

## DTDOR: An Efficient Routing Algorithm for Torus Networks

Xinyu Wang<sup>1</sup> and Nan Li<sup>2</sup>

<sup>1</sup>*College of Management Science and Engineering, Dongbei University of Finance and Economics, China*

<sup>2</sup>*College of Management Science and Engineering, Dongbei University of Finance and Economics, China*

<sup>1</sup>*Distribute\_2008@163.com,* <sup>2</sup>*linan\_123456@163.com*

### Abstract

*Many different factors, such as topology, routing technique, selection function, flow control policy, complexity of router design, contribute to the performance of networks on chip (NoCs). Among these factors, the choice of topology and routing function has a significant effect on the average packet latency and saturation behavior. Torus is popular in many application domains, while the problem of virtual channel misbalance caused by its deadlock avoidance scheme brings lots of performance pathologies. In this paper, we present a novel deadlock avoidance scheme based on draining scheme, and propose a deadlock-free routing scheme for torus networks. We quantify the effects of the proposed routing scheme on the overall network performance by presenting simulation results for 1-D torus and 2-D torus NoCs. Experimental results show that the novel routing algorithm for torus networks could make more efficient use of virtual channel resources, and hence cutting down packet latency and boosting throughput. Note although this paper has considered 1-D and 2-D torus networks in the experimental evaluation, the new scheme is flexible enough to support high-radix torus networks.*

**Keywords:** *networks on chip, torus, deadlock avoidance scheme, draining scheme, routing algorithm*

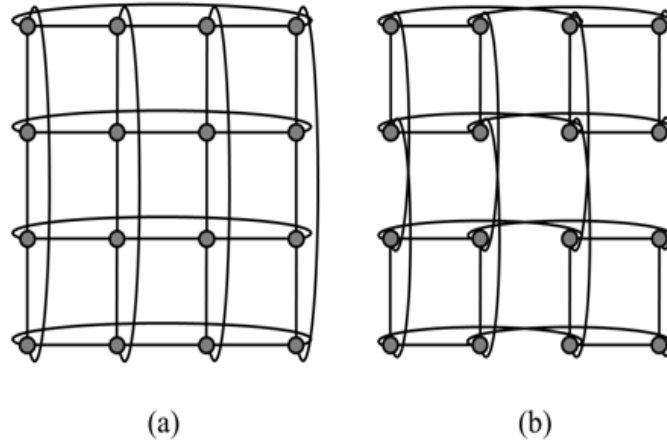
### 1. Introduction

NoCs are diving into the marketplace aggressively for its better scalability and superior performance than traditional forms of on-chip interconnects [9]. Communication in NoCs is accomplished by sending messages through the fundamental network. The NoC can be simply characterized by three elements, including topology, routing algorithm and flow control policy [12].

Torus, as an essential ingredient for NoCs, are very popular in the real application domains for its superior architecture proprieties, such as topology regularity (symmetry), high path diversity, low node degree, and short physical links. Torus networks are widely used in both commercial and experimental supercomputers and datacenter networks (such as Cray XT series and Eurotech Aurora Datacenter (3-D torus) [12], IBM Blue Gene L/Q (5-D torus) [1], Fujitsu K computer (6-D mesh /torus) [3]).

There are many different criteria to classify the routing schemes [10]. Minimal routing is often a preferred choice, as it helps not only in reducing the energy consumption of communication, but also in keeping the network free from livelock situations [13]. In this paper, we deal with minimal routing in NoCs. Further, we can classify the minimal routing schemes into deterministic routing and adaptive routing depending on the adaptivity. In

deterministic routing, packets with the same source and destination will always route along the same one path, without exception. In contrast, a packet may have many different routing



**Figure 1. 4 × 4 Network: (a) Torus, (b) Folded Torus**

paths from its source to its destination, and it can select one path from the options according to the output selection strategy. So packets with the same source and destination may route along different paths. The two classes of routing schemes have their own advantages: the former is usually simple and easy to implement as well as that it guarantees in-order packet transmitting; the latter always supplies adaptivity, which will contribute to reduced contention, and thus low latency and high throughput.

