Deadlock-Free Multi-Path Routing for Torus-Based NoCs[1]

Yaoting Jiao[1], Mei Yang[2], Yulu Yang[1], Yingtao Jiang[2], Xiaochun Yun[3]

[1]College of Information Technology and Science, Nankai University, China
[2] Dept. of Electrical and Computer Engineering, University of Nevada, Las Vegas, USA
[3]Institute of Computing Technology, Chinese Academy, Beijing, P. R. China
Emails: [1]yaoting_jiao@hotmail.com, yangyl@nankai.edu.cn, [2]{meiyang, yingtao}@egr.unlv.edu

Abstract

In our previous work, a Multi-Path Routing (MPR) scheme was proposed to maximize the data throughput for torus-based NoCs by utilizing multiple paths for concurrent data transmission. In this paper, a deadlock-free virtual channel model is proposed for the MPR scheme. In this virtual channel model, every physical channel on the network is split into about 3.5 virtual channels on average. It is proved that any minimal routing algorithm (including the MPR scheme) using this model is deadlock-free. The MPR scheme employing this new virtual channel model is still a fully adaptive one. The performance of the MPR scheme using the proposed virtual channel model is evaluated through simulations and compared with the fully adaptive Single-Path minimal Routing (SPR) scheme with the same virtual channel model. Simulation results show that MPR achieves better average message latency and normalized accepted traffic than SPR under both uniform and nonuniform traffic in general.

1. Introduction

In Networks-on-Chips (NoCs) designs [1][8], crosstalk noise [11][13] has become a serious issue which may cause the communication channel unreliable. The crosstalk problem can be mitigated by wide spacing of serial lines [8]. However, the wider spacing of serial lines will reduce the number of the lines, thus reduce the data throughput. In [9], an innovative Multi-Path Routing (MPR) scheme is proposed for mesh/torus-based NoCs [5][10] to maximize the data throughput by utilizing multiple paths for concurrent data transmission. For the proposed MPR algorithm, two transport models are considered: the Full-wire-bank transport Model (FM) and the Half-wire-bank transport Model (HM) [9]. Theoretical analysis shows that, compared with the single-path routing algorithm, the MPR scheme under FM achieves much higher data throughput and the MPR scheme under HM retains the data throughput while the crosstalk is reduced [3] for single-source situations (i.e., when single pair of source and destination nodes are in communication) [9].

Deadlock-freeness is an important and desirable property of a routing scheme designed for interconnection networks [4]. In this paper, a virtual channel model for the MPR scheme is proposed to ensure that the MPR scheme is deadlock-free and fully adaptive. It is proved that any minimal routing algorithm (including the MPR scheme) using the proposed virtual channel model is deadlock-free. Simulations have been conducted to evaluate the performance of the MPR scheme under the proposed virtual model for multi-source situations (i.e., when multiple pairs of source and destination nodes are in communication).

The rest of the paper is organized as follows. Section 2 describes the node model and the MPR scheme in brief. Section 3 presents the virtual channel model and the proof that all minimal routing algorithms using this model are deadlock-free. Section 4 presents and discusses the simulation results of the MPR scheme and the single-path routing scheme using this virtual channel model. Section 5 concludes the paper.

2. Node Model and MPR Scheme

Without loss of generality, a torus-based NoCs with $2N$x$2N$ processing units (referred as nodes in later text) is considered in this paper. Each node is composed of a processor and a router which connects the processor to the interconnection network. For simplicity, a node is represented as a square in all figures. And all nodes are represented as a $2N$x$2N$ matrix, where each node is indexed with a pair of coordinates (x, y), $0 \le x \le 2N-1$ and $0 \le y \le 2N-1$, on the X and Y dimensions, respectively. A node which has either 0 or $2N$-1 in one of its index number (x or y) is called a *boundary node.*

Each node in the NoCs has four physical channels, each connecting to a neighbor node. Fig. 1 shows the directions of the four channels. Each physical channel may be split into several virtual channels [4].

[1]This work is in part supported by NSF under grant no. ECCS-0702168.

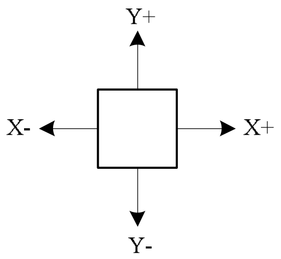


Fig. 1 Directions of the four channels.

