A MULTI-PATH ROUTING SCHEME FOR TORUS-BASED NOCS[1]

Y. Jiao[1], Y. Yang[1], M. Yang[2], and Y. Jiang[2]

[1]*College of Information Technology and Science, Nankai University, China*

[2] *Dept. of Electrical and Computer Engineering, University of Nevada, Las Vegas, USA*

Emails: [1]yangyl@nankai.edu.cn, [2]{meiyang, yingtao}@egr.unlv.edu

Abstract: In Networks-on-Chip (NoC) designs, crosstalk noise has become a serious issue which may cause the communication channel unreliable. The crosstalk problem can be mitigated by wide spacing of serial lines. However, the wider spacing of serial lines will reduce the number of the lines, thus reduce the data throughput. In this paper, a new fully adaptive Multi-Path Routing (MPR) scheme is proposed 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). Theoretical analysis shows that the MPR scheme under both FM and HM achieves improvement in data throughput when single pair of nodes are in communication. When multiple pairs of nodes are in communication, simulation results demonstrate that the MPR scheme under FM significantly improves the normalized accepted traffic and throughput as well as average message latency than the YX routing algorithm in most network loads.

Key Words: Networks-on-Chip, crosstalk, multi-path routing

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

1. Introduction

Networks-on-Chips (NoC) designs will become more sensitive and prone to delay variations, noise, transient faults, and other interferences [1~3] due to the continuously shrinking geometry of semiconductor devices and the increasing switching speed. One of the main noise sources is crosstalk, which becomes a serious issue with technology scaling and can cause errors across a range of adjacent bits [2]. The crosstalk problem can be mitigated by wide spacing of adjacent wires [2]. However, for a fixed chip area, wider spacing of adjacent wires will reduce the number of wires between routers, thus reduce the data throughput.

With their simple structure, mesh/torus-type networks are widely used as on-chip interconnection networks [4, 5]. On mesh/torus networks, there exist multiple shortest paths between most pairs of source and destination nodes, but the traditional routing schemes only choose one of them for data transmission. Based on this observation, we propose the multi-path routing scheme, which features in separating the data message to be sent into multiple data streams and sending them on different paths concurrently. Using such a scheme, the data throughput can be retained while the crosstalk is reduced when wider spacing between adjacent wires is used [6]. When the spacing between adjacent wires is unchanged, a higher data throughput can be achieved using this scheme.

In this paper, we study the Multi-Path Routing (MPR) scheme for torus-type NoCs. Two transport models are considered: the full-wire-bank transport model and the half-wire-bank transport model. The proposed routing scheme is the same under both transport models. Through analysis, we show that improvement in throughput is achieved using the MPR scheme under both transport models when single pair of nodes are in communication (namely single-source situation). When multiple pairs of nodes are in communication (namely multi-source situation), simulations of the proposed MPR scheme and traditional YX routing algorithm under both uniform and nonuniform traffic have been conducted. The simulation results demonstrate the advantages of the MPR scheme over the YX routing algorithm.

The rest of the paper is organized as follows. Section 2 presents the preliminaries. Section 3 describes the multi-path routing scheme in details. Section 4 presents the performance study of the proposed scheme. Section 5 concludes the paper.

2. Preliminaries

2.1 Node and Channel Models

Each node in a torus-based NoC network is composed of a processor and a router which connects the node to the interconnection network. For simplicity, we represent a node as square in all figures. And we represent all nodes in a torus-based NoC as $2N$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.

Each node in the NoC has four physical channels, each connecting to a neighbor node. Fig. 1 shows the directions of the four channels. Every channel is separated into several virtual channels. To realize the full-adaptive routing, it is assumed that all the virtual channels are separated into two virtual networks (*Network* 0 and *Network* 1) [7, 8].

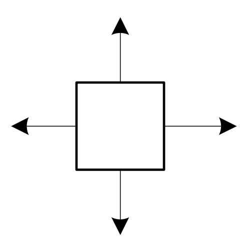


Fig. 1 Directions of the four channels.

