A New Routing Algorithm for Irregular Mesh NoCs

V. Samadi Bokharaei1, A. Shamaei2,3 , H. Sarbaziazad2,3 and M. Abbaspour1,2
(1) Department of Electrical and Computer Engineering, Shahid Beheshti University, Tehran, Iran
(2) IPM School of Computer Science, Tehran, Iran
(3) Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
v_samadi@std.sbu.ac.ir, shamaei@ipm.ir, azad@ipm.ir, maghsoud@ipm.ir

Abstract— **Network-on-Chips (NoCs) usually use regular mesh-based topologies. Regular mesh topologies are not always efficient because of power and area constraints which should be considered in designing system-on-chips. To overcome this problem, irregular mesh NoCs are used for which the design of routing algorithms is an important issue. This paper presents a novel routing algorithm for irregular mesh-based NoCs called "i-route". In contrast to other routing algorithms, this algorithm can be implemented on any arbitrary irregular mesh NoC without any change in the place of IPs. In this algorithm, messages are routed using only 2 classes of virtual channels. Simulation results show that using only 2 virtual channels, "i-route" exhibits a better performance compared to other algorithms already proposed in the same context.**

***Keywords*: NoCs, Mesh, Irregular mesh, Routing algorithm, Virtual channels, Performance.**

I. Introduction

Continues growth in the number of transistors on a chip has led to the idea of Network-on-Chip (NoC). It is predicted that on 2010 there will be chips available which would contain more than one billion transistors [1]. NoC uses an interconnection network to connect different IPs embedded on a chip instead of using old bus network structures.

Because of area and power usage restriction in designing of chips, NoCs usually use simple topologies and also simple routing algorithms. The 2D mesh topology is one of common topologies in designing NoCs because it is simple and also compatible with the basic 2D nature and constraints of VLSI layout designs. The XY routing algorithm is the best known routing algorithm for mesh NoCs which is perhaps the simplest one too. According to this algorithm, a message is routed along axis X until the x-value of the current node address is equal to that in the destination node. Then, the message is routed towards its destination along Y dimension.

In some SoCs, IPs may be of different sizes that lead to using irregular mesh NoCs. An algorithm for routing in irregular mesh NoCs has been introduced in [2]. This algorithm is based on odd-even turn model [3] and extended-XY routing [4]. The main drawback of the algorithm is that it is not applicable to all irregular mesh-based NoCs. To overcome this drawback and to support routing in more irregular mesh topologies, an IP placement algorithm is introduced in [2]. This algorithm changes locations of IPs in a way that the proposed routing algorithm can be applicable to the new topology. Because of restrictions on power consumption and network performance and also because of hot-spot considerations, it is not possible to change the places of IPs always and this solution may not be feasible for some cases. In [5] a routing table minimization algorithm for irregular NoCs is proposed. The routing is based on Dijkstra's shortest path algorithm. It uses Dally's algorithm [6] to avoid deadlocks. The number of virtual channels used by the routing algorithm depends on network topology and the formation of shortest paths. In other words, the number of virtual channels which are used by the routing algorithm is not known in advance.

In this paper, we propose a new routing algorithm that can route messages in an irregular mesh NoC with any arbitrary topology using only 2 virtual channels. Simulation results reveal that for different working conditions the proposed routing method here shows a better performance compared to other routing algorithms already proposed in the literature.

II. Preliminaries

Regular 2D Mesh. In a 2D $K_0\times K_1$ mesh network, a router R is identified by a two-element vector (x_0,x_1), $0\leq x_0<K_0$ and $0\leq x_1<K_1$, where x_0 and x_1 are called the coordinates of dimension 0 and dimension 1, respectively, and show the row number and column number of a router on the network. Router (x_0,x_1) is connected to 4 routers $(x_0\pm1, x_1\pm1)$ (some may be unavailable). A 4x4 regular mesh NoC is shown in Fig. 1.

Irregular 2D Mesh. In this type of mesh NoCs, in addition to boundary routers, there are some other router(s) which have less than 4 neighbors. This is because of the heterogeneity in IP sizes which causes irregularity in the shape of mesh structure. Figure 2 (top) shows a 4x6 irregular mesh NoC.