The basic idea of the MPR scheme is to divide the data message to be sent into several data streams and send them on different shortest paths concurrently [9]. At the source node, the number of shortest paths (corresponding to the number of data streams that can be sent out) is determined based on the difference between the indexes of the source node and the destination node. At the same time, the control information represented in control bits is also determined and added to the data streams.

At each intermediate node, once receiving a data stream, it will decide which output port the data stream will be forwarded to by checking the difference between the indexes of the current node and the destination node and the control bits. The control bits may be updated according to the routing choice made. It is guaranteed that all routing choices follow the shortest paths. And all the possible shortest paths from the current node to the destination node are included in these choices. Hence, the MPR routing scheme is a fully adaptive one.

As discussed in [9], an important property of the MPR scheme is that it is block-free when single pair of nodes are in communication and all the data streams on the network are from the same message.

When multiple pairs of nodes are in communication, there may exist blockings in the network. A *deadlock* occurs when some data streams are requesting buffers (associated with virtual channels) held by other data streams while holding buffers requested by other data streams [6]. All the data streams involved in a deadlock configuration are blocked forever. The deadlocks can be avoided by providing sufficient number of virtual channels. In the next section, a new virtual channel model is presented which ensures the deadlock-freeness for any minimal routing scheme.

3. Deadlock-Free Virtual Channel Model

3.1 Virtual Channel Model

The physical channels on the torus network are split into virtual channels in the following way. Firstly, it's assumed that every physical channel on the X dimension is split into two virtual channels and every physical channel on the Y dimension is split into four virtual channels. All these virtual channels are divided into six groups of virtual channels as shown in Fig. 2. On average, every physical channel on the network is split into about 3.5 virtual channels. We denote a virtual channel from node U to node V as $C_{U,V,n,m}$, where m represents its channel no. and n represents the virtual channel group no.

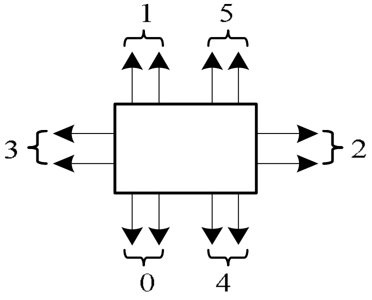


Fig. 2 Groups of virtual channels.

Secondly, additional virtual channels are added according to the following rules. For any node (x, y),

if $x \in [1, N\text{-}1]$, $y \in [0, N\text{-}2]$, an additional virtual channel is added to *Group* 0;

if $x \in [1, N\text{-}1]$, $y \in [N\text{+}1, 2N\text{-}1]$, an additional virtual channel is added to *Group* 1;

if $x \in [0, N\text{-}2]$, $y \in [1, 2N\text{-}2]$, an additional virtual channel is added to *Group* 2;

if $x \in [N\text{+}1, 2N\text{-}1]$, $y \in [1, 2N\text{-}2]$, an additional virtual channel is added to *Group* 3;

if $x \in [N, 2N\text{-}2]$, $y \in [0, N\text{-}2]$, an additional virtual channel is added to *Group* 4;

if $x \in [N, 2N\text{-}2]$, $y \in [N\text{+}1, 2N\text{-}1]$, an additional virtual channel is added to *Group* 5.

The number m in $C_{U,V,n,m}$ of all the additional virtual channels is 2.