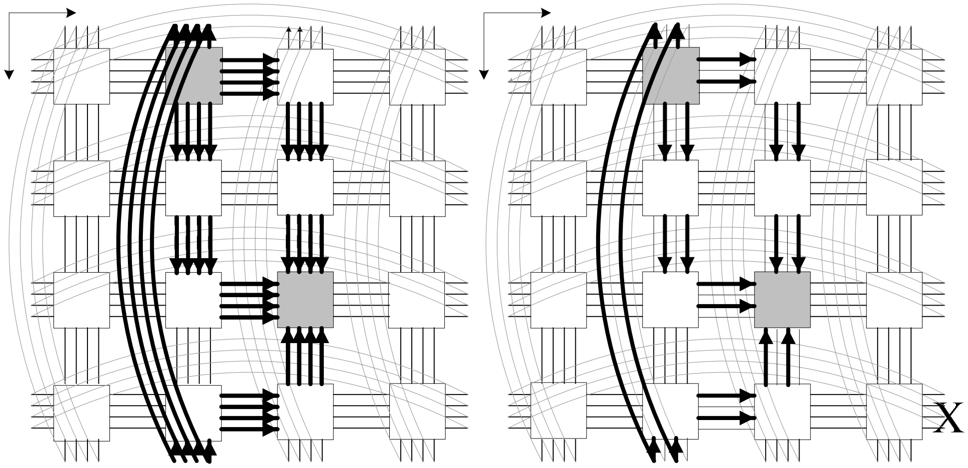


Fig. 2 MPR under FM (a) and MPR under HM (b) for 4x4 torus.

2.2 Transport Models

In the MPR scheme, when the source node needs to send data to a destination node, it will first compute the number of the shortest paths between the source and destination nodes, then partition the message into multiple data streams and send each on one of the shortest paths. We consider two transport models of the MPR scheme: the full-wire-bank transport model (FM), and the half-wire-bank transport model (HM), which are same on the routing scheme and transport control but different on their usage of the wire bank and the buffer size.

In FM, all wires on each communication link will be used for data transmission. Fig. 2 (a) illustrates an example of the MPR scheme under FM on a 4x4 torus. In this example, node 01 is the source node and node 22 is the destination node. Three shortest paths (indicated by dark lines with arrowhead in the figure) will be used for transmitting three data streams.

Different from FM, in HM, each data stream will be transmitted on half of the wires (either on odd numbered wires or even numbered wires) on each link to avoid crosstalk. Compared to FM, the crosstalk in HM is dramatically reduced according to the study in [9]. Fig. 2 (b) illustrates an example of the MPR scheme under HM on a 4x4 torus network with the same source and destination nodes as in Fig. 2 (a). Three shortest paths (indicated by dark lines with arrowhead in the figure) are also used in this model.

In the following, we will not differentiate the two models when we describe the details of the routing scheme. We will analyze and compare the performance of the MPR scheme under both models in Section 4.

3. Multi-Path Routing Scheme

3.1 Priority Dimension

The following concepts are introduced firstly before we describe the details of the MPR scheme.

Definition 1 (Boundary nodes). A node which has either 0 or $2N$-1 in one of its index number (x or y) is called a *boundary node.* And a channel that connects two boundary nodes is called a *boundary channel*.

Definition 2 (Slop over). When a data stream is transmitted from one boundary node with its x or y as 0 (or $2N$-1) to another boundary node with its x or y as $2N$-1 (or 0), we say that the data stream *slops over*.

Definition 3 (Reach boundary). When a data stream is transmitted from a node with its x (or y) not equal to 0 (or $2N$-1) to a boundary node with its x (or resp. y) as 0 (or $2N$-1), we say the data stream *reaches boundary.*

Definition 4 (Priority direction). The *priority direction* of a data stream is the direction of the channel that connects the current node (the source node or an intermediate node) to the next node along the path.

Definition 5 (Priority dimension). The *priority dimension* is the dimension that the priority direction belongs to. Note that for torus-type networks, each priority dimension can have maximally two priority directions.

The priority dimension and priority direction for a data stream at a particular node will be changed according to the following rules.

Rule I: When the data stream reaches a node, the node will find out the output directions of the shortest paths according to its index and the index of the destination node of the data stream. If there are several directions to choose, the direction on the priority dimension will be chosen. If two directions of the priority dimension can be chosen, the non-slop over direction (i.e., the dimension that the direction belongs to won't slop over on the path from the current node to the destination node) will be chosen.

Rule II: When the data stream reaches the boundary, the node will decide whether the priority dimension should be changed from dimension X (Y) to dimension Y (X) according to the values of the control bits (which will be discussed in Section 3.2).

