Adaptive Routing in Hexagonal Torus Interconnection Networks

Arash Shamaei and Bella Bose
School of Electrical Engineering and
Computer Science
Oregon State University
Corvallis, OR 97331–5501
Email: {shamaei,bose}@eecs.oregonstate.edu

Mary Flahive
Department of Mathematics
Oregon State University
Corvallis, OR 97331–5501
Email: flahive@math.oregonstate.edu

***Abstract*—The hexagonal torus network is a degree six toroidal network with rich topological properties. It was used in the design of HARTS machine at the University of Michigan, and more recently it has been proposed for cellular networks. The low diameter and less average hop distance of this network make it advantageous over other $2D$ toroidal network such as meshes and tori. This paper proposes a fully adaptive and deadlock-free routing algorithm for hexagonal torus networks based on virtual channel partitioning and the algorithm requires three virtual channels per physical channel to remain deadlock-free. Simulation results show that this algorithm is superior to the fully adaptive routing algorithm for $2D$ meshes and $2D$ tori of the same size.**

***Index Terms*—Interconnection topology, routing, hexagonal networks, deadlock-free**

I. Introduction

In a multiprocessor supercomputer, computing nodes connect to each other via an interconnection network. The nodes perform the computation by passing messages to other nodes through the network using a standard message passing mechanism such as the Message Passing Interface (MPI) [1]. Therefore, the topology of the network which identifies how the nodes are connected to each other is of great importance in terms of the network performance.

Toroidal topology including meshes and tori are well-known class of interconnection networks. This topology has been adopted by many parallel computers, such as the IBM BlueGene [1], the Cray T3D and T3E (3D torus) [2], the HP GS1280 multiprocessor [3], the J-machine [4], and the Tera Parallel Computer [5].

Another alternative for 2D torus network is the Eisenstein-Jacobi networks (often abbreviated as EJ networks). The EJ network was originally developed over two decades ago [6]. It is a 2D wraparound network of degree six and its topological properties were explored in [7] and [8]. Recently [9] shows that the EJ networks are a generalization of the hexagonal wraparound mesh (hexagonal torus) networks developed earlier in [10], [11], and [12]. The hexagonal torus network was used in the design of the Hexagonal Architecture for Real-Time Systems (HARTS) machine at the University of Michigan [10] and more recently it has been proposed for cellular networks [13].

A routing algorithm specifies the path that a message could take in order to reach to its destination in an interconnection network. Therefore, efficient routing algorithms and in particular shortest path routing algorithms are crucial for an interconnection topology. The authors of [10], [11], and [12] have proposed some shortest path routing algorithms for hexagonal networks. The hexagonal network has three edge disjoint Hamiltonian cycles [8]. In [10] and [11] each node is represented using a three component addressing scheme where i-th component specifies the order of the node in the i-th Hamiltonian cycle ($i = 1, 2, 3$).

Later on, [13] improve this addressing scheme by using fewer bits in each component; however, under this addressing every node can be addressed in more than one possible way. Recently, a two component addressing scheme for hexagonal networks has been proposed in [7] and [8]. This addressing scheme requires the minimal number of addressing bits and addresses every node utilizing its position at the angles of 0 and 60 degrees from the horizontal axis. We will employ this addressing scheme which is summarized in Section II.

As stated earlier, the routing algorithm plays an important role in the performance of a network. A routing algorithm needs to satisfy some properties. First, it has to be deadlock-free. That is, it has to guarantee the delivery of messages to their destination by avoiding infinite waiting cycles. In wraparound topologies routing algorithms often use extra virtual channels per physical channel to remain deadlock-free. This issue was not considered in any of the aforementioned routing algorithms. Another appealing property of a routing algorithm is the degree of adaptivity that it offers; that is, the number of different paths that the routing algorithm provides for a pair of source and destination. Algorithms are classified as deterministic (no adaptivity at all), partially adaptive and fully adaptive [14].

In this paper we propose a deadlock-free and fully adaptive routing algorithm for hexagonal torus networks. The routing algorithm uses three virtual channels per physical channel to remain deadlock-free. Moreover, it offers all possible shortest paths between any pair of source and destination. This algorithm is based on the shortest path routing algorithm given in [9]. Simulation results is Section IV show that the proposed

978-1-4799-1365-7/13/$31.00 ©2013 IEEE

algorithm for hexagonal torus network is far better than a fully adaptive routing algorithm for other 2D toroidal networks such as meshes and tori.