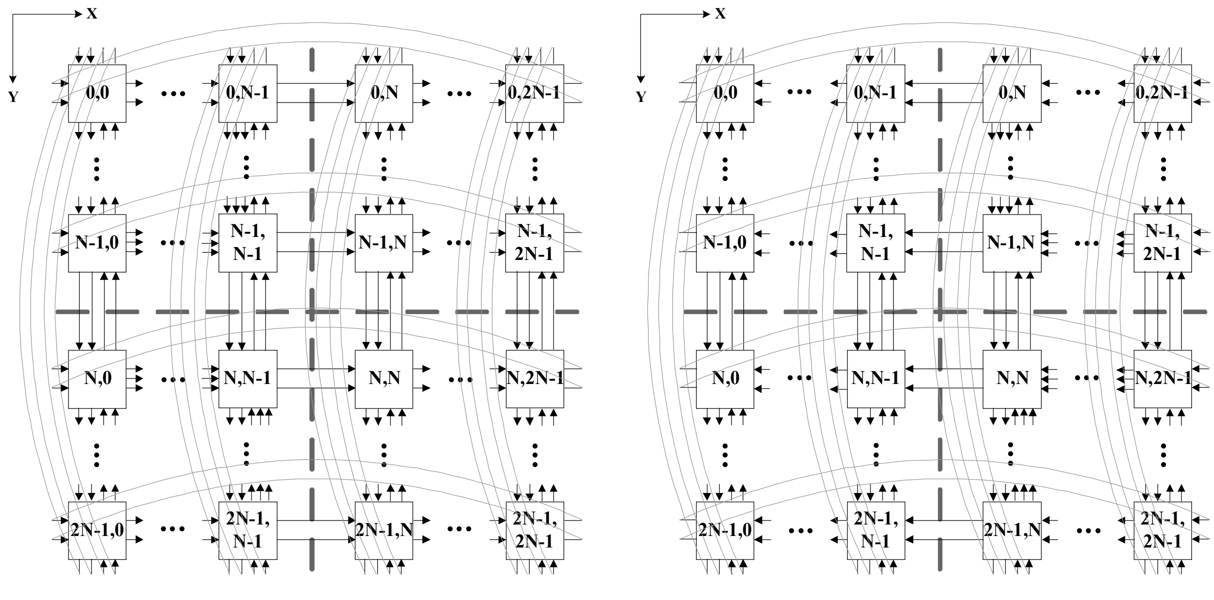
These virtual channel groups are divided into two virtual networks [11], where (virtual channel) *Groups* 0, 1, 2 compose *Network* 0 and *Groups* 3, 4, 5 compose *Network* 1. As shown in Fig. 3, there's no virtual channel pointing to left in *Network* 0, and there's no virtual channel pointing to right in *Network* 1.

Rule 1. The usage of the virtual channels is as follows:

1) At the source node, messages are sent to the virtual channel with $m = 0$ and $n \neq 4, 5$.

2) If and only if a data stream is sent from a boundary node with its x (or y) is 0 (or $2N$-1) to a node with its x (or respective y) is not 0 (or $2N$-1), the number m in $C_{U,V,n,m}$ will add 1.

3) On any routing path, the number m of $C_{U,V,n,m}$ won't decrease.



(a) Network 0 **(b) Network 1**

Fig. 3 Virtual networks.

4) When a message moves from a virtual channel in one group to a virtual channel in another group (for simplicity, described as a message moves from one group to anther group in later text), it will maintain the number *m* in $C_{U,V,n,m}$ except the situation described in 2).

5) As a message on *Group* 3 moves to dimension *Y*, it will move to *Group* 4 or 5.

6) Messages can move from *Network* 0 to *Network* 1 only.

3.2 Proofs of Deadlock-freeness

Next we show that any minimal routing algorithm based on the proposed virtual channel model is deadlock-free. Here a routing algorithm is said to be *minimal* if it only uses shortest paths. For convenience, it is assumed that there is an additional virtual channel for each group of virtual channels on every node. Namely, each physical channel on the *X* dimension is split into three virtual channels and each physical channel on the *Y* dimension is split into six virtual channels. This virtual channel model is called the Integrity Model (IM).

In the following, we first show that any minimal routing algorithm using IM is deadlock-free by proving that there is no cyclic dependency between these channels (i.e., there is no cycle in the *Channel Dependency Graph* (CDG)) [4].

Lemma 1 For any minimal routing algorithm using IM, there is no cyclic dependency between channels on single dimension.

Proof: In any minimal routing algorithm, there is no 180° turn. Hence, there is no dependency between virtual channels belonging to two groups on one dimension. Therefore, we only need consider the dependency between virtual channels belonging to the same group, which is shown below,

$C_{N_0,N_1,n,0} \rightarrow \dots \rightarrow C_{N_n,N_0,n,0} \rightarrow C_{N_0,N_1,n,1} \rightarrow \dots \rightarrow C_{N_n,N_0,n,1} \rightarrow C_{N_0,N_1,n,2} \rightarrow \dots \rightarrow C_{N_n,N_0,n,2}$

which has no cycle. Hence, there is no cyclic dependency between channels on single dimension.