Rule III: When there is blocking on the selected direction, the priority dimension will be decided such that blocking can be avoided.

The purpose of Rule II is to make the data streams from the same message will not block each other when there is only one source node in the network at a time (as described in Section 3.3). It is important to point out that the priority dimension can change only once for a data stream.

In the following, we will describe the details of the MPR scheme, which is composed of the operations at the source node and at intermediate nodes.

3.2 Operations at the Source Node

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, which is explained as follows.

Let (x_S, y_S) and (x_D, y_D) denote the indexes of the source and destination nodes, respectively.

Then we assign $x' = x_D - x_S$, $y' = y_D - y_S$ and name x' as the low-order difference value, and y' as the high-order difference value. The value of x' or y' falls in four different cases, each corresponding to a different operation, as shown in Table 1.

Table 1 Different cases of the value of x' or y' and corresponding operations.

Cases	Value of x' or y'	Corresponding operation
1	[1, N-1] or [-(2N-1), -(N+1)]	The data stream needs to travel along the positive direction of the X/Y dimension
2	[-(N-1), -1] or [N+1, 2N-1]	The data stream needs to travel along the negative direction of the X/Y dimension
3	$\pm N$	The data stream can travel along the positive direction or the negative direction of the X/Y dimension.
4	0	The data stream doesn't need to travel on the X/Y dimension

The number of the shortest paths is then decided by the combination of the cases of the values of x' and y', as shown in Table 2.

Table 2 Number of shortest paths vs. the combination of different cases of the values of x' and y'.

Situation	Case of x' value	Case of y' value	# of paths	C's value
1	Case 1 or Case 2	Case 4	One	1
	Case 4	Case 1 or Case 2		
2	Case 1 or Case 2	Case 1 or Case 2	Two	1
3	Case 1 or Case 2	Case 3	Three	1 if distance on priority dimension less than N, 0 otherwise
	Case 3	Case 1 or Case 2		
4	Case 3	Case 3	Four	0

After determining the number of shortest paths, the source node will check how many output ports that are available and decide the actual number of data streams that can be sent out. Then the node partitions the message into the actual number of data streams and sends the data streams on the shortest paths through the first virtual channel of the corresponding output ports.

Each data stream contains the source node index, destination node index, and two control bits (*D* and *C*), which will be used for making routing decision at the intermediate nodes on the path.

D is used to record the priority dimension and D=0 or 1 represents the priority dimension is *X* or *Y*, respectively.

C is used to indicate if the priority dimension should be changed when the data stream reaches boundary and C=0 or 1 represents the priority dimension should be changed or should not be changed, respectively. The setting of *C* is shown in Table 2.

3.3 Operations at Intermediate Nodes

Once receiving the data stream, each intermediate node will decide which output port it will forward the data stream, i.e., the corresponding priority direction to take. The decision is based on the following calculation.

Each node will first calculate the difference between its index and the index of the destination node as $x'' = x_D - x_C$, $y'' = y_D - y_C$, where (x_C, y_C) represents the index of the current intermediate node. And three sets of binary variables (A_x, A_y), (B_x, B_y), and (C_x, C_y) are derived, where

A_x = 0 or 1 represents that x'' belongs to $[1, N\text{-}1] \cup [-(2N\text{-}1), -(N+1)]$ or $[-(N\text{-}1), -1] \cup [N+1, 2N\text{-}1]$, respectively,

A_y = 0 or 1 represents that y'' belongs to $[1, N\text{-}1] \cup [-(2N\text{-}1), -(N+1)]$ or $[-(N\text{-}1), -1] \cup [N+1, 2N\text{-}1]$, respectively,

B_x = 0 or 1 represents if $|x''| \neq N/2$ or not, respectively,

B_y = 0 or 1 represents if $|y''| \neq N/2$ or not, respectively,

C_x = 0 or 1 represents whether $x'' = 0$ or not, respectively,

C_y = 0 or 1 represents whether x'' = 0 or not, respectively.

Then the priority direction on each dimension is determined according to the combination of these variables as shown in Table 3.

Table 3 Priority direction on *X* and *Y* dimension (- means either 0 or 1).