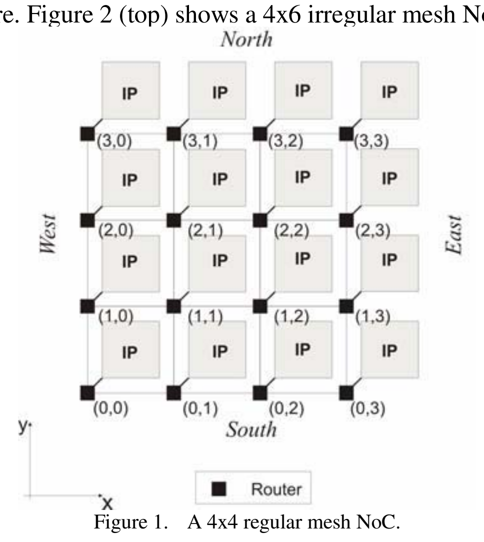


Figure 1. A 4x4 regular mesh NoC.

Figure 2 (down) shows an IP of 4 times the size of a normal IP, surrounded by the routers. Routers which surround an irregular IP are defined as surrounding ring or briefly called as "ring". Irregular IPs have routers on their ring with less than four neighbors. In this case, some of messages can not continue their normal way using XY routing since the current router may have some missing neighbors. This is the situation that we call it *blocking* of a normal routing path for a message. A solution is to route the message around the irregular IP on its related ring, to get to the next hop on its normal path. This situation is shown in fig. 3.

Message Types. In XY routing messages are classified as row messages and column messages. Row messages travel along a row till they get to the column which the destination is resided on it. At this point, the message changes its type to a column message and continue its way to get to destination along that column. Row messages, according to their direction of traveling, are classified as West-to-East (WE) and East-to-West (EW) messages. Column messages are also classified as North-to-South (NS) and South-to-North (SN) messages.

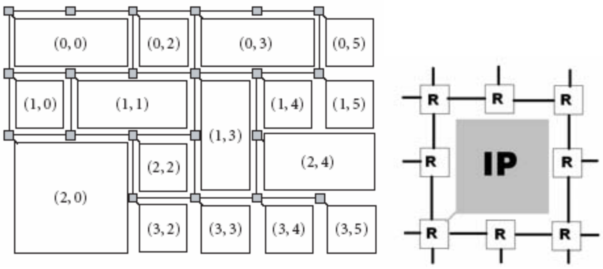


Figure 2. A 4x6 irregular mesh (left); the way that an IP is connected to the network using sorrounding routers (right).

As it is mentioned before, in regular mesh networks, a row message travels only in rows. In irregular mesh networks, it is a little bit different. Because of the oversized IPs some of row messages should travel along the oversized IPs' rings. In these situations, messages should travel along vertical (column) channels as part of its way to destination. Like regular mesh networks, a row message changes its type to column message when it gets to the column which the destination is located in it.

III. ROUTING IN IRREGULAR MESH NOCS

In this section, we introduce a new routing algorithm that can support all kinds of irregular mesh NoCs with oversized IPs. The proposed algorithm, called *i-route*, can route packets on any irregular mesh structure only using 2 virtual channels.

The Routing Algorithm. The pseudo-code of the proposed routing algorithm is shown in Figure 4. In the 1st step of *i-route*, the message is consumed in the current router if it is already in its destination. If a row message has reached the destination column, the message type is changed to column type in the 2nd step of the algorithm. In the 3rd step, the message direction for routing to the next hop is determined. The message routing direction can be one of these three values: *Null, Clockwise, Counter-clockwise*.

Message direction is *null* if the message can continue on its normal XY hop and the next XY hop router is available; otherwise it continues its journey around the ring of irregular IP in clockwise or counter-clockwise direction, based on the message directions and message type. In the last step, the message is routed to the next hop based on the direction it has moved. Procedure *Set_direction* () is one of the main parts of the routing algorithm. It chooses the message direction such that it can continue its way to the final destination without causing deadlock in the network.

Virtual Channel Usage. We have used a simple method to decide which class of virtual channels should be used by different types of messages in order to prevent circular dependency among virtual channels and deadlock formation in the network. There are two classes of virtual channels: vc_0 and vc_1. In Figure 5, it is shown how messages use virtual channels according to their types.

IV. SIMULATION RESULTS

In this section, we report the results of simulation experiments for the proposed routing algorithm and the one proposed in [2] (called *EX-XY*). These two algorithms were implemented in Xmulator environment [7]. Xmulator is a simulator which is developed using C# language. Using Xmulator we have modeled different NoCs with different topologies under different traffic conditions. Using the simulator, the performance of the network under different message generation rates is evaluated and reported.

The duration of run-time is defined by the number of events happened in the simulation experiment. We have simulated 5 different 10x10 irregular mesh networks shown in Figure 6. In comparison to a 10x10 regular mesh network, there are 8 routers missing in each of the five simulated irregular mesh NoC. As can be seen in the figure, these networks are different in the number and size of oversized IPs.