Lemma 2 For any minimal routing algorithm using IM, there is no cyclic dependency between channels on two dimensions in single virtual network.

Proof: Because *Network* 0 and *Network* 1 are symmetrical, a proof is made for *Network* 0 first. Then the same conclusion can be made for *Network* 1.

As shown in Fig. 4, every node has some virtual channels in *Network* 0. Suppose in *Network* 0, there exists a cycle *D* composed of channels on two dimensions as:

$C_{U_0,V_0,n_0,m_0} \rightarrow \dots \rightarrow C_{U_i,V_i,n_i,m_i} \rightarrow C_{U_j,V_j,n_j,m_j} \rightarrow \dots \rightarrow C_{U_0,V_0,n_0,m_0}$.

According to Rule 1, we have

$$m_0 = m_1 = \dots = m_i = m_j = m_0 \qquad (1)$$

Since in *Network* 0, there is only *Group* 2 on dimension X, it is clear that *D* cannot be in the form of cycle 1 shown in Fig. 5(a) but be in the form of cycle 2 in Fig. 5(b). This means that *D* includes a movement of 2*N* steps between channels on *Group* 2. That is to say, cycle *D* must include a virtual channel corresponding to a wraparound channel on dimension *X.* Hence, according to Rule 1, on cycle *D,* there must exist two channels C_{U_k,V_k,n_k,m_k} and C_{U_h,V_h,n_h,m_h} such that

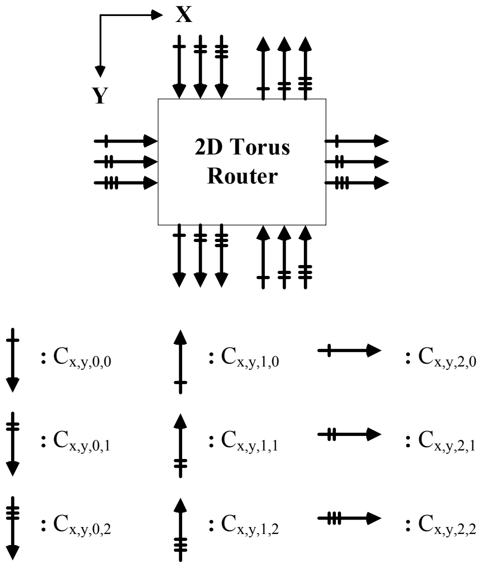


Fig. 4 Virtual channels in *Network* 0.

$$m_k = m_h + 1 \neq m_h \qquad (2)$$

Apparently, (1) and (2) are contradictory. Therefore, there's no cyclic dependency between two dimensions in *Network* 0. Similarly, the same conclusion can be made for *Network* 1.

Lemma 3 For any minimal routing algorithm using IM, there is no cyclic channel dependency between channels of two virtual networks.

Proof: Because messages can only move from *Network* 0 to *Network* 1, hence, there is no cyclic channel dependency between channels of two virtual networks.

Theorem 1 Any minimal routing algorithm using IM is deadlock-free.

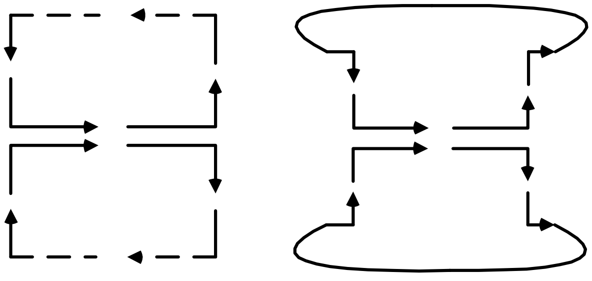
Proof: By Lemmas 1, 2 and 3, there is no cycle on the CDG of any minimal routing using IM. Hence, it is deadlock-free.

Theorem 2 Any minimal routing algorithm using the proposed virtual channel model is deadlock-free

Proof: By Theorem 1, it is clear that there is no cycle in the CDG of any minimal routing algorithm using IM [13]. As the CDG of a minimal routing algorithm using the proposed virtual channel model is the subgraph of the CDG of that using IM, there is no cycle in the CDG of virtual channel model. Hence, any minimal routing algorithm using our virtual channel model is deadlock-free.

