Yoffset = Ydest – Ycurrent;	C_1	1
3. if Xoffset < 0 then	C_2	1
4. Port = West;	C_3	1
endif		
5. if Xoffset > 0 then	C_2	1
6. Port= East;	C_3	1
endif		
7. if Xoffset = 0 and Yoffset < 0 then	C_4	1
8. Port= North;	C_3	1
endif		
9. if Xoffset = 0 and Yoffset > 0 then	C_4	1
10. Port= South;	C_3	1
endif		
11. if Xoffset = 0 and Yoffset = 0 then	C_5	1
12. Port= Local;	C_3	1
endif		
End		

The execution time can be computed as $2C_1+2C_2+5C_3+2C_4+C_5$ as all the statements are executed in the single unit of the time, but the constant associated with them are different based on the type of the operation is performed. It is not feasible to use the complexity notation like Big-Oh notations as it will be always constant not in a polynomial representation.

2.3. Level based Routing algorithm

In the detailed performance analysis of the Routing algorithms we have studied the Level Based (LB) Routing algorithms provided by the author's [21] in which they have claimed that their Routing algorithm is faster than the existing XY Routing algorithm as it uses the less number of comparisons. The algorithm first directs the packet in the Y direction, then in X direction so it can be described as a variant of the YX Routing algorithm. To Estimate the cost, we have used the same constants for the same operations so we can get the exact idea of the execution time. Here

division is performed in place of subtraction performed in the *XY* Routing algorithm. The division is 4 times more complex than that of subtraction let this cost be represented by C_6 . To compute port address, instead of assignment an arithmetic addition operation is used, but the cost of the execution of arithmetic operation is higher in comparison to that of assignment that why C_1 is used instead of C_3 . The Total time complexity of the existing algorithm will be $4C_1+3C_2+2C_6$.

Algorithm for Level based routing

<i>Input:</i> Source ID and destination ID		
<i>Output:</i> Destination Port		
	Cost	Time
int LB_routing(int curr_node,int dest_node)		
{		
int level1=curr_node/n;	C_6	1
int level2=dest_node/n;	C_6	1
1. if(level1==level2)	C_2	1
{		
2. if(curr_node<dest_node)	C_2	1
return(curr_node+1);	C_1	1
else		
3. return(curr_node-1);	C_1	1
}		
4. if(level1<level2)	C_2	1
return(curr_node+n);	C_1	1
else		
5. return(curr_node-n);	C_1	1
}		

3. Level based routing using Dynamic programming

The Level Based routing using Dynamic Programming (LBDP) algorithm is based on the fundamental that in Dynamic Programming to reduce the cost of execution the redundant computations are stored rather than of computing again and again. The same approach has been used in the proposed level Routing algorithm. In the level, based routing we first calculates level for the current node, and then compares the level with the destination node. The calculation of the level for the current is redundant at each packet which can be calculated only once and used again for all the coming packets. This will reduce the time consumed by the Routing algorithm. To do so, we have initialized a constant with level number for the router. Another modification that has been introduced in the level based Routing algorithm is a correctness introduced in the level based routing. The existing level based routing was unable to route the packets correctly as there doesn't exist the code for routing the packet to the local port of the router. The detailed Routing algorithm has been described below. The running cost of the algorithm is again $4C_2+ 5C_3+C_6$.

Proposed level based routing using Dynamic programming

<i>Input:</i> Source ID, destination ID and level assigned at the time initialization		
<i>Output:</i> Destination Port		
	Cost	Time
int LBDP_routing(int curr_node,int dest_node)		
{		
int level2=dest_node/n;	C_6	1
1. if (level1==level2)	C_2	1
{		
2. if (curr_node<dest_node)	C_2	1
return East;	C_3	1
3. if (curr_node>dest_node)	C_2	1
return West;	C_3	1
else		
return local;	C_3	1
}		
4. if (level1<level2)	C_2	1
return South;	C_3	1
else		
5. return North;	C_3	1
}		

3.1. Proof of correct

To prove the correctness of the proposed algorithm we have two cases:

When current and destination addresses are at the same level, we can have three sub-cases.

a. Current address is greater than destination address: The packet is directed to the West port according to the topology the node on the left of the current node will be less at a particular level.

b. Current address is less than destination address: The packet is directed to the east port as the node to the left of the current node at particular level is always grater.

c. Current address is equal to destination address: When the current address and destination address are the same, this implies packet is at destination node and should be routed to the local port.

When current address and destination address are not at the same level, we can have two sub cases.

d. Current node is at a lower level than the destination router: The packet has to be routed in the southward node as node above the given node will be always at the lower level.

e. Current node is at a greater level than the destination router: The packet has to be routed in the northward node as node above the given node will be always at the lower level.

As all the five cases of routing the packets are working correctly, therefore the Routing algorithm stated is working correctly.