Simulation experiments were run for 10,000,000 simulation events. In our simulation, each physical channel consists of 2 virtual channels as mentioned before. The routing algorithm *EX-XY* was evaluated only on networks 1, 2 and 3 since it can not route messages in networks 4 and 5. The *i-route* algorithm was however evaluated on all of the five considered networks. The simulation results are shown in Figure 7. Comparing the average message latency for networks 1, 2, and 3 shows that *i-route* exhibits a better performance compared to the EX-XY routing algorithm. Note that using routing algorithm EX-XY causes the network to reach the saturation point earlier.

As can be seen in the figure, the result for *i-route* is different for networks 1, 2, and 3. The relative performance of *i-route* compared to EX-XY routing is reduced from network 1 to network 3.

In the next simulation experiment, we fix the traffic generation rate at each node to 0.002 for a period of 10,000,000 simulation events. During this period, the number of messages which have been blocked since an output channel is not available in routes to move further is counted. This happens when some other message is using an output channel that is required by the message. The number of messages which have to wait to access the output channel is a good criterion to estimate the congestion on that router and associated output channel.

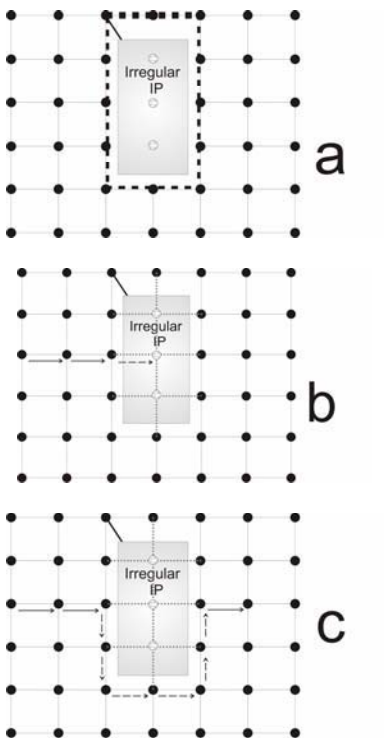


Figure 3. (a) the ring around an irregular IP shown with dashed lines, (b) the normal way of a message blocked by the irregular IP, (c) the message is routed around the ring of the irregular IP.

Algorithm i-route (message *M*)
/* Message *M* is currently reside at node (c_1,c_0) and the destination node is (d_1,d_0) */
1- if $(c_1,c_0)=(d_1,d_0)$ give the message to the local node & return.
2- if M is a row message and $c_0=d_0$ then change its type to
  a. NS, if $c_1>d_1$
  b. SN, if $c_1<d_1$
3- Set_Direction (*M*)
4- if the direction of *M* is null, then route *M* along its XY path,
  else route *M* on the ring of irregular IP according to the specified direction.

Procedure Set_Direction (*M*)
0. if *M* is a column message and its direction is null then set $(x_1,x_0)=(c_1,c_0)$
1. if M is a column message and c0=x0 and c1≠x1 then set its direction to null
2. if M is a column message and $c_0 \neq x_0$ then return
3. if next XY hop of *M* is available, set its direction to null and return
4. if direction of M in not null, then return
5. if *M* is WE message set its direction to Counter-Clockwise
6. if *M* is EW message set its direction to Clockwise
7. if *M* is a NS message Set its direction to:
  i. Counter-Clockwise if the current node is not located on the West boundary
  ii. Clockwise otherwise
  And set $(x_1,x_0)=(c_1,c_0)$
8. if *M* is a SN message Set its direction to:
  iii. Clockwise if the current node is not located on the West boundary
  iv. Counter-Clockwise otherwise
  And set $(x_1,x_0)=(c_1,c_0)$

Figure 4. The *i-route* AlgorithmThe *i-route* Algorithm

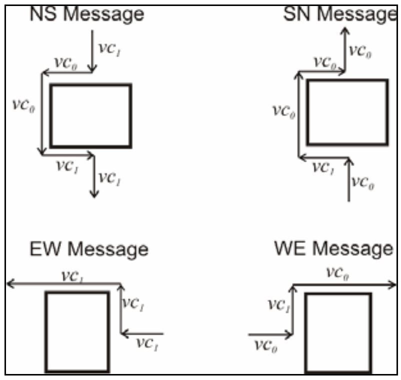


Figure 5. Virtual channel usage for different message types

In Table 1, the 10 most congested virtual channels of the network are listed. These virtual channels belong to the routers shown in Figure 8. It can be seen that these 10 routers are placed around the oversized IP. This happens because oversized IPs in the network obstacle the normal routing paths of messages. Thus, messages are led to turn around the oversized IPs on their corresponding rings. The congested virtual channels act as a bottleneck to route messages. We have chosen the most congested links as a criterion for estimating the network congestion. Figure 9 shows the number of blocked messages in most congested virtual channels of the 5 different networks simulated.