Communication between different processing elements (PEs) is realized by sending messages through the fundamental network. Usually, a message can be divided into different packets, and each packet consists of several flits including one head flit, several data flits and one tail flit. Here, flit is the smallest flow control unit [4]. Flow control policy deals with the allocation of channel and buffer resources to a packet as it travels along the routing path [11]. Buffer and link are two types of important resources in NoCs. Buffer resources are decoupled from transmission resources by using virtual channels, which makes it possible that active packets pass blocked packets. The resource decoupling of buffers and channels in the network makes more effective use of the two types of resources [4]. Using virtual channels, physical channel bandwidth can be allocated in a more flexible manner. Wormhole switching and virtual channel flow control are adopted in modern routers of NoCs in order to achieve high performance.

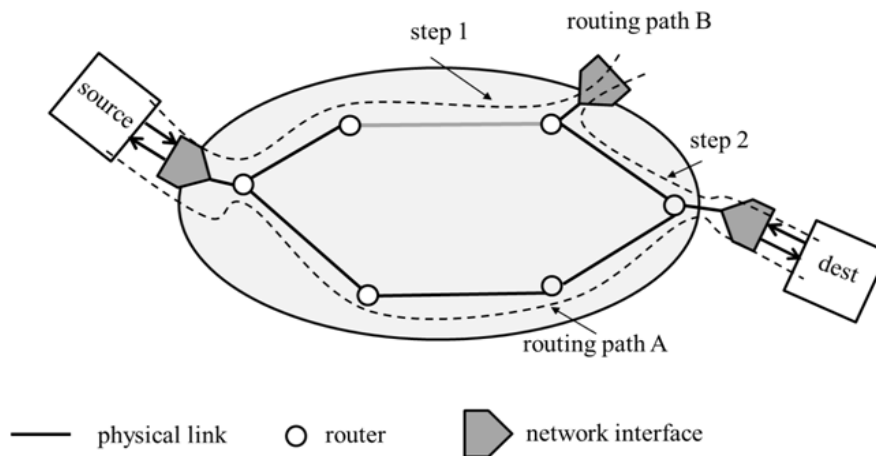
In the rest of the paper, we present related work in Section 2. Section 3 illustrates the idea behind the proposed routing scheme for n-D torus networks, and also gives the deadlock-freedom proof. Section 4 gives the performance comparisons by applying in 1-D and 2-D torus networks. Finally, Section 5 concludes the paper.

## 2. Related Work

A lot of previous work has focused on the research of routing technique in mesh-based NoCs. The well-known routing algorithms [11] include dimension order routing (DOR), west-first routing (WFR), north-last routing (NLR) [5], odd-even turn model [6], and duato's protocol. Fe, *et al.*, [14] extended the turn model by providing more adaptivity for packets, and proposed the abacus turn model (AbTM), which aims to design time/space-efficient reconfigurable routing algorithm.

When the network scales, the diameter and average hops of the whole network grow rapidly in mesh network. Torus was proposed as a solution to the problems by introducing one bidirectional wraparound link in each row (and column). The long wraparound links can be avoided by rearranging routers, that is, folded torus. Figure 1 (a) and (b) show the  $4 \times 4$  torus and folded torus, respectively. However, due to the wraparound links introduced in torus networks, the classical algorithms proposed for mesh network could not directly be applied to torus. For example, the DOR algorithm for mesh network requires only one virtual channel, while the dimension order routing in torus requires at least two virtual channels to prevent the cyclic dependencies within each row (and column). Besides, the utilization of the two virtual channels is extremely unbalancing.

In order to break the cyclic channel dependency and make better use of the virtual channels, plenty of papers presented solutions in the 1-D torus (ring). Among these routing algorithms, at least two virtual channels are required to avoid deadlock in the network. For ease of use, we use  $VC_0$  and  $VC_1$  to denote the two virtual channels. Dally proposed a simple scheme to break the cyclic channel dependency by restricting the use of virtual channel [11]. The wraparound link is designated as the dateline. If a packet has never traversed the dateline link, virtual channel  $VC_0$  is selected to transmit the packet; otherwise, virtual channel  $VC_1$  is selected. Dividing the use of the two virtual channels based on whether or not a packet has traversed the dateline in effect converts the cycle of resource dependencies into an acyclic spiral. Hence, deadlock will not occur. In-order allocation of resources could always guarantee deadlock-freedom. First, each router is assigned a label, packets with source label greater than destination label will route in  $VC_0$ ; otherwise, route in  $VC_1$ . The restriction in use of the two virtual channels is released in [11] to balance the average number of packets that traverse each of the virtual channels for a particular physical channel. In this scheme, packet that need to travel across the dateline must start in  $VC_0$  and transit to  $VC_1$  after traversing the dateline; while packets that do not need to travel across the dateline can use either  $VC_0$  or  $VC_1$ .



