Routing Algorithms for Torus Networks

Jatin Upadhyay, Vara Varavithya, and Prasant Mohapatra
Department of Electrical and Computer Engineering
Iowa State University
Ames, IA 50011
E-mail: *prasant@iastate.edu*

Abstract

Routing in mesh networks is studied in great detail in the literature. Interest in torus networks however is steadily increasing due to its symmetry in traffic distribution. In this paper, we present a new concept of *virtual mesh network* to map the torus interconnection networks on to the mesh networks. Using this concept, we show how the routing algorithms for meshes can be applied to the torus networks with the use of minimum extra hardware. The concept is illustrated by an example of the fully adaptive PFNF algorithm. Although all the discussion assumes two dimensional mesh and torus networks, the idea is applicable to multi- dimensional networks.

Key words: Message routing, PFNF algorithm, Torus networks, Virtual mesh network, Wormhole routing.

1 Introduction

In distributed parallel computers, tasks are executed by a set of interconnecting nodes or processors. The performance of the interprocessor communication depends on the selection of the interconnection network switching technique, the routing algorithm.

Interconnection network refers to the physical connection between the processors or the nodes in the system. Since direct networks utilize the locality of the message references more efficiently, most of the existing systems use direct networks such as k-ary n-cubes or n-dimensional meshes. Performance analysis of direct networks have shown that with wormhole switching technique, lower dimensional networks offer improved latency and throughput results for the same network bandwidth [1, 7]. Recent interest in multicomputer systems is therefore concentrated on 2 and 3 dimensional mesh and torus networks [2, 3, 11, 12, 13].

Switching technique determines the allocation of resources as the message travels through the network. Virtual cut-through and wormhole routing are the main switching techniques used in multicomputer systems. Amongst the three, virtual cut-through and wormhole routing result in lower latency than store and forward routing. Due to its lower buffer requirements, wormhole routing is preferred over virtual cut through, and is being used in serveral contemporary multicomputer systems [15].

Routing algorithm determines the path a message follows to reach its destination. If the path between every pair of source and destination is fixed, the algorithm is called a *deterministic algorithm.* For better system performance, it is however preferable that the algorithm adapt itself to the network faults and traffic congestion and allow alternate paths. Depending upon whether the algorithm can use all the possible physical paths between the source and the destination, adaptive algorithms are classified as *partially adaptive* or *fully adaptive.* Turn model[10], direction restriction model [5] and planar routing [6] are examples of partially adaptive algorithms. Fully adaptive algorithms are developed using the concept of virtual channels[8]. Examples of fully adaptive routing algorithms include [4, 9, 14, 16, 17]. All these algorithms differ from one another in terms of their virtual channel requirements and the efficiency they provide.

Mesh interconnection networks are simple and easily scalable. However, they are not symmetric at the edges which causes uneven traffic distribution in the network even though the generated traffic is uniform [17]. This uneven distribution limits its performance. Many research and commercial systems therefore use the torus networks instead. The wrap-around connections in the torus make scaling of the network difficult. But better performance is delivered by the torus network because of its symmetricity throughout the whole network. The recent commercial system Cray T3D [12] uses a torus interconnection topology.

In this paper, we present a new concept of *virtual mesh network* to map the torus interconnection networks to the mesh interconnection networks. Since routing in mesh is widely studied[10, 5, 6, 16, 4, 17], the concept of virtual mesh networks helps in addressing the torus routing issues in a simpler and methodological manner. In most cases, the routing algorithms for meshes can be easily extended or modified for use in torus networks. We illustrate this concept by giving example of the fully adaptive PFNF[17] routing algorithm. Throughout the paper we have assumed wormhole switching and two dimensional networks. The idea is, however, easily applicable to other switching techniques and higher dimensions.

The rest of the paper is organized as follows. Section 2 summarizes the required definitions. The concept of virtual mesh networks is discussed in Section 3. Section 4 illustrates the concept with an example of the PFNF algorithm. Conclusions are presented in Section 5.

2 Definitions

This section presents the terminologies used throughout the paper. Some of these definitions are reiterated from previous works [9, 4, 7] for the sake of completeness.