Now it is easy to justify different behaviors of the curves shown for the 5 networks in Figure 7. When a normal routing path for a message is blocked, the message starts to turn around the oversized IP. As the size of IP grows, the probability that a message is blocked by the oversized IP increases and, as a result, more messages may be routed around oversized IPs. This can cause more congestion in the routers located around oversized IPs.

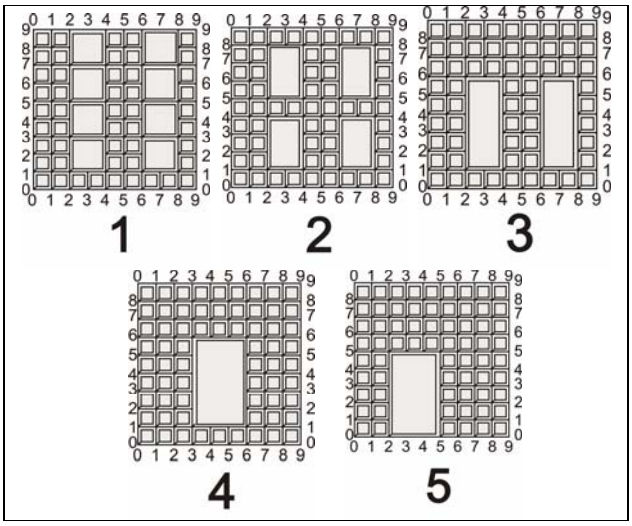
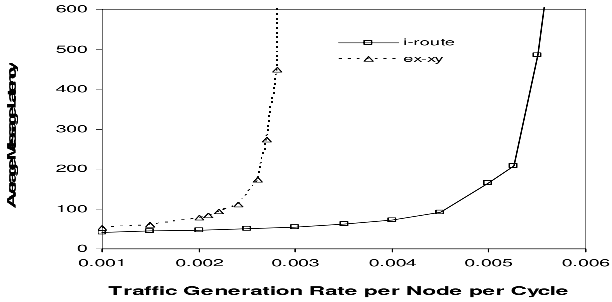


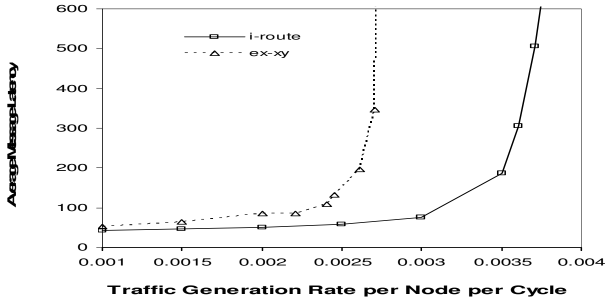
Figure 6. The five different irregular mesh NoCs simulated.

TABLE I. THE 10 MOST CONGESTED VIRTUAL CHANNELS

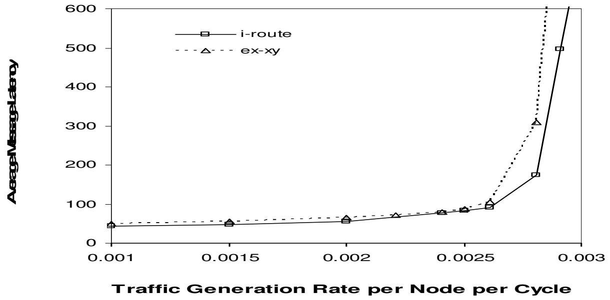
Router (row, column)	Output Physical channel	Virtual channel	# of blocked message
63	East	0	3722
66	West	1	2220
56	South	1	1942
46	South	1	1632
53	South	1	1438
43	South	1	1261
36	South	1	1237
33	South	1	815
65	West	1	698
64	West	1	580