A_x B_x C_x	**Priority direction**	**Other direction**	A_y B_y C_y	**Priority direction**	**Other direction**
- 0 0	none	none	- 0 0	none	none
0 0 1	*X*+	none	0 0 1	*Y*+	none
1 0 1	X-	none	1 0 1	*Y*-	none
0 1 1	*X*+	*X*-	0 1 1	*Y*+	*Y*-
1 1 1	*X*-	*X*+	1 1 1	*Y*-	*Y*+
- 1 0	impossible		- 1 0	impossible	

The final priority direction of the incoming data stream can be determined based on Table 3 and the value of *D*. The directions on *X* dimension are more preferred than those of *Y* dimension for *D*=0, and vice versa for *D*=1.

Then the node will check if the availability of the corresponding output ports according to the following order: the priority direction on preferred dimension, other direction on preferred dimension, priority direction on non-preferred dimension, other direction on non-preferred dimension. The data stream is sent out from the first output port that is available. If there is no free output port, the data stream is blocked at the current node.

In the case that there is no direction to choose, the data stream reaches its destination.

According to above description, we can see that each data stream can choose its path based on the availability of the output ports of the current node. Hence, the MPR scheme is fully adaptive.

The following lists the steps carried at intermediate node after receiving a data stream.

Step 1. Check that whether the data stream reaches its destination. If it does, store the data stream to memory; otherwise, continue the following steps.

Step 2. When the data stream reaches boundary, if C=0, then flips the priority dimension (namely changes D's value from 0 to 1 or 1 to 0) and sets C=1; otherwise, continue the following steps.

Step 3. Decide the output directions of the data stream according to (A_x, A_y), (B_x, B_y), (C_x, C_y) and two control bits (D and C) as described above.

Step 4. Decide the output virtual channel of the data stream according to following rules.

1) Iff the data stream received at X- will change dimension, send it to *Network* 1.

2) The data streams can not move from *Network* 1 to *Network* 0.

3) Iff the current node is the boundary node and the next node is not a boundary node, the virtual channel number will add 1, otherwise the number remains the same.

Then the data stream can be sent out to the next node or blocked at the current node.

One property of the MPR scheme is stated as follows.

Theorem 1 The MPR scheme is block-free in single-source situations.

Due to space limit, the proof is omitted here.

4. Performance Study

To study the performance of the proposed MPR scheme, theoretical analysis of the data throughput is performed for single-source situations and simulations are conducted for multi-source situations and compared with the YX routing algorithm.

4.1. Analysis for Single-Source Situations

In this analysis, the data throughput refers to the throughput of valid (correct) data, which obtained by the amount of valid data received divided by the time taken to transfer the data.

Without loss of generality, assume the data unit to be considered is one message with fixed length, next we will derive the average transfer time of a message for both MPR and YX routing. Here *the rate of valid data* is defined as the amount of valid data that is transferred between two nodes connected by a channel composed of a group of wires in a unit time and the *average number of shortest paths* is defined as the average of the number of the shortest paths for all the possible cases.

We use the following notations in our analysis.

P_1: the rate of valid data when FM is used.

P_2: the rate of valid data when HM is used.

V: the data rate.

E: the average distance of the network.

H: the average amount of the shortest paths.

F_b: the length of a message.

F_h: the length of the head of a data stream.

T: the average transfer time of a message in YX routing.

T_1: the average transfer time of a message in MPR under FM.

T_2: the average transfer time of a message in MPR under HM.

For the $2N$*$2N$ torus network, the average number of shortest paths H can be derived as below.

$$H = \frac{1\times[(2N-2)\times 2]+2\times[2N\times 2N-2-(2N-2)\times 4]+3\times[(2N-2)\times 2]+4\times 1}{4N^2-1} = \frac{8N^2}{4N^2-1}$$

In YX routing, there is only one data stream for each message. And assuming wormhole switching is used, the average transfer time of a message in YX routing can be derived as,

$$T = \frac{F_h \times E + F_b}{P_1 \times V} \qquad \textbf{(1)}$$

In FM and HM, the message is transferred by H data streams at the same time. From Theorem 1, we know that there is no blocking for single-source situations. Hence T_1 and T_2 can be derived as

$$T_1 = \frac{F_h \times E + \frac{F_b}{H}}{P_1 \times V} \text{ and } T_2 = \frac{F_h \times E + \frac{F_b}{H}}{P_2 \times \frac{V}{2}} \qquad \textbf{(2)}$$