The rest of the paper is organized as follows. Section II briefly explains the topology of hexagonal torus networks. The proposed routing algorithm is presented in Section III. Section IV discusses the simulation results of the proposed algorithm and the summarizing conclusions are given in Section V.

II. The Hexagonal Torus Topology

The hexagonal torus topology which was used in the design of HARTS network at the University of Michigan [10], [11], [12] is denoted by $\mathcal{H}_n$ for every positive integer n. The points in the infinite equilateral triangle grid that lie on the first n concentric hexagons about the origin form the nodes of $\mathcal{H}_n$, and each node is adjacent to six nodes in six directions listing counter-clockwise as E, NE, NW, W, SW, and SE. In [8], it is shown that every node can be addressed using a two-component addressing scheme $\langle x, y\rangle$ where x and y respectively represent the signed distances of a node in E and NE direction from the origin. For example, $\langle 0, y\rangle$ corresponds to the coordinates of the nodes on the ray at an angle of 60 degrees through the origin. Fig. 1 depicts the addressing scheme for the nodes of $\mathcal{H}_3$.

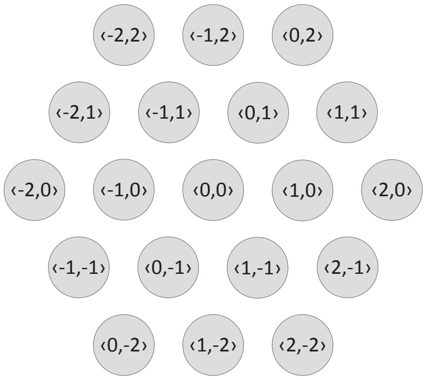


Fig. 1. The node addresses for $\mathcal{H}_3$.

There are two types of edges in $\mathcal{H}_n$: *regular* and *wraparound* edges. Regular edges, which are the sides of triangles in the infinite triangle grid, connect two internal nodes of $\mathcal{H}_n$ whereas wraparound edges connect two boundary nodes of $\mathcal{H}_n$. We next explain how to find the two ends of a wraparound edge using the modulo operation in EJ numbers.

The NE direction in the infinite triangular grid is defined by the complex number $\omega = \frac{1+i\sqrt{3}}{2}$. Thus, $x + y\omega$ uniquely represents the node $\langle x, y\rangle$ in the infinite grid where x and y are integers. The complex number $x + y\omega$ forms a subring of the field of complex numbers called the *Eisenstein-Jacobi* numbers or EJ numbers.

An EJ number $\alpha = a + b\omega$ is used to generate an EJ interconnection network [6], [7] where the nodes of the network are the congruence classes modulo α, and two nodes $x = x_1 + x_2\omega$ and $y = y_1 + y_2\omega$ are adjacent if and only if $y - x$ equals ± 1, $\pm\omega$, or $\pm\omega^2$ modulo α. Therefore, α generates a homogeneous, degree six EJ network which has $|\alpha|^2 = a^2 + ab + b^2$ nodes.

$\mathcal{H}_n$ (the HARTS network) is the special case of the EJ network where the generator is $\alpha = n + (n-1)\omega$ and so has $|\alpha|^2 = 3n^2 - 3n + 1$ nodes. In [8] it is shown that its diameter is $n - 1$ and has the maximal number of nodes for all regular networks of degree six.

We next consider $\mathcal{H}_3$ which is generated by $\alpha = 3 + 2\omega$. The infinite grid can be tiled by triangles of $\mathcal{H}_3$. Three of these tiles are pictured is Fig. 2. The centers of the tiles are $(s + t\omega)\alpha$ for any integer s, t.