4. Performance analysis

The performance of any algorithm depends upon two factors that are the execution time and Memory consumption of the algorithm.

4.1. Comparison based on the execution time

The execution time of the level based routing is compared to the proposed dynamic programming based Routing algorithm. To analyse the performance of the Routing algorithm we have called the Routing algorithm N number of times. The reason for doing so is to identify the difference in the two Routing algorithms because the single execution of the algorithm is very fast and the difference in execution time cannot be identified exactly on the machine. The results are obtained on the hardware configuration as described in Table 1.

Table 1. Describing the various hardware used for obtaining the results

No	Parameter	Specification
1	Processor	Intel® Core™ 2 CPU T5200@ 1.6 GHz
2	Ram	2 GB

The results obtained have been reported in microseconds and are described in Table 2 and compared in Fig. 3.

Table 2. Describing the Execution time of various Routing algorithms

N	Level based Routing algorithm, μs	Proposed Dynamic programming based algorithm, μs
10 000	424	169
40 000	1205	626
90 000	3020	1197
160 000	4352	3046
250 000	10 643	4451
360 000	16 252	6395
490 000	21 753	9465
640 000	29 725	13 566
810 000	38 090	16 713
1 000 000	48 529	20 739

From the results shown in Fig. 3, we can infer that the proposed dynamic programming based level based routing is always efficient in comparison to that of existing level based Routing algorithm. The result obtained reveals that in the terms of execution time, proposed level based routing using a dynamic programming algorithm is always efficient and has given the improvement of almost of 2X.

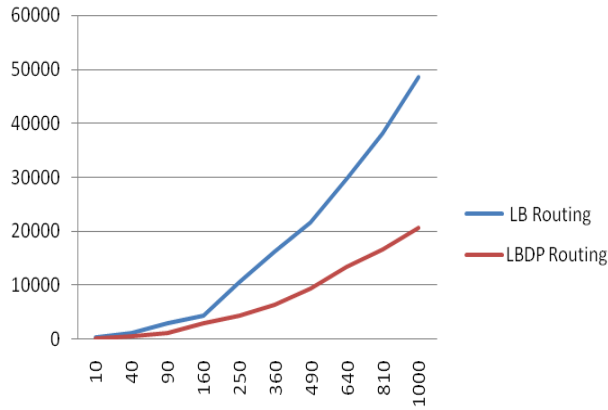


Fig. 3. Execution time of the Routing algorithms (in μs) per Number of nodes ($N \times 1000$)

4.2. Comparison based on the Space complexity

Another commonly used parameter to analyse an algorithm is the amount of the memory required by the particular algorithm. In the present Routing algorithms, our main attention is on the number of bits required to store the port number onto which the packet is send.

In the case of LB routing the port numbers are $m - 1$, $m + 1$, $m - k$ and $m + k$, m for the five ports of the router. Here m is the id of the current node and k is the number of nodes at a particular level. Now, as the Routing algorithm should be the same for every router in the mesh, so the maximum value of the entire possible router will be the actual number of bits required. As the maximum id in the mesh is $N - 1$. That is the number of nodes in the mesh. So, the number of bits required, will be given by the equation

$$(1) \quad b = \log_2 N.$$

In the case of proposed LBDP Routing algorithm the maximum number of bits required, will be equal to 3 as there will be maximum 5 ports in any router. The detailed comparison of the number of bits required is shown in the Fig. 4 and Table 3.

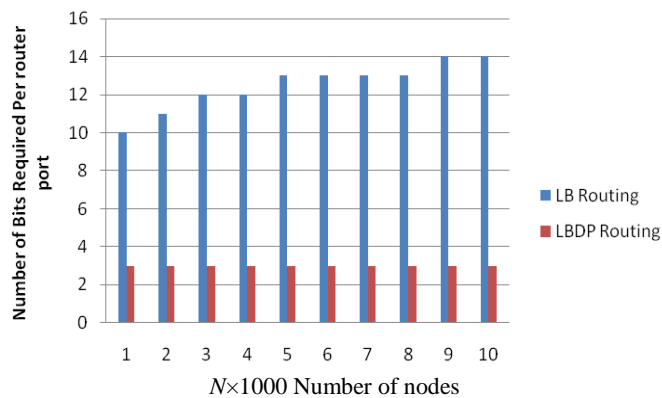


Fig. 4. Bits required for representing the port number

Table 3. Describing the number of bits required per port per router

Number of nodes	Number of bits required per router per port address	
	LB routing	Proposed DPLB routing
1000	10	3
2000	11	3
3000	12	3
4000	12	3
5000	13	3
6000	13	3
7000	13	3
8000	13	3
9000	14	3
10 000	14	3

5. Conclusion

The proposed Routing algorithm has proved to be efficient in the terms of both space and time complexity of the existing LB Routing algorithm, which has been suggested as an alternative to the *XY* Routing algorithm. The presented Routing algorithm will always generate the correct path and has been designed to getting the speed up by the factor of $2X$. The fastest algorithm will definitely require less number of clock cycles, thus it will reduce the hardware cost and the power consumption that might be consumed due to the calculations in the redundant part of the algorithm being removed by using the dynamic programming approach. In future, our focus will be on an adaptive routing that has complex computation which may have redundant computations.