Definition 1: An *n-dimensional torus* is defined as an interconnection structure that has $K_0 \times K_1 \times \ldots \times K_{n-1}$ nodes with n as the number of dimensions and K_i as the number of nodes in ith dimension. Each node in the torus is identified by an n-coordinate vector $(x_0, x_1, \ldots, x_{n-1})$, where $0 \leq x_i \leq K_i - 1$. Two nodes, $(x_0, x_1, \ldots, x_{n-1})$ and $(y_0, y_1, \ldots, y_{n-1})$, are connected if and only if there exists an i such that $x_i = (y_i \pm 1) \bmod K_i$, and $x_j = y_j$, for all $j \neq i$.

Definition 2: A *Physical Interconnection Network*, PN, is a strongly connected graph, $PN(PV, PC)$, where PV represents the set of processing nodes and PC represents the set of physical channels connecting the nodes.

Definition 3: A *Virtual Interconnection Network*, VN, is a strongly connected graph, $VN(PV, VC)$, where PV represents the set of processing nodes and VC represents the set of virtual channels, that are mapped to the set of physical channels PV.

Definition 4: A Channel along dimension i is termed as *positive* channel if its source node $X(x_0, x_1, \ldots, x_{n-1})$ and sink node $Y(y_0, y_1, \ldots, y_{n-1})$ differ in the ith coordinate such that $x_i = y_i - 1 \bmod K_i$.

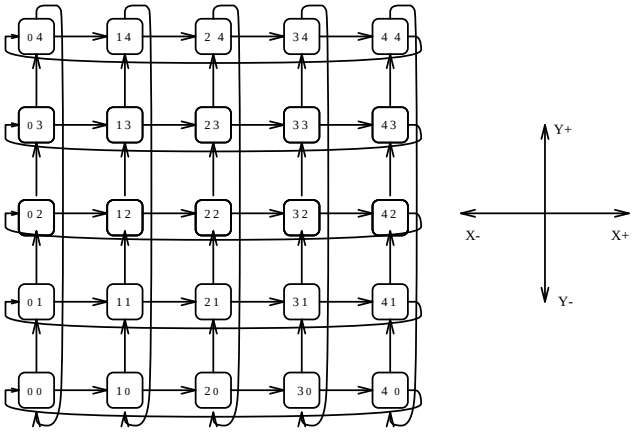


Figure 1: A 5×5 two-dimensional unidirectional torus.

Definition 5: A Channel along dimension i is termed as *negative* channel if its source node $X(x_0, x_1, \ldots, x_{n-1})$ and sink node $Y(y_0, y_1, \ldots, y_{n-1})$ differ in the ith coordinate such that $x_i = y_i + 1 \bmod K_i$.

Definition 6: An n-dimensional torus is defined as *unidirectional* if the nodes connected in the network are connected either by all the positive or all the negative channels.

Definition 7: An n-dimensional torus is defined as *bidirectional* if the nodes connected in the network are all connected both by the positive and the negative channels.

Definition 8: A *routing algorithm* $R : N \times N \rightarrow \rho(C)$, where $\rho(C)$ is the power set of C, supplies a set of alternative output channels to send a message from current node n_c to the destination node n_d. $R(n_c, n_d) = (c_1, c_2, \ldots, c_p)$.

Figure 1 shows a 5×5 two-dimensional unidirectional torus network. The nodes are all connected by the positive channels. The figure also shows the directional notations used throughout this paper.

3 Virtual Mesh Network

In wormhole switching, a message is split into a number of small *flits*. Only the header flit has the address of the destination and all the other flits follow the header flit in a pipelined fashion. Since no trailing flits has any information about the destination, if the header is blocked at some node, the whole message gets blocked. If the blocked messages in the network form a cycle, it can create a deadlock.

In mesh interconnection networks, cyclic dependency can occur due to the inter-dimensional turns made by the messages. All the possible turns a message can make are shown in Figure 2(a). The cycles formed in Figure 2(a) also refer to the deadlock configurations.