4. Performance Evaluation

The performance of MPR using the proposed virtual channel model has been evaluated and compared with the fully adaptive Single-Path minimal Routing (SPR) algorithm for both single-source situation and multi-source situations. For single-source situation, the analysis of data throughput is same as in [9], which is omitted here.



(a) cycle 1

(b) cycle 2

Fig. 5 Possible cycles in *Network* 0.

For multi-source situations, there may exist blockings in the network. One the one hand, since multiple data streams generated from each message are transmitted in the network, the blocking probability of the MPR scheme tends to be larger than that of the SPR scheme in multi-source situations. On the other hand, because the MPR scheme improves the average data transfer time, it may reduce the blocking probability. In order to evaluate the performance of the MPR scheme for multi-source situations, simulations have been conducted for the MPR scheme and the SPR scheme on torus-based networks.

4.1 Simulation Settings

In the simulations, $0.8\mu m$ gate array technology is selected as the reference circuit technology [2]. On the torus network, all nodes generate messages independently. Each data message has fixed number of flits, and one flit only includes one phit with 16 bits. The time unit used in the simulations is the time needed to send one flit on the physical channel, referred as *cycle*. Assume that wormhole switching is used in the network. Hence, the data transfer time (latency) can be calculated as

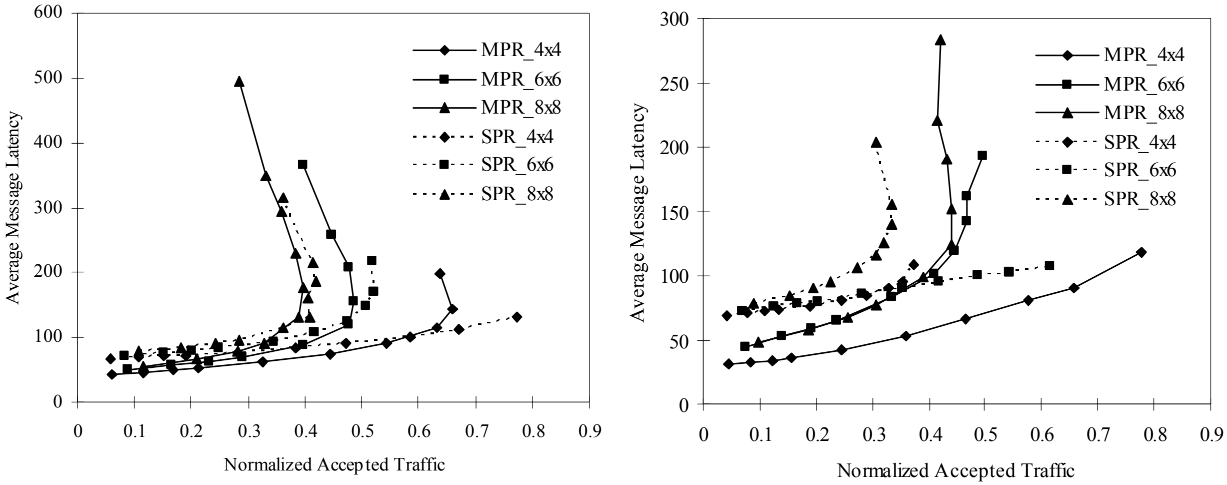
$$t_{wormhole} = t_{setup} + t_{data} \text{ [6]},$$

where t_{setup} is the setup time of a path, which is defined as the time needed for the header to set up a path from the source node to the destination node, and t_{data} is the transfer time of the data, which is defined as the time that the data is transferred from the source node to the destination node through the path set up by the header and all the flits are accepted.

Two traffic scenarios are simulated:

1) Uniform traffic: each node sends data to one of the other nodes with equal probability;

2) Nonuniform traffic: traffic is generated in bit reversal pattern [6], in which node indexed with binary number $a_0a_1\ldots a_{n-1}$ communicates with node $a_{n-1}\ldots a_1a_0$.



(a) Uniform traffic

(b) Nonuniform traffic

Fig. 6 BNF graphs for message length = 60 flits.

In the following, the Burton Normal Form (BNF) graphs [6] are presented for the simulation results of the MPR scheme (represented as MPR in all figures) and the SPR scheme (represented as SPR in all figures) for 4x4, 6x6, and 8x8 torus networks. In all simulations, the same 10 normalized network loads are applied which correspond to the 10 performance points on each line of all figures.