Then the ratio of the average transfer time of MPR under FM to that of YX routing and the ratio of MPR under HM to YX routing are given by

$$\frac{T}{T_1} = \frac{F_h \times E + F_b}{F_h \times E + \frac{F_b}{H}} \text{ and } \frac{T}{T_2} = \frac{F_h \times E + F_b}{F_h \times E + \frac{F_b}{H}} \times \frac{P_2}{2 \times P_1} \textbf{(3)}$$

Since F_h and E are very small compared with F_b, we can derive the above expressions as $T / T_1 = H$ and $T / T_2 = (H / 2) \times (P_2 / P_1)$. It is clear that the speedup of MPR under FM to YX routing lies on the average number of shortest paths. Since the valid date rate of MPR under FM is low than that of MPR under HM due to the severer crosstalk existing in FM, i.e., $P_1 / P_2 < 1$, and $H > 2$, MPR under HM also achieves speedup than YX routing. Hence, MPR under FM/HM has better data throughput than YX routing.

From Eqn. (3), $T_1 / T_2 = P_2 / (2 \times P_1)$ can be derived. That's to say,

1) when $2 * P_1 - P_2 < 0$, $T_1 / T_2 > 1$, namely MPR under HM is better than MPR under FM.

2) when $2 * P_1 - P_2 > 0$, $T_1 / T_2 < 1$, namely MPR under FM is better than MPR under HM.

Particularly, when $P_1 > 0.5$, $2 * P_1 > 1$. And $P_2 <= 1$, thus it is true that $2 * P_1 - P_2 > 0$. Namely when FM is used, if the rate of valid data is larger than 0.5, MPR under FM has better data throughput than MPR under HM.

4.2. Analysis for Multi-Source Situations

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 YX routing 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 under FM and YX routing on torus-based networks.

In the simulations, 0.8 μm gate array technology is selected as the reference circuit technology [10]. 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 bit width of each physical channel is 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 $t_{wormhole} = t_{setup} + t_{data}$ [11], 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).

We consider two traffic scenarios: uniform (where each node sends data to one of the other nodes with equal probability) and nonuniform [12] (where traffic is generated in bit reversal pattern [13], in which node indexed with binary number $a_0a_1\ldots a_{n-1}$ communicates with node $a_{n-1}\ldots a_1a_0$). In the following, we present the Burton Normal Form (BNF) graphs [14] for the simulation results of the MPR scheme (represented as MPR in all figures) and YX routing (represented as YX 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.

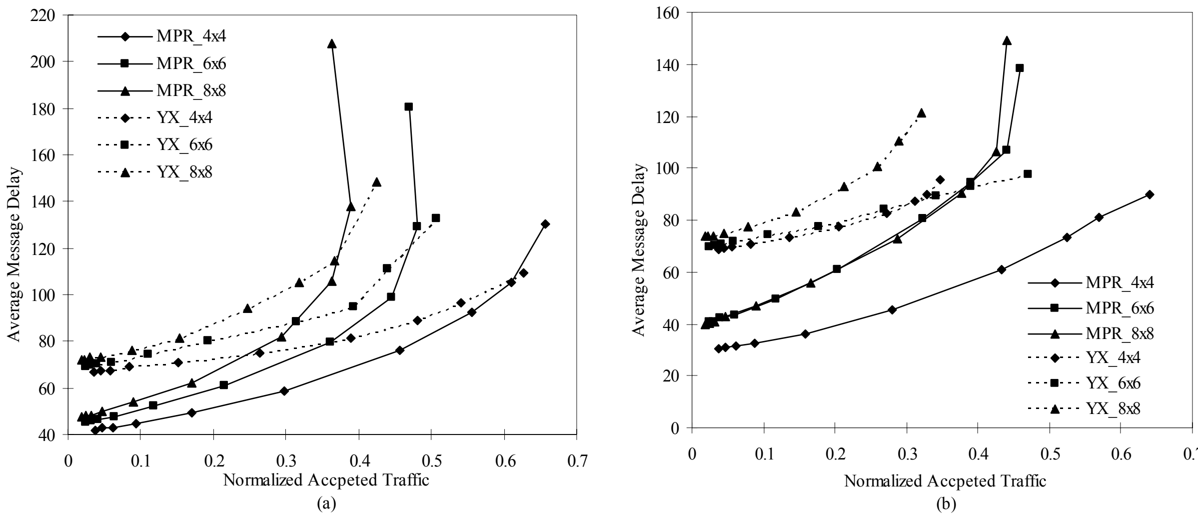


Fig. 3 Average message latency vs. normalized accepted traffic under uniform traffic (a) and nonuniform traffic (b) with message length = 60 flits.