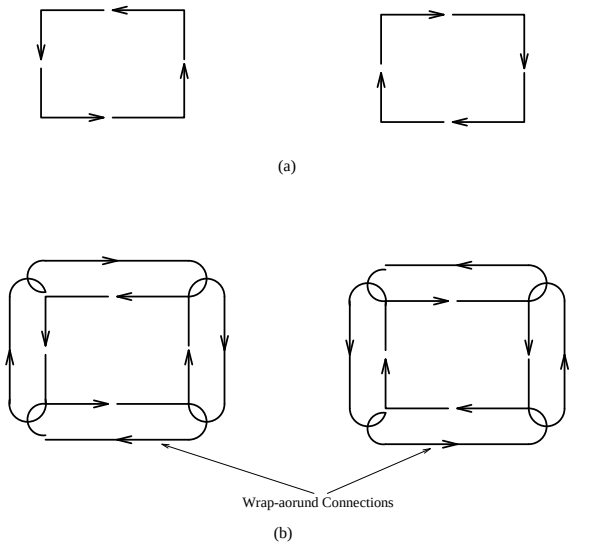


Figure 2: Deadlock configurations in (a) mesh and (b) torus networks.

In torus networks, since there are wrap-around connections (refer to Figure 1), besides the inter-dimensional turns, cyclic dependency can also occur in the same dimension. The possible deadlock configurations for a torus are shown in Figure 2(b).

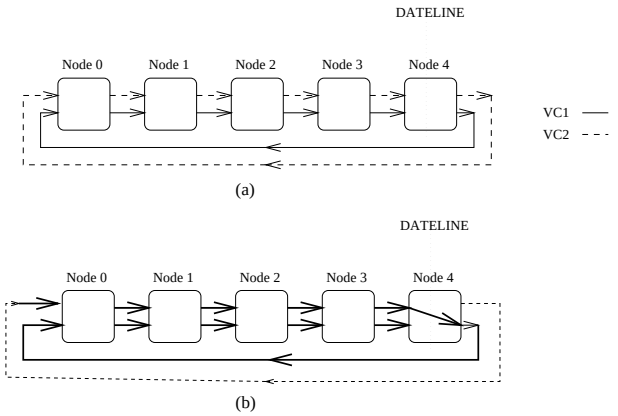


Figure 3: Avoiding Wrap-around dependency

Since deadlock situations in meshes are prevented by the routing algorithms, if the wrap-around dependencies are some how broken, routing algorithms for mesh networks can be efficiently applied to torus networks. The concept of *virtual mesh network* is introduced to enable this mapping. It defines the additional hardware and routing constraints required to break the wrap-around dependencies and therefore the mapping of torus interconnection networks to the corresponding mesh interconnection networks.

3.1 Breaking wrap-around dependency

Wrap-around dependencies can be broken by using two virtual channels per physical channel in each direction and enforcing a rule that messages should travel only on one virtual channel before it crosses one particular node and on the other virtual channel after crossing that node. Consider only one row of nodes as shown in Figure 3 and routing through the positive x direction.

Enforcing the rule that all the messages traveling positive x would travel on virtual channel $VC1$ before they cross node 4 and on virtual channel $VC2$ after they cross node 4, there can not be any cyclic dependency due to the wrap-around connection. The same concept can be extended to the whole torus network and to all the dimensions. We call the set of nodes which force a transition of channels to form a *dateline*. Determination of the nodes that would require these additional virtual channels would depend on the selection of the dateline nodes and on the torus topology (unidirectional or bidirectional).

The dateline and the extra virtual channels required for the purpose of breaking wrap-around dependencies can be considered to form a different layer of the network. We term this network as a *virtual mesh network*. Next subsections describe the virtual mesh networks for two-dimensional unidirectional and bidirectional torus networks.

3.2 Bidirectional Torus

Bidirectional networks have both positive and negative links between all the connected nodes. Figure 4(a) shows a bidirectional torus with two virtual channels $VC1$ ad $VC2$ per physical channel. The wrap-around connections are not shown. If we consider routing only through positive y direction and assume that the dateline in positive y direction is at row 4 as shown in the figure, then only nodes at rows 0 to 2 would require two virtual channels to break the cyclic dependency in positive y direction. this is because in minimal routing, no message would cross row 2 in positive y direction after taking the wrap-around. If the message destination is anywhere above row 2, going in