4.2 Simulation Results

Fig. 6 shows the average message latency (in number of cycles) vs. normalized accepted traffic (i.e., the received traffic in number of flits per node per cycle) under both uniform and nonuniform traffic scenarios with message length = 60 flits. From Fig. 6(a) and Fig. 6(b), it can be seen that for the same network size and same network load, when the network has not reached the saturation point (reflected as the turning point on each line), the normalized accepted traffic of MPR is better than that of SPR and the average message latency of MPR is less than that of SPR.

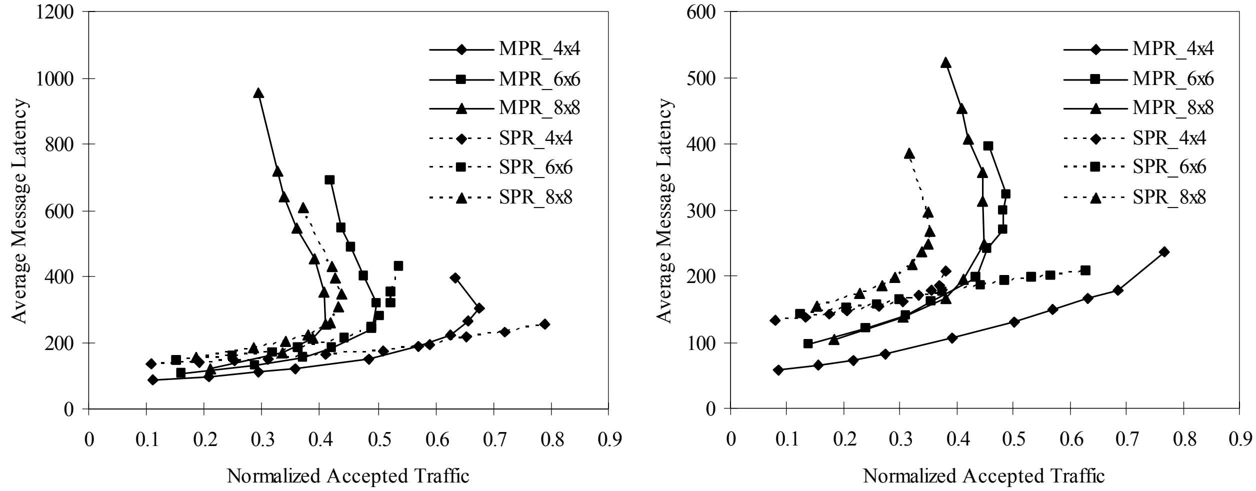
The improvement of MPR over SPR in average message latency is more significant (up to 36.5%) when the network is less loaded (corresponding to the less accepted traffic). The reason is explained below. When there are fewer messages transmitted in the network, there is nearly no blocking in the network using either SPR or MPR. Thus the setup time of the path is mainly determined by the number of hops on the path, which is same for both schemes due to their property of minimal routing. However, the shorter data stream size in MPR results in its less data latency on the path than in SPR routing. Consequently, the average message latency of MPR is less than that of SPR.

When network load is growing, the multiple data streams in MPR cause more blocking than in SPR, which increases the setup time as well as the latency experienced by each data stream. This degrades its improvement in the data latency. Noticeably there is a small performance degradation when MPR and SPR reach the saturation point (indicated by the maximum accepted traffic in the figure). If the injected traffic is sustained at this point, message latency increases considerably while accepted traffic decreases.

The figures also show that under both traffic scenarios, for the same algorithm, with network size increasing, the average message latency increases and the maximum accepted traffic (i.e., the throughput) decreases. For instance, the throughput for MPR on 4x4, 6x6, 8x8 networks is 0.66, 0.49, 0.38, respectively. This is due to the fact that the average number of hops on the path is increased and more blocking exists with network size increasing.

Comparing Fig. 6(a) and Fig. 6(b), for the same algorithm with the same network size, both the average message delay and normalized accepted traffic under uniform traffic are worse than those under nonuniform traffic. The reason is that all pairs of communicating nodes are fixed in nonuniform traffic and usually every node has only one communicating node, which will reduce the blocking in the network. Due to the less blocking encountered under this traffic scenario, MPR achieves dramatic improvement in average message latency (up to 55%) and throughput than SPR.