Fig. 3 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. 3(a) and Fig. 3(b), we can see that for the same network size and same network load, the normalized accepted traffic of MPR is better than that of YX routing and the average message latency of MPR is less than that of YX routing for most network loads. The improvement of MPR over YX routing in average message latency is more significant (up to 37%) 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 for both MPR and YX routing, thus the setup time of the path is mainly determined by the number of hops on the path, which is same for both algorithms (as shortest paths are used in both algorithms). However, the shorter data stream size in MPR results in its less data latency on the path than in YX routing. Consequently, the average message latency of MPR is less than that of YX routing.

When network load is growing, the multiple data streams in MPR cause more blocking than in YX routing, which increases the setup time and degrades its improvement in the data latency. Noticeably there is a small performance degradation when MPR reaches 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. In Fig. 3(a), the saturation points for MPR_6x6 and MPR_8x8 are 0.48 and 0.39, respectively.

The figure also shows 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. 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. 3(a) and Fig. 3(b), for the same algorithm with the same network size, the performance under uniform traffic is worse than that 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.

Fig. 4 shows the average message latency vs. normalized accepted traffic under both uniform and nonuniform traffic scenarios with message length = 120 flits. The trend shown in Fig. 4(a) is similar to the trend shown in Fig. 3(a). However, the improvement achieved by MPR than YX routing in Fig. 4(a) is less significant than in Fig. 3(a). The reason is that longer messages cause more blockings in the network. Different from the YX routing, 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. 3, the results shown in Fig. 4(b) are better than those in Fig. 4(b) for the same algorithm and with the same network size.

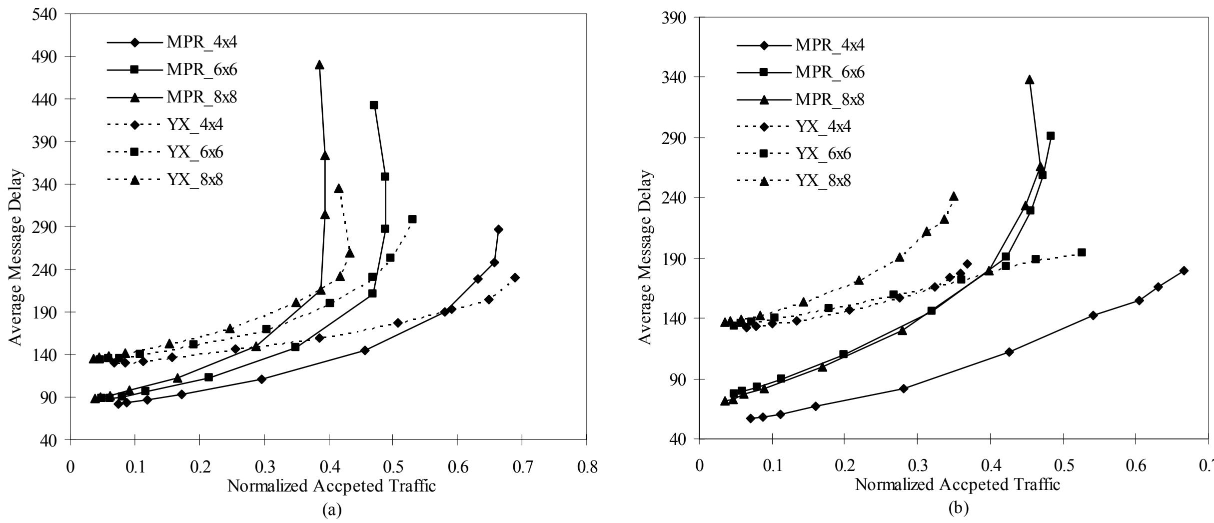


Fig. 4 Average message latency vs. accepted traffic under uniform traffic (a) and nonuniform traffic (b) with message length = 120 flits.

5. Conclusion

This paper considered the crosstalk problem existing in NoCs and proposed a fully adaptive multi-path routing scheme for torus-based NoCs. The proposed MPR scheme features in using multiple shortest paths to transfer data concurrently. Through analysis, we showed that the MPR scheme under both FM and HM achieves improvement in data throughput for single-source situations. By simulations, we showed that the MPR scheme achieves better performance under both uniform and nonuniform traffics compared with the YX routing algorithm in general. Hence, the proposed MPR is promising for NoC applications. Future work includes the study of deadlock-freeness and optimization of the number of virtual channels for the MPR scheme.