**Figure 2. Packet Routing Path in the Fundamental Network with and without Draining Technique**

The routing technique mentioned above could be applied to 1-D torus, and also be combined with the idea of dimension order routing thus extending to n-D torus. However,

among those methods, they have the following common drawbacks: (1) at least two virtual channels are required; (2) use of the two virtual channels is strictly restricted; and (3) the utilization of the two virtual channels is extremely unbalancing.

### 3. The Proposed Routing Algorithm

Since modern routers are required to be operational with a high clock frequency, a good routing algorithm should be simple but efficient. In addition to increasing throughput, virtual channels provide an additional degree of freedom in allocating resources to packets, that's the reason why we aims to release the restriction in use of the virtual channels.

#### 3.1 Draining Technique

Modern routers of a NoC often break a message into several fixed-size packets, and each packet routes separately in the fundamental communication network. In the traditional NoCs, a packet generated from source and destined to destination PE will traverse a batch of resources including physical links and routers. Before arriving at the destination PE, the packet will not be removed from the network through any router. That is, a packet will always route directly from source to destination. As shown in Figure 2, routing path A illustrates this situation.

The basic idea of draining technique is quite simple. When a packet has traversed a special link in the last hop, it is removed from the network by the local port of the router and delivered to the messaging layer of the local node's operating system for reinjection. Here, we name the process of removing and reinjecting with draining process. This technique was originally proposed for fault-tolerance. Wang, *et al.*, [7] extended this scheme to their proposed TM network. Flich [15] applied this scheme to unregular network and proposed the deadlock-free routing scheme based on ITB (in transit buffers) scheme. In a network adopting draining technique, packets routing may be divided into several steps, and each intermediate step contains a draining process. As shown in Figure 2, routing path B with two steps (step 1 and step 2) illustrates another situation for routing the packet from source to dest.

#### 3.2 Description of the Proposed Routing Algorithm

For packets routing in the draining-based networks, we should check whether the packet has traversed the special link in the last hop. It is very important to establish which links are special and which are not. In this subsection, we first give the definition of the special links.

In torus networks, all links in each dimension form a ring. In order to break the cyclic channel dependency, we assign the wraparound link special link, and assign other link regular link. To distinguish special links in different dimensions, we partition the special links into  $n$  sets in an  $n$ -D torus network. Each set are composed of all special links in the same dimension. For example, all special links in dimension 1 belong to the same set, and we label it with  $S_1$ .

Packet routing in torus network could be divided into two classes: (1) that does not need to traverse any special link; (2) that need to traverse one or more special links. For a packet that does not need to traverse any special link, it routes directly from source to destination without any draining process in the intermediate router. When transmitting packets with type of (2), the routing component first checks the link taken by the packet in the previous hop. If it is a special link, the routing component will incur the draining process in the current router; otherwise, the packet will be routed towards to its

destination router. It is worth to mention that, if the packet has traversed all hops in dimension  $i$ , and its previous hop is a special link belonging to  $S_i$ , the packet will not incur the draining process in the current router since it will traverse to the next dimension.

---

Input: coordinate of current router ( $c_1, c_2, c_3, \dots, c_n$ )  
coordinate of destination router ( $d_1, d_2, d_3, \dots, d_n$ )  
link taken in the previous hop (prelink)

Output: a selected output port

---

1. for ( int  $i = 1$ ;  $i \leq n$ ;  $i++$  )
2.     if(  $c_i \neq d_i$  ) break;
3.     if(  $i == n + 1$  ) { route packet to  $P_0$ ; return; }
4.     if( prelink.type == special && prelink  $\in S_i$  )
5.         { route packet to  $P_0$ ; return; }
6.     if(  $c_i < d_i$  ) route packet to  $P_{i+}$ ;
7.     else route packet to  $P_{i-}$ . return;

---

**Figure 3. Description of DTDOR in n-D torus**

In an n-D torus network, each router has  $(2n + 1)$  output ports. Among them, one is local port, while the others are output ports along different dimensions. In this paper,  $P_0$  stands for the local output port, while  $P_i$  ( $1 \leq i \leq n$ ) stands for the output port along dimension  $i$ . For example,  $P_{1+}$  stands for the positive output port along dimension 1; while  $P_{2-}$  stands for the negative output port along dimension 2.