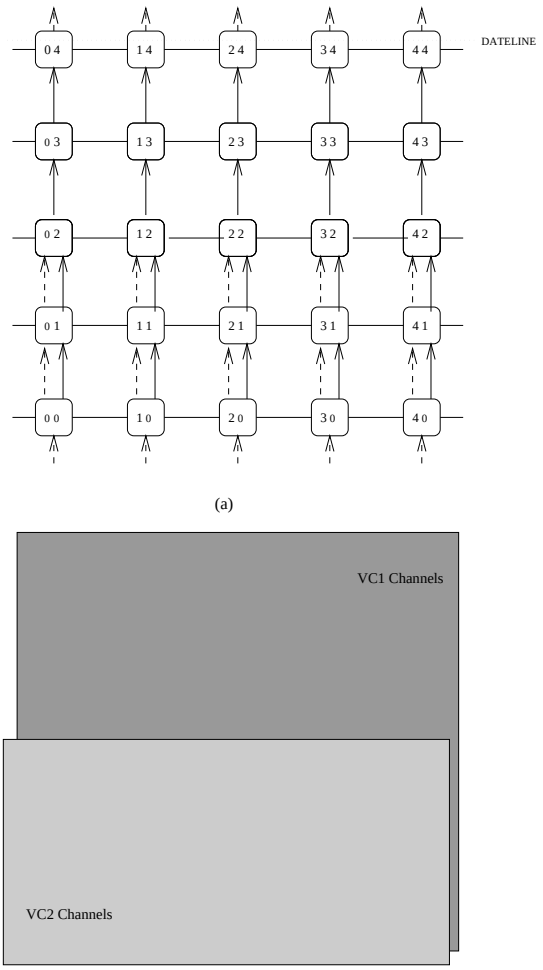


Figure 4: Virtual mesh networks for a bidirectional torus with dateline at Row 4.

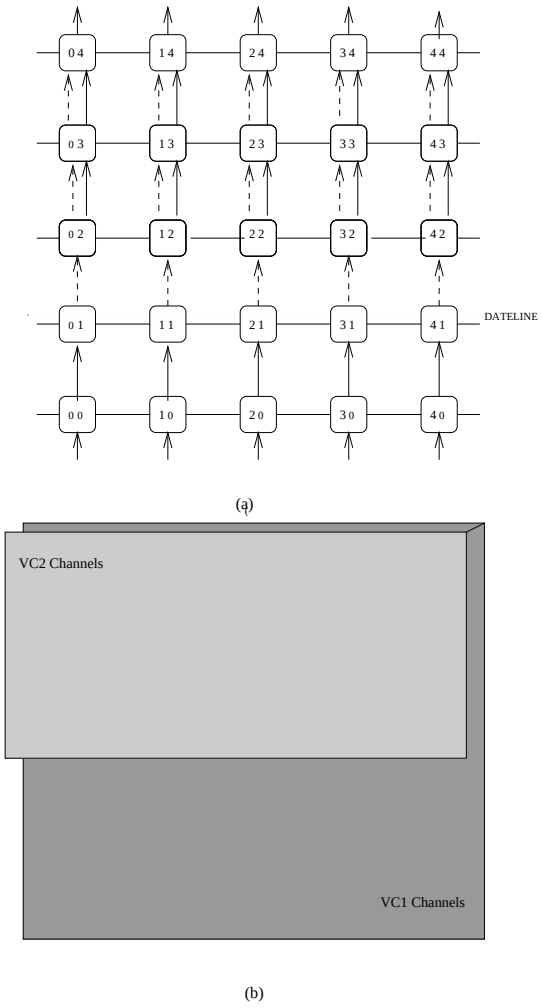


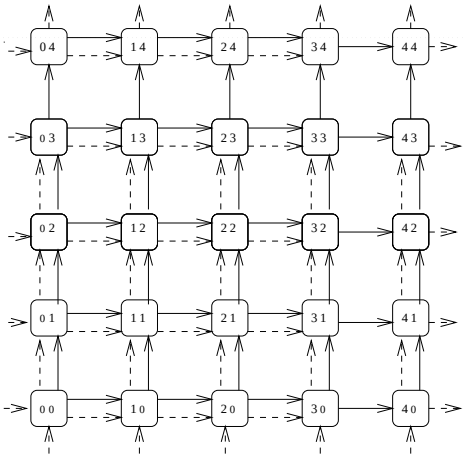
Figure 5: Virtual mesh networks for a bidirectional torus with dateline at Row 1.

negative y direction initially itself would be a shortest path. The virtual mesh network for bidirectional torus would thus require only half of its nodes to have two virtual channels per direction. Figure 4(b) shows the virtual mesh networks for the dateline considered.