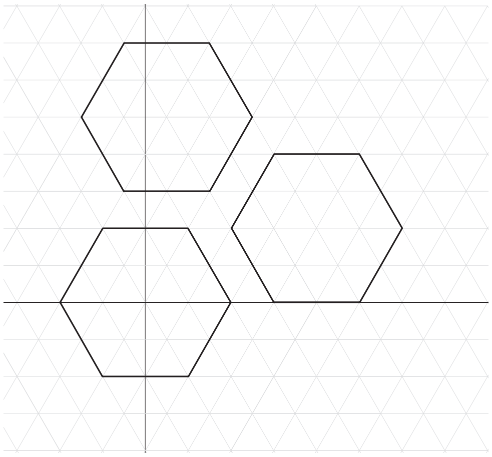


Fig. 2. Three hexagons in the tiling of the plane by hexagons where the centers of the hexagons are $0 = \langle 0, 0\rangle$, $3 + 2\omega = \langle 3, 2\rangle$, and $2\omega^2 + 3\omega = \langle -2, 5\rangle$

Note that, the node $\langle 0, 2\rangle$ has six neighbors in the infinite grid which are listed in clockwise as $\langle -1, 3\rangle$, $\langle 0, 3\rangle$, $\langle 1, 2\rangle$, $\langle 1, 1\rangle$, $\langle 0, 1\rangle$, and $\langle -1, 2\rangle$. The first three neighbors do not lie within $\mathcal{H}_3$. The two nodes $\langle -1, 3\rangle$ and $\langle 0, 3\rangle$ are located in the hexagon centered at $\langle -2, 5\rangle$ and the node $\langle 1, 2\rangle$ is in hexagon centered at $\langle 3, 2\rangle$. By modulo $\alpha = 3 + 2\omega$ operation these hexagons can be translated back to be centered at origin, and the three points correspond to $\langle 1, -2\rangle$, $\langle 2, -2\rangle$, and $\langle -2, 0\rangle$, respectively. Thus, the node $\langle 0, 2\rangle$ is adjacent to the nodes $\langle 1, -2\rangle$, $\langle 2, -2\rangle$, and $\langle -2, 0\rangle$, by wraparound links in $\mathcal{H}_3$.

III. Routing in the Hexagonal Networks

This section presents a new fully adaptive routing algorithm for hexagonal torus networks. The algorithm first finds all the possible shortest paths between any pair of source and destination using the shortest path algorithm given in [9]. Then, the algorithm specifies how to assign virtual channels to the messages to obtain a deadlock-free and fully adaptive routing. First, we briefly review the shortest path algorithm in [9].

As stated earlier in Section II, the two-component addressing scheme $\langle x, y\rangle$ represents a node in a hexagonal torus network where x, y are integers and $x + y\omega$ uniquely specifies the node in the infinite triangle grid. Considering $\omega = \frac{1+i\sqrt{3}}{2}$ we have

$$\omega^2 = \omega{-}1 \; ; \; \omega^3 = -1 \; ; \; \omega^4 = -\omega \; ; \; \omega^5 = 1{-}\omega \; ; \; \omega^6 = 1 = \omega^0$$

Thus, $\omega^0, \omega^1, \omega^2, \omega^3, \omega^4, \omega^5$ respectively define the E, NE, NW, W, SW, and SE directions in the infinite triangle grid. It is shown in [9] that each point in the infinite triangle grid lies in one of the six sectors j ($j = 1..6$) written as $a\omega^{j-1} + b\omega^j$ where a and b are *nonnegative* integers. Fig. 3 depicts the six sectors in the infinite triangle grid.

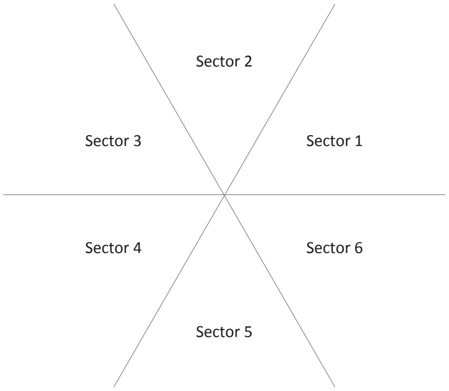
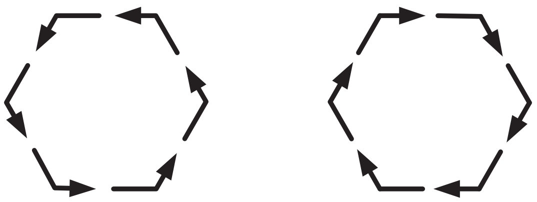


Fig. 3. Sectors in the infinite triangle grid.