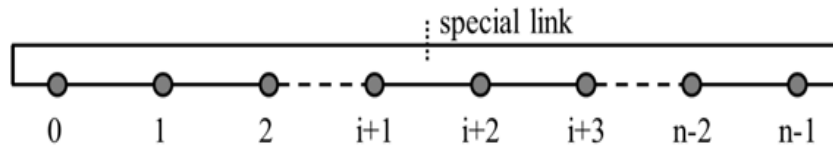
The dimension order routing in torus network proposed by Dally [11] is often referred to DOR in torus. Since the routing algorithm proposed for torus network adopts draining technique and adhere to the dimension order principle, we call the novel proposed algorithm DTDOR in torus.

As we all know, routing algorithm directs packets routing in the fundamental network, and provides the output port in each hop. Figure 3 presents the pseudo code of DTDOR in torus networks. The inputs of the algorithm include coordinates of current router and destination router as well as the link taken in the previous hop, since it is required to check whether the packet has traversed a special link. The algorithm returns a selected output port using for routing the packet. First, the DTDOR algorithm scans all dimensions from low to high, and aborts when the source coordinate differs from the destination along some dimension (line 1-2). If source coordinate is equal to destination, the packet is now in the destination router, and will be sent to the local processing element via the local output port (line 3). Otherwise, suppose the dimension is  $i$  where source coordinate differs from destination, routing packets in the following rule: (1) if the packet has traversed a special link along dimension  $i$  in the previous hop, a draining process will be triggered in the current router, that is the packet will be consumed in the current router via the local output port, and later will be re-injected into the network through the local input port (line 4-5); (2) if not, the packet will route in dimension  $i$ , that is, transverse to  $P_{i+}$  if  $c_i$  is less than  $d_i$  and transverse to  $P_{i-}$  if  $c_i$  is greater than  $d_i$  (line 6-7).

In the transmitting end of the network interfere, a message is divided into several fixed-length packets; while in the receiving end, packets of the same message will be combined into one message, and then sent to its destination processing element. In the network with draining technique, the network interfere will check the destination address of each packet, and compare with that of its local router. If the same, the packet has arrived its destination; if not, the packet will be sent to the local input port and queue to be injected in to the network. Even though some packets are removed from the network before arriving their destination node, they will not be wrongly sent to other processing elements.

### 3.3 Deadlock Freedom Proof of DTDOR

Deadlock freedom and livelock freedom are two important metrics for routing design in NoCs. Since the proposed routing scheme in this paper is minimal, livelock-freedom can be guaranteed. In this subsection, we will show that the proposed DTDOR is deadlock-free.



**Figure 4. An N-node Ring (1-D torus) Network**

**Lemma 1.** DTDOR is deadlock-free for n-D torus.

**Proof:** Consider an arbitrary unidirectional ring with n nodes, as shown in Figure 4. From the description of the routing algorithm, we know that two types of channel dependencies exist in the network: (i) from regular link to regular link; (ii) from regular link to special link. Due to the draining scheme, channel dependency from special link to regular link is disallowed. In minimal routing, U-turn does not exist in the path, thus, a cyclic channel dependency should contain both regular and special links. Due to the broken of channel dependency from special link to regular link, the cyclic channel dependency are broken, thus, deadlock will not occur in the 1-D torus network adopting the proposed scheme based on draining scheme.

As for DTDOR in n-D torus, we combine the draining scheme with dimension order routing. Channel dependency from a low dimension to a high dimension is legal, while not vice versa. Thus, cyclic channel dependency among dimensions will not occur. Since each dimension will not form cyclic channel dependency, the whole n-D torus network adopting DTDOR will not form any cyclic channel dependency.

Based on the analysis above, it is clear to conclude that the DTDOR proposed for n-D torus network is deadlock-free.

## 4. Evaluation

Although the proposed routing algorithm DTDOR can be extended to n-D torus network. Here, n can be set to arbitrary positive integer value. For the sake of the example, we developed an architectural level cycle-accurate C++ simulator of 1-D and 2-D torus-based wormhole-switched NoC, since low dimensional torus are widely used