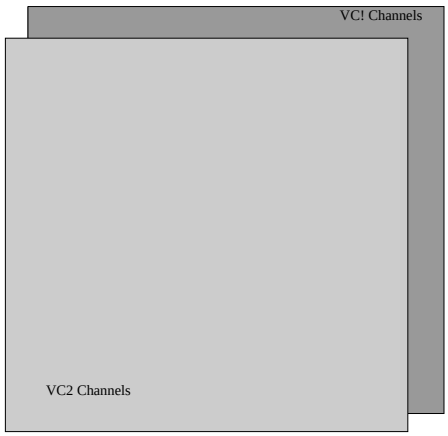
Note again that the position of the dateline determines which nodes would require additional virtual channels in a given direction. If we choose dateline at row 1 instead of at row 4, only nodes at row 2 and above will require two virtual channels. The virtual mesh network in positive y direction would then be as shown in Figure 5.

3.3 Unidirectional Torus

Unidirectional networks are connected only by either positive or negative links. Figure 6(a) shows a 5×5 unidirectional torus network with positive links. Consider again, routing through positive y direction and dateline at row 4, i.e., a message can route on VC1 until it crosses row 4, after which it uses virtual channel VC2. Since the torus has connectivity only through positive links, all the nodes would require two virtual channels for full connectivity under this dateline condition. The virtual mesh network can be visualized as a superposition of two virtual networks comprising VC1 channels and VC2 channels as shown in Figure 6(b).



(a)



(b)

Figure 6: Virtual mesh networks for a unidirectional torus.

4 Mapping PFNF algorithm to Torus - An Example

In this section we show how the concept of virtual mesh network can be used to map the algorithms for mesh to the torus networks. This is illustrated by the example of the fully adaptive algorithm PFNF [17] proposed for the mesh networks.

Note that the virtual mesh network describes how a mesh network - a physical network or a virtual network translates to the corresponding configuration in the torus interconnection. As shown in the previous section for example, one virtual network in the unidirectional mesh would translate to two virtual networks in the torus network and one virtual network in the bidirectional mesh would translate to 3/2 virtual networks in the corresponding torus. When mapping the mesh algorithms to torus the main issue is, if the algorithm for mesh involves more than one virtual networks, how to choose the corresponding datelines so as to minimize the overall hardware requirements in the torus.

Consider the example of the PFNF algorithm in bidirectional mesh network. The algorithm requires two virtual channels (i.e. two virtual networks). Routing is positive first in one virtual networks and negative first in the other. While mapping this algorithm to torus, each virtual network would require 3/2 virtual networks in torus. The overall virtual channels can be minimized by choosing the datelines in these two networks halfway across from each other as shown in the Figures 4 and 5. Thus the implementation of PFNF in torus would then require totally three virtual channels per physical channel (Figure 7.

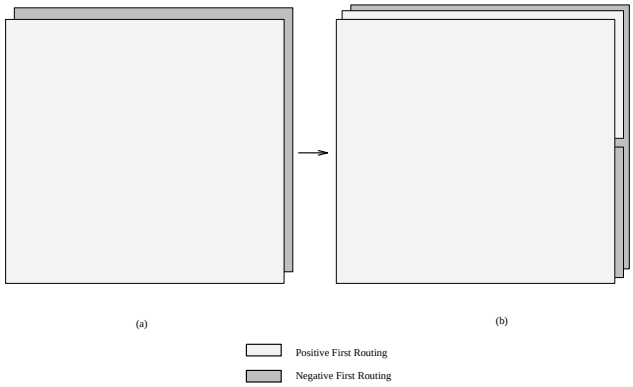


Figure 7: PFNF algorithm mapped from mesh(a) to torus (b).