There are six types of messages in hexagonal torus network that can be placed in any one of the six sectors. The message routing from S to D first calculates $D - S = x + y\omega = a\omega^{j-1} + b\omega^j \mod \alpha$ where $a, b \geq 0$. Then minimal routing from S to D takes any a edges along direction ω^{j-1} and any b edges along direction ω^j and we refer this as type j message ($j = 1..6$).

In every type j message there are two possible 120 degree turns: the turn from ω^{j-1} to ω^j in the counter-clockwise direction and the other turn from ω^j to ω^{j-1} in the clockwise direction. These turns create cycles in the resource dependence graph [14] as depicted in Fig. 4. The existence of cycles in the resource dependence graph is a necessary condition for causing deadlock in the routing algorithm. Next, we describe how to assign virtual channels in order to avoid cycles in the resource dependence graph which results in a deadlock-free routing algorithm.



(a) Counter-clockwise turns

(b) Clockwise turns

Fig. 4. 12 possible turns in an hexagonal network

A. Fully Adaptive Routing

One way to break the potential cycle shown in Fig. 4 is to add some extra virtual channels for each physical channels. If there is no such cycle within every classes of virtual channels, then there would be no deadlock [14].

Another cause of deadlock which is common in wraparound topologies is the inherent deadlock that happens because of the wraparound links. In order to break those cycles, the proposed fully adaptive routing algorithm utilizes an approach very similar to the dateline classes introduced by Dally and Towles in [14]. That is, regular messages that do not need to take any wraparound link on their way to destination use one class of virtual channel, whereas wraparound messages (message that need to traverse one or more wraparound links) use another class of virtual channel.

The fully adaptive routing algorithm proposed in this section utilizes three virtual channels per physical channel to avoid both inherent cycles and cycles created by the resource dependence graph shown in Fig. 4. The first step of the algorithm is to find all the minimal paths between the source S and the destination D as explained earlier in this section. Once $D - S = x + y\omega \mod \alpha$ is calculated and the type of message is decided, then the virtual channel for each type is assigned based on the TABLE I. Since $x + y\omega = a\omega^{j-1} + b\omega^j \mod \alpha$ $(a, b \geq 0)$, the type j message can fully adaptively take any a edges of direction ω^{j-1} and b edges of direction ω^j.

Type	Regular	Wraparound
1	0	1
2	0	2
3	1	2
4	1	0
5	2	0
6	2	1

TABLE I
VIRTUAL CHANNEL ASSIGNMENT OF THE FULLY ADAPTIVE ROUTING FOR EACH TYPE OF MESSAGE

Before proving that the fully adaptive routing algorithm is deadlock-free, let us consider some routing examples.

Fig. 5 shows two routing examples in $\mathcal{H}_5$ which is a hexagonal network generated by $\alpha = 5 + 4\omega$. The first example is routing from $a = 3$ to $b = 1 - 2\omega$. Here $D - S = -2 - 2\omega = 2\omega^3 + 2\omega^4$ which is located at the hexagonal centered at origin. Using the minimal routing algorithm explained earlier in this section, this message is of type 4. Now, applying TABLE I gives the virtual channel class which is 1 in this case. Therefore, the message can adaptively take the virtual channel class 1 of the ω^3 and ω^4 directions.

The second example is a little bit more complicated since it involves wraparound links. In this example the routing is from $c = -3 + 3\omega$ to $f = -3\omega$. Now, $D - S = 3 - 6\omega$ is not located at the hexagonal centered at origin, but is located in the hexagonal centered at $\omega^4\alpha$. Thus, $\omega^4\alpha$ needs to be subtracted from $D - S$. That is,