Fig. 7 shows the average message latency vs. normalized accepted traffic under both uniform and nonuniform traffic scenarios with message length = 120 flits. Comparing Figs. 6 and 7, for the same algorithm with the same network size under the same traffic scenario, the average message latency for message



(a) Uniform traffic

(b) Nonuniform traffic

Fig. 7 BNF graphs for message length = 120 flits.

length = 120 flits is longer than that of message length = 60 flits. The trend shown in Fig. 7(a) is similar to the trend shown in Fig. 6(a). However, the improvement achieved by MPR than SPR in Fig. 7(a) is less significant than in Fig. 6(a). The reason is that longer messages cause more blockings in the network. Different from SPR, the message latency in MPR is determined by the slowest data stream, which results in its longer message latency when the network is heavily blocked. Similar to Fig. 6, the results shown in Fig. 7(b) are better than those in Fig. 7(b) for the same algorithm and with the same network size.

5. Conclusion

In this paper, we proposed a virtual channel model for the MPR scheme, a promising routing scheme for torus-based NoCs. It is proved that any minimal routing algorithm (including MPR) using this virtual channel is deadlock-free. Through simulations, we showed that the MPR scheme using the proposed virtual channel model generally has better performance over the SPR scheme with the same virtual channel under both uniform and nonuniform traffic, especially when the network is lightly loaded. Particularly, under nonuniform traffic, MPR achieves much higher throughput and significant improvement in average message latency than SPR. These results confirm that MPR is more suitable when less blocking exists in the network. Future work includes the optimization of the virtual channel model and study of the implementation issues.

References

[1] L. Benini and G. DeMicheli, “Networks on chips: a new SoC paradigm,” *Computer*, vol. 35, no. 1, pp. 70-78, Jan. 2002.

[2] A. Chien, “A cost and speed model for k-ary n-cube wormhole routers,” *IEEE Trans. Parallel and Distributed Systems*, vol. 9, no. 2 pp.150-162, 1998.

[3] Crosstalk calculation and analysis, available at: http://www.eetchina.com/ARTICLES/2004MAY/1/2004MAY10_BD_NTFORUM01.HTM.

[4] 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.

[5] W.J. Dally, B. Towles, “Route packets, not wires: on-chip interconnection networks,” *Proc. Design Automation Conf. (DAC)*, 2001, pp. 684-689.

[6] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection Networks: an Engineering Approach*, Morgan Kaufmann Publishers, San Francisco, CA, 2003.

[7] N. Eisley and L-S. Peh, “High-level power analysis for on-chip networks,” *Proc. CASES*, 2004, pp. 22-25.

[8] A. Jantsch and H. Tenhunen, *Networks on Chip*, Kluwar Academic Publishers, 2003.

[9] Y. Jiao, Y. Yang, M. He, M. Yang, and Y. Jiang, “Multi-path routing for mesh/torus-based NoCs,” *Proc. 5th Int’l Conf. Information Technology: New Generations (ITNG)*, 2007, pp. 734-739.

[10] N. Kavaldjiev and G. M. Smit, “An energy-efficient network-on-chip for a heterogeneous tiled reconfigurable systems-on-chip,” *Proc. Euromicro Symp. Digital System Design (DSD)*, 2004, pp. 492-498.

[11] K. Lee, S-J. Lee and H-J. Yoo, “SILENT: serialized low energy transmission coding for on-chip interconnection networks,” *IEEE Int’l Conf. Computer Aided Design (ICCAD)*, 2004, pp. 448-451.

[12] P. Magarshack and P. G. Paulin, “System-on-chip beyond the nanometer wall,” *Proc. Design Automation Conf. (DAC)*, 2003, pp. 419-424.

[13] F. Petrini and M. Vanneschi, “K-ary n-trees: high performance networks for massively parallel architectures," *Proc. Parallel Processing Symp*, 1997, pp. 87-93.

[14] D. Rossi, C. Metra, A. K. Nieuwland, and A. Katoch, “Exploiting ECC redundancy to minimize crosstalk impact,” *IEEE Design & Test of Computers*, vol. 22, no. 1, pp. 59-70 , Jan. 2005.