With a proper choice of datelines, all the algorithms for mesh can be mapped to torus in a methodical way as described for the PFNF algorithm.

5 Conclusions

In this paper, we have presented a new way of looking at the torus networks by visualizing them as a mesh topology with few additional routing constraints. This is shown by the concept of *virtual mesh network*. The concept defines the minimum hardware and conditions required to break the wrap-around dependencies and thereby maps the torus interconnection networks to the mesh networks. Since routing in mesh has been studied in great detail, it presents an elegant way to map the algorithms for mesh networks onto the torus networks. An example mapping of the fully adaptive PFNF algorithm is shown. The same

methodology can be used for other routing algorithms and for higher dimensional k-ary n-cubes.

References

[1] A. Agrawal, "Limits on Interconnection Network Performance," *IEEE Trans. on Parallel and Distributed Systems*, vol. 2, no. 4, pp. 398-412, Oct. 1991.

[2] A. Agrawal, B. Lim, D. Kranz,, and J. Kubiatowicz, "APRIL: A Processor Architecture for Multiprocessing," *Proc. of the 17th Annual Intl. Symposium on Computer Architecture*, vol. 18, no. 2, pp. 104-114, June 1990.

[3] R. Alverson *et al.*, "The Tera Computer System," *Proc. of the 1990 Intl. Conference on Supercomputing*, pp.1-6, June 1990.

[4] Y. M. Boura and C. R. Das, "Efficient fully adaptive wormhole routing in n-dimensional Meshes," *Proc. of the 14th Intl. Conference on Distributed Computing Systems*, 1994.

[5] Y. M. Boura and C. R. Das, "A class of partially adaptive routing algorithms for n-dimensional meshes," *Proc. of the 23rd Intl. Conference on Parallel Processing*, vol. 3, pp. 175-182, August, 1993.

[6] A. A. Chien and J. H. Kim, "Planar Adaptive Routing: Low-cost adaptive networks for multiprocessors," *Proc. of the Intl. symposium on Computer Architecture*, pp. 268-277, May 1992.

[7] W. J. Dally, "Performance Analysis of k-ary n-cube Interconnection Networks," *IEEE Trans. on Computers*, vol. 39, no. 6, pp. 775-785, June 1990.

[8] W. J. Dally., "Virtual channel flow control," *IEEE Trans. on Parallel and Distributed Systems*, vol 3, pp. 194-205, March 1992.

[9] J. Duato, "A New Theory of Deadlock-Free Adaptive Routing in Wormhole Network," *IEEE Trans. on Parallel and Distributed systems*, vol. 4, No.12, Dec. 1993.

[10] C. J. Glass and L. M. Ni, "The Turn model for adaptive routing," *Journal of the ACM*, vol. 41, no. 5, pp. 874-902, Sep. 1994.

[11] Intel Corporation, *A Touchstone DELTA System Description*, 1990.

[12] R. E. Kessler and J. L. Schwarzmeier, "CRAY T3D: A New Dimension for Cray Research," *Compcon*, pp. 176-182, Spring 1993.

[13] D. Lenoski, J. Laudon, K. Gharachorloo, W. Weber, A. Gupta, J. Hennesy, M. Horowitz, and M. Lam, "The Stanford DASH Multiprocessor," *IEEE Computer*, pp. 63-79, March 1992.

[14] D. H. Linder and J. C. Harden, "An Adaptive and fault tolerant wormhole routing strategy for k-ary n cubes," *IEEE Trans. on Computers*, vol. 40, pp. 2-12, Jan. 1991.

[15] P. Mohapatra, "Wormhole Routing Techniques in Mulicomputer Systems," Technical Report, Department of Electrical and Computer Engineering, Iowa State University, 1995.

[16] C. Su and K. G. Shin, "Adaptive deadlock-free routing in multicomputers using only one extra channel," *Proc. of the 22nd Intl Conference on Parallel Processing*, vol. 3, pp. 175-182, Aug. 1993.

[17] J. Upadhyay, V. Varavithya, and P. Mohapatra, "Efficient and Balanced Routing in Two-Dimensional Meshes," *Proc. of the First Intl. Symposium on High Performance Computer Architecture*, pp. 112-122, Jan. 1995.