$$D - S = 3 - 6\omega - (-9\omega + 4) = -1 + 3\omega = 2\omega + \omega^2$$

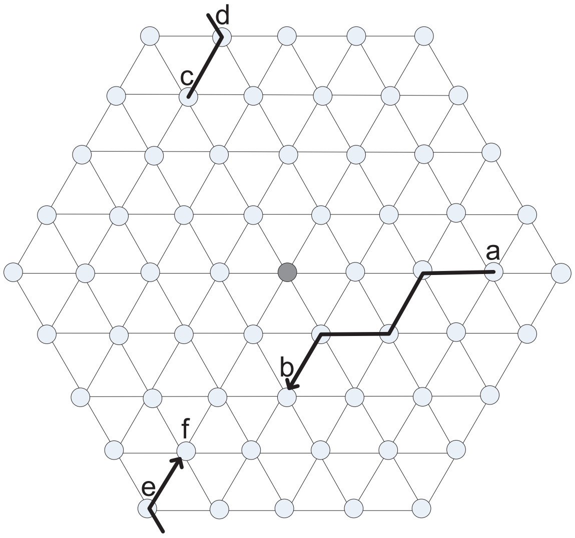


Fig. 5. Two examples of fully adaptive routing in $\mathcal{H}_5$.

Thus the minimal route from c to f takes two ω edges and one ω^2 edges. Then, we use TABLE I to find the virtual channel class. Since the message is type 2 and it is a wraparound message, it uses virtual channel class 2. Now the message can take the virtual channel class 2 of any available ω and ω^2 directions. In this example this message takes one hop along ω direction to reach d. From there it uses the wraparound link in ω^2 direction to reach e, and from e it takes one hop along ω direction to reach to its destination f.

B. Deadlock-freedom Proof of the Fully Adaptive Routing

Now we prove that the fully adaptive routing algorithm proposed in this section is deadlock-free. First, Theorem 1 shows that the potential cycles shown in Fig. 4 cannot cause any deadlock.

Theorem 1. *There exists no cyclic channel dependency on the regular links and wraparound links of the hexagonal torus network when the virtual channel assignment scheme of TABLE I is used.*

Proof: In order to form the cyclic channel dependency of Fig. 4, there need to be six channel dependencies for the six type of messages utilizing the same class of virtual channels. However, TABLE I shows that each class of virtual channel is only used by four different type of messages. For instance, virtual channel class 0 is used by messages of type 1,2,4, and 5. Therefore, the fully adaptive routing algorithm does not create channel dependency of the form of Fig. 4 either among regular links or wraparound links or their combinations. ■

Next, we show that within one type of message there cannot be any cyclic channel dependency caused by inherent wraparound links. The idea that we use here is very similar to the dateline classes introduced by Dally and Towles in [14]. Since the virtual channel usage of regular and wraparound messages are different, there exists no dependency among regular and wraparound links. Regular messages cannot create dependency within one message type because they do not form a cycle. However, wraparound messages are prone to create cycles as depicted in Fig. 6 for type 1 messages. Theorem 2 shows that within one message type there cannot be any cyclic channel dependency caused by wraparound links.

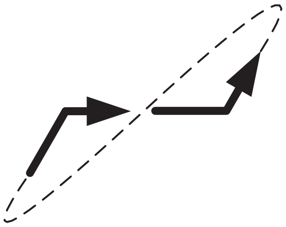


Fig. 6. Potential cyclic channel dependency for type1 messages.

Theorem 2. *There exists no cyclic channel dependency on the wraparound links within one message type.*

Proof: Cycle within one message type starts from one node and ends with the same node. We know that in an EJ network every repeat point is obtained after $n+(n-1)\omega$ hops. Thus, the length of any cycle within one message type is $2n-1$ and the diameter of the network is $n-1$. Consider the cycle shown in Fig. 7. Choosing the counter-clockwise direction, the wraparound link is used only when one of the $n-1$ nodes on the right wants to send a message to one of the $n-1$ nodes on the left provided the distance between them along the right to left is greater than $n-1$. In addition, the nodes at the left of the center node send messages to the nodes to the right of the center node through the center node when the distance between them is less than or equal to $n-1$. These messages use virtual channels different from the virtual channels used by the messages going along the wraparound links. Therefore, no cycle can be created. ■

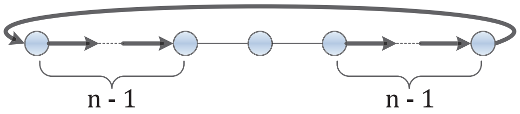


Fig. 7. Acyclic channel dependency within one type of message.