References

[1] D. Rossi, C. Metra, A. K. Nieuwland, and A. Katoch, Exploiting ECC redundancy to minimize crosstalk impact, *IEEE Design & Test of Computers*, *22*(1), 2005, 59-70.

[2] A. Jantsch and H. Tenhunen, *Networks on chip* (Norwell, MA: Kluwar Academic Publishers, 2003).

[3] L. Benini and G. DeMicheli, Networks on chips: a new SoC paradigm, *Computer*, *35*(1), 2002, 70-78.

[4] W. J. Dally, *A VLSI architecture for concurrent data structures*, Ph.D. Dissertation, California Institute of Technology, 1986.

[5] 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, 492-498.

[6] W. J. Dally and C. L. Seitz, Deadlock-free message routing in multiprocessor interconnection networks, *IEEE Trans. Computers*, *C-36*(5), 1987, 547-553.

[7] D. H. Linder and J. C. Harden, An adaptive and fault tolerant wormhole routing strategy for k-ary n-cubes, *IEEE Transactions on Computers*, *C-40*(1), 1991, 2-12.

[8] J. Duato, A new theory of deadlock-free adaptive routing in wormhole networks, *IEEE Trans. on Parallel and Distributed Systems*, *4*(12), 1993, 1320-1331.

[9] J. P. Uyemura, *Introduction to VLSI circuits and systems* (New York: John Wiley & Sons, 2002).

[10] A. Chien, A cost and speed model for k-ary n-cube wormhole routers, *IEEE Trans. Parallel and Distributed Systems*, *9*(2), 1998, 150-162.

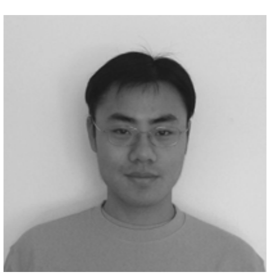
[11] J. Duato, S. Yalamanchili, L. Ni, *Interconnection networks: an engineering approach* (San Francisco, CA: Morgan Kaufmann Publishers, 2003).

[12] J. H. Kim and A. A. Chien, An evaluation of the planar/adaptive routing, *Proceedings of the 4th IEEE Symposium on Parallel and Distributed Processing*, 1992, 470-478.

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

[14] J. Duato and P. Lopez, Performance evaluation of adaptive routing algorithms for k-ary n-cubes, *Proc. 1st Int'l Workshop on Parallel Computer Routing and Communication (PCRCW)*, 1994, 45-59.

Biographies



Yaoting Jiao received the BE degree from Nankai University, China in 2005. He is currently a graduate student in College of Information Technology and Science, Nankai University, China. His research interests include interconnection network, networks-on-chip and parallel processing.

Yulu Yang received the B.E. from Beijing Agriculture Engineering University, China, in 1984, the M.E. and Ph.D degrees From Keio university, Japan, in 1993 and 1996. He works in the Department of Computer Science, Nankai University, China. His research interests include interconnection network， parallel processing， reconfigurable computing and Grids.

Mei Yang received her Ph. D. degree in Computer Science from the University of Texas at Dallas in Aug. 2003. She is currently an Assistant Professor in the Department of Electrical and Computer Engineering at University of Nevada, Las Vegas (UNLV). Before she joined UNLV, she was an Assistant Professor in the Department of Computer Science at Columbus State University from Aug. 2003 to Aug. 2004. Her current research interests include wireless sensor networks, mobile ad hoc networks, computer architectures, and embedded systems.

Yingtao Jiang received the B.Eng. degree in biomedical engineering and electronics from Chongqing University, Chongqing, China, the M.A.Sc. degree in electrical engineering from Concordia University, Montreal, Canada, and the Ph.D. degree in computer science from the University of Texas at Dallas, Richardson, Texas, in 1993, 1997, and 2001, respectively. He has been with the Department of Electrical and Computer Engineering, University of Nevada, Las Vegas since 2001, firstly as Assistant Professor, and now as Associate Professor. His research interests include algorithms, VLSI architectures, and circuit level techniques for the design of DSP, networking, and telecommunications systems, computer architectures, nano technologies, and biomedical signal processing, instrumentation, and medical informatics.