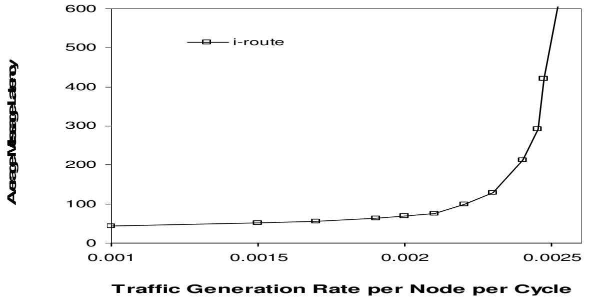
1



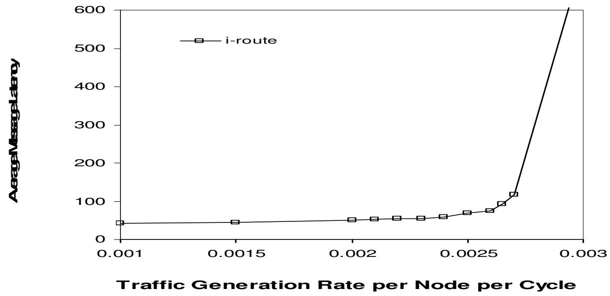
2



3



4



5

Figure 7. Simulation results for the 5 networks

As can be seen in the figures, the size of oversized IPs is increased in networks 1, 2, 3 and 4 and the number of oversized IPs are decreased. Although the area occupied by the oversized IPs is decreased from network 1 to network 4, the performance is decreased. Network 1 has the smallest oversized IPs and exhibits the best performance. Network 5 has the biggest oversized IP and has the worst performance among all simulated networks. In network 4, with regard to network 5, we have changed the position of the oversized IP from the center to the corner leading to a better performance. The reason is that the probability of blockage of normal routing paths for the messages in the network is decreased.

In brief, having more moderately oversized IPs compared to less large oversized IPs, with constraint of having the same number of routers, can result in a better performance. Placing the oversized IPs at corners (rather than placing them in centric parts of the network) can also improve the performance.

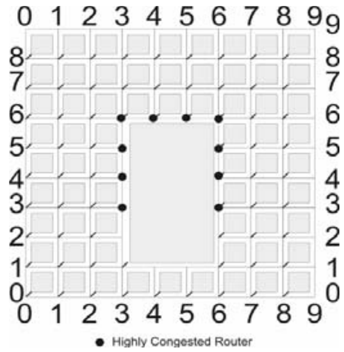


Figure 8. The 10 most congested routers

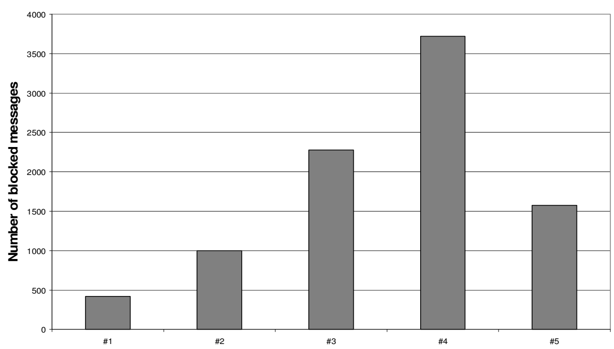


Figure 9. The number of blocked messages in most congested virtual channels of the 5 different networks

V. CONCLUSION

In this paper, we proposed a new routing algorithm for irregular mesh NoCs, called *i-route*. Only 2 Virtual channels are required to implement this algorithm in any arbitrary irregular mesh topology. Simulation results show that the performance of the algorithm is better than the other algorithm recently introduced in the literature for irregular meshes. The performance of *i-rotue* algorithm is dependent to the size and place of oversized IPs in the network. The performance of the network with the same number of routers is improved when smaller oversized IPs are used (compared to the cases where less number of larger oversized IPs are used). Among NoCs with similar oversized IPs (in number and size), those with oversized IPs located in the corners of the network provide better performance.

REFERENCES

[1] ITRS roadmap 2003. http://itrs.net.

[2] M.K.F. Schafer; T. Hollstein, H. Zimmer, M. Glesner, "Deadlock-free routing and component placement for irregular mesh-based networks-on-chip", In Computer-Aided Design, IEEE/ACM International Conference on, Nov. 2005, pp. 238- 245

[3] G.-M. Chiu, "The odd-even turn model for adaptive routing" In IEEE Transactions on Parallel and Distributed Systems, Vol.11, pp. 729–738, 2000.

[4] J.Wu, "A fault-tolerant and deadlock-free routing protocol in 2D-meshes based on odd-even turn model". In IEEE Transactions on Computers, Vol. 52, pp. 1154–1169, 2003

[5] E. Bolotin, I. Cidon, R. Ginosar, A. Kolodny, "Routing Table Minimization for Irregular Mesh NoCs", In Design, Automation, and Test in Europe, pp. 942-947,2007.

[6] W. Dally et al, "Deadlock-free message routing in multiprocessor interconnection networks" in IEEE Transaction of Computers, Vol. C-36, Issue 5, pp. 547-553, 1987.

[7] http://xmulator.org/