Regular messages of one type along with wraparound messages of a neighboring type is another source of potential deadlock. Every type j message has one direction in common with type $j-1$ message and type $j+1 \pmod 6$. Thus, there are 12 possible scenarios for deadlock. One of them is depicted in Fig. 8. This picture shows that regular messages of type 1 combine with wraparound messages of type 2 can create cycles in the resource dependence graph if they use the same class of virtual channels. Theorem 3 proves that these scenarios cannot cause deadlock.

Theorem 3. *There exists no cyclic channel dependency on regular messages and wraparound messages of two neighboring message types.*

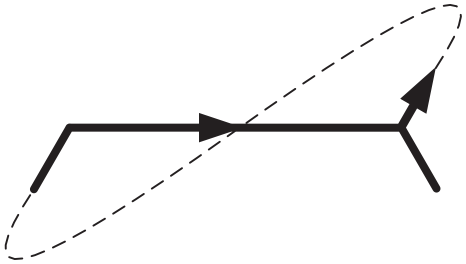


Fig. 8. Potential cyclic channel dependency on two neighboring message types.

Proof: In order to have any cyclic dependency of Fig. 8 regular messages of one type and wraparound messages of a neighboring type must use the same class of virtual channels. However, TABLE I reveals that the virtual channel class used by regular messages is different than those of wraparound messages for every message type. Therefore, there is no dependency between two neighboring message types. ■

From the above Theorems we conclude that the fully adaptive routing algorithm presented in this paper is deadlock-free.

Note that, the proposed fully adaptive routing algorithm utilizes the same class of virtual channels for each direction within one type of message. For example, both ω^0 and ω^1 directions in regular type 1 message use virtual channel class-0. Next we show that under this assumption, there exist no deadlock-free routing algorithm for hexagonal torus network using less than three classes of virtual channels. The argument is as follows.

Let regular messages of type 1 use class-0. Since wraparound messages need to use different class of virtual channels, let wraparound messages of type 1 use class-1. According to Theorem 3 regular messages of type 2 need to use different class than wraparound messages of type 1. If there are only two classes of virtual channels, then type 2 regular messages must use class-0 and type 2 wraparound messages must use class-1. Following this strategy, all regular messages have to use class-0 and all wraparound messages have to use class-1 of virtual channels. Consequently, the cycle in the resource dependence graph of Fig. 4 will form and the algorithm would not be deadlock-free.

IV. Simulation Results

This section presents the performance study of the proposed routing algorithm. We compare the performance of the fully adaptive routing algorithm for hexagonal torus network to fully adaptive routing algorithm for meshes and tori. We picked the Duato's protocol [15](DP) as a fully adaptive routing algorithm for meshes and tori. DP separates the virtual channels into two partitions: escape channels and fully adaptive channels. Messages can take any free virtual channel from the fully adaptive channels. If no such free channel is found, then a deadlock-free and minimal routing algorithm is employed utilizing escape channels. We choose the well-known dimension order routing as a deadlock-free algorithm to go with DP.

In order to compare the performance of the proposed fully adaptive routing algorithm with the corresponding algorithms in meshes and tori, we perform an extensive simulation. The average time from the injection to consumption of messages which is called *average message latency* is the performance metric we used in this simulation study.

Simulation experiments are performed using XMulator [16]. XMulator is a complete, flit-level, event-based, and extensively detailed package for simulation of interconnection networks which can simulate different interconnection networks with arbitrary topology, switching methods, routing algorithms, and even in the presence of faults.

The following assumption has been made in the simulation. No delay for switching and routing is considered and the only considered delay is physical channels delay. Message consumption and generation bandwidth is unlimited. Messages are generated based on exponential distribution and their destination is distributed uniformly. Each physical channel utilizes three virtual channels. Injection channels and ejection channels utilize three virtual channels as well. The delay of physical channel is one cycle and message length is 64 bits. The flow control used in the networks is wormhole flow control [14].

Since we considered three virtual channels for all simulations, the assignment of virtual channels to the virtual channel classes of each algorithm is as follows. The proposed fully adaptive routing algorithm for hexagonal torus uses one virtual channel for each of its classes of virtual channels. The fully adaptive routing algorithm for mesh/torus utilizes one/two virtual channel(s) as the escape channels and two/one virtual channel(s) as the adaptive channels.