References

1. Dally, W. J., B. Towels. Principle and Practices of Interconnection Networks. Elsevier, 2004.
2. Duato, J., S. Yalamanchili, L. Ni. Interconnection Networks. Elsevier, 2003.
3. Palesi, M., M. Daneshdala b. Routing Algorithms in Networks-on-Chip. Berlin, Springer, 2014.
4. Bhardwaj, V. P., R. V. Nitin. On the Minimization of Crosstalk Conflicts in a Destination Based Modified Omega Network. – Journal of Information Processing Systems, Vol. **9**, 2013, No 3, pp. 301-314.
5. Nitin, R. V., U. Srivastava. On a Deadlock and Performance Analysis of ALBR and DAR Algorithm on X-Torus Topology by Optimal Utilization of Cross Links and Minimal Lookups. – Journal of Supercomputing, Vol. **59**, 2010, No 3, pp. 1252-1288.
6. Nitin, R. V., D. S. Chauhan. Stochastic Communication for Application Specific Networks-on-Chip. – Journal of Supercomputing, Vol. **59**, 2010, No 2, pp. 779-810.
7. Nitin, R. V., S. Garhwal, N. Srivastava. Designing a Fault-Tolerant Fully-Chained Combining Switches Multi-Stage Interconnection Network with Disjoint Paths. – Journal of Supercomputing, Vol. **55**, 2009, No 3, pp. 400-431.

8. Yu-Hang, L., Z. Ming Fa, W. Jue, Li-Min Xiao, G. Tao. X-Torus: An Extended Torus Topology for On-Chip Massive Data Communication. – In: Proc. of 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW'12), 2012, pp. 2061-2068.
9. Hu, W. H., S. E. Lee, N. Bagherzadeh, Z. Xiao-Jing, H. Wei-Wu, M. Ke, Z. Long-Bing. X-Mesh: A Mesh-Like Topology for Network on Chip. – Network. Chip Architecture, 2008, p. 14.
10. Wang, Y., H. Du, X. Shen. Topological Properties and Routing Algorithm for Semi-Diagonal Torus Networks. – J. China Univ. Posts Telecommunication, Vol. **18**, 2011, No 5, pp. 64-70.
11. Arora, L. K., Rajkumar. C2-Mesh. – In: Proc. of 3rd IEEE International Advance Computing Conference (IACC'13), 2013, pp. 282-286.
12. Yadav, S., C. R. Krishna. CC-Torus: A New Torus Topology for Interconnection Networks. – In: Proc. of International Conference on Advanced Computational Technologies & Creative Media (ICACTCM'14), 2014, pp. 14-15.
13. Manish, Bhardwaj. C2-Torus New Interconnection Network Topology Based on 2D Torus, American. – Journal of Network Communication, Vol. **4**, 2015, No 3, pp. 1-4.
14. Punhani, C. A., R. V. Nitin, P. Kumar. A Modified Diagonal Mesh Interconnection Network. – In: Annual IEEE India Conference (INDICON'14), 2014, pp. 1-6.
15. Punhani, C. A., P. Kumar, R. V. Nitin. Diagonal Connected T-Mesh. – Indian Journal of Science and Technology, Vol. **9**, 2016, No 32.
16. Punhani, C. A., R. V. Nitin. EMC2-Mesh. – In: Annual IEEE Conference (INDICON'15), India, 2015, pp. 1-5.
17. Nitin, R. V., D. S. Chauhan. Comparative Analysis of Traffic Patterns on k -Ary n -Tree Using Adaptive Algorithms Based on Burton Normal Form. – Journal of Supercomputing, Vol. **59**, 2010, No 2, pp. 569-588.
18. Mejjia, M. Palesi, J. Flich, S. Kumar, P. Lopez, R. Holmsmark, J. Duato. Region-Based Routing: A Mechanism to Support Efficient Routing Algorithms in NoCs. – IEEE Trans. Very Large Scale Integration of Systems, Vol. **17**, 2009, No 3, pp. 356-369.
19. Chiu, G. M. The Odd-Even Turn Model for Adaptive Routing. – IEEE Trans. Parallel Distributed Systems, Vol. **11**, 2000, No 7, pp. 729-738.
20. Glass, J., L. M. Ni. The Turn Model for Adaptive Routing. – In: Proc of 19th Annual International Symposium on Computer Architecture, 1992, pp. 278-287.
21. Arora, L. K., R. Kumar. Alternatives of XY-Routing for Mesh. – In: Special Issue of International Journal of Computer Application ICNICT, 2012, pp. 6-8.