We choose two different groups of network sizes for the simulations. The small-size group consists of 8×8 $2D$ mesh/torus and $4 \times 4 \times 4$ $3D$ mesh/torus with 64 nodes, and $\mathcal{H}_5$ with 61 nodes. The large-size group includes 16×16 $2D$ mesh/torus with 256 nodes, $7 \times 7 \times 7$ $3D$ mesh/torus with 343 nodes, and $\mathcal{H}_{10}$ with 271 nodes. Since hexagonal network is of degree six we extend the simulation to degree six mesh/torus.

Fig. 9 shows the average message latency of the routing algorithms for three networks as a function of message generation rate for the small-size group of network. As it can be seen, the proposed fully adaptive routing algorithm for hexagonal torus outperforms the fully adaptive routing algorithm of a $2D$ mesh/torus of the same size.

The same scenario is run for the large-size network and the result is depicted in Fig. 10. In this case, the proposed routing algorithm is far better than the fully adaptive routing in $2D$ mesh/torus. This is mainly due to the small diameter of the hexagonal torus compare to the diameter of $2D$ mesh and torus. As the network size increases the diameter of torus network grows slower than the diameter of $2D$ mesh and torus.

In the simulations, the performance of the fully adaptive routing in $3D$ tori is superior to that of hexagonal network. This is reasonable, since the diameter of $3D$ torus is less than the diameter of hexagonal torus network. Moreover, the proposed fully adaptive routing algorithm outperforms the

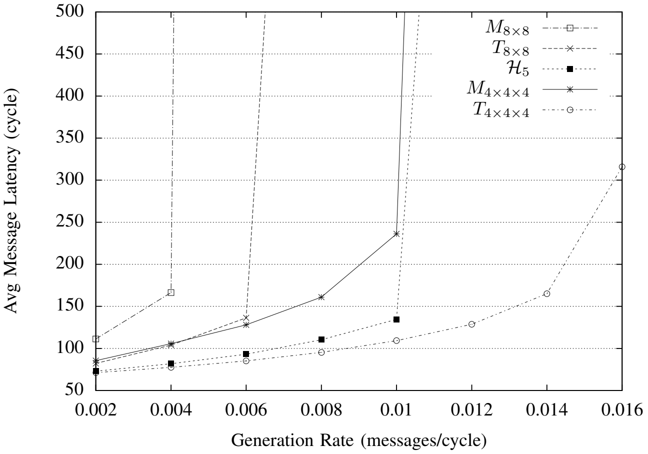


Fig. 9. Fully adaptive routing for small-size network

fully adaptive routing in $3D$ mesh in the small-size network and provides similar performance in the large-size network.

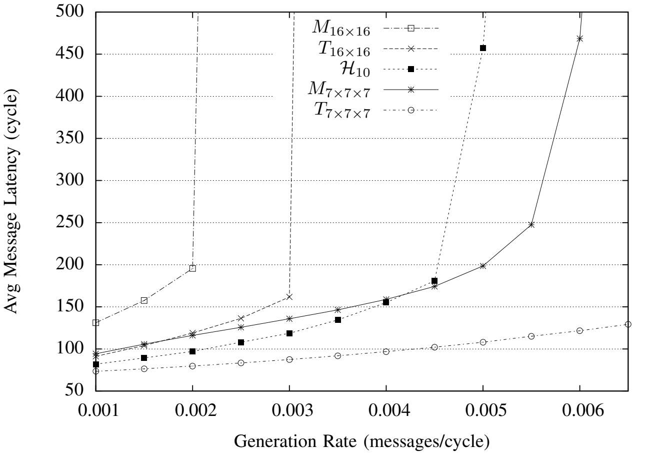


Fig. 10. Fully adaptive routing for large-size network

V. Conclusion

This paper proposes a minimal, fully adaptive and deadlock-free routing algorithm for hexagonal torus interconnection network. By carefully assigning virtual channels to the message types, the proposed routing algorithm remains deadlock-free. The algorithm requires three virtual channels per physical channel. In addition, since the algorithm is minimal, no livelock can happen in the network.

Simulation results showed that the fully adaptive routing algorithm for hexagonal torus outperforms fully adaptive routing for $2D$ meshes and $2D$ tori of the same network size. This performance improvement is more remarkable as the network size increases.

As the number of nodes increases in an interconnection network the probability of node and link failure also increases. Thus, as a future direction we are looking to propose fault tolerant routing algorithms for hexagonal torus network. A fault tolerant routing would be able to deliver messages to their destinations in the presence of some faulty nodes/links in the network.

Acknowledgment

This work has been supported by the National Science Foundation grant CCF-1015804.

References

[1] N. Adiga et. al., "An overview of the BlueGene/L supercomputer," in *Supercomputing, ACM/IEEE 2002 Conference*, Nov. 2002, p. 60.

[2] S. L. Scott et. al., *The Cray T3E Network: Adaptive Routing in a High Performance 3D Torus*, 1996.

[3] Z. Cvetanovic, "Performance analysis of the alpha 21364-based HP GS1280 multiprocessor," *SIGARCH Comput. Archit. News*, vol. 31, no. 2, p. 218229, May 2003. [Online]. Available: http://doi.acm.org/10.1145/871656.859643

[4] M. D. Noakes, D. A. Wallach, and W. J. Dally, "The j-machine multicomputer: an architectural evaluation," *SIGARCH Comput. Archit. News*, vol. 21, no. 2, p. 224235, May 1993. [Online]. Available: http://doi.acm.org/10.1145/173682.165158

[5] R. Alverson, D. Callahan, D. Cummings, B. Koblenz, A. Porterfield, and B. Smith, "The tera computer system," *SIGARCH Comput. Archit. News*, vol. 18, no. 3b, p. 16, Jun. 1990. [Online]. Available: http://doi.acm.org/10.1145/255129.255132

[6] K. Huber, "Codes over eisenstein-jacobi integers," in *Finite Fields: Theory, Applications, and Algorithms*. American Mathematical Soc., 1994, vol. 168, pp. 165–179.

[7] C. Martnez, E. Stafford, R. Beivide, and E. Gabidulin, "Modeling hexagonal constellations with eisenstein-jacobi graphs," *Problems of Information Transmission*, vol. 44, no. 1, pp. 1–11, 2008. [Online]. Available: http://www.springerlink.com/content/40v2r2355r6x0m30/abstract/

[8] M. Flahive and B. Bose, "The topology of gaussian and eisenstein-jacobi interconnection networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 8, pp. 1132–1142, Aug. 2010.

[9] B. Albader, B. Bose, and M. Flahive, "Efficient communication algorithms in hexagonal mesh interconnection networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 69–77, Jan. 2012.

[10] D. D. Kandlur and K. G. Shin, "Reliable broadcast algorithms for HARTS," *ACM Trans. Comput. Syst.*, vol. 9, no. 4, p. 374398, Nov. 1991. [Online]. Available: http://doi.acm.org/10.1145/118544.119233

[11] M.-S. Chen, K. Shin, and D. Kandlur, "Addressing, routing, and broadcasting in hexagonal mesh multiprocessors," *IEEE Transactions on Computers*, vol. 39, no. 1, pp. 10 –18, Jan. 1990.

[12] J. Dolter, P. Ramanathan, and K. Shin, "Performance analysis of virtual cut-through switching in HARTS: a hexagonal mesh multicomputer," *IEEE Transactions on Computers*, vol. 40, no. 6, pp. 669 –680, Jun. 1991.

[13] F. Garcia Nocetti, I. Stojmenovic, and J. Zhang, "Addressing and routing in hexagonal networks with applications for tracking mobile users and connection rerouting in cellular networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 9, pp. 963 – 971, Sep. 2002.

[14] W. J. Dally and B. Towles, *Principles and practices of interconnection networks*. Morgan Kaufmann, 2004.

[15] J. Duato, "A new theory of deadlock-free adaptive routing in wormhole networks," 1993. [Online]. Available: http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.53.4729

[16] A. Nayebi, S. Meraji, A. Shamaei, and H. Sarbazi-Azad, "XMulator: a listener-based integrated simulation platform for interconnection networks," in *First Asia International Conference on Modelling Simulation, 2007. AMS '07*, Mar. 2007, pp. 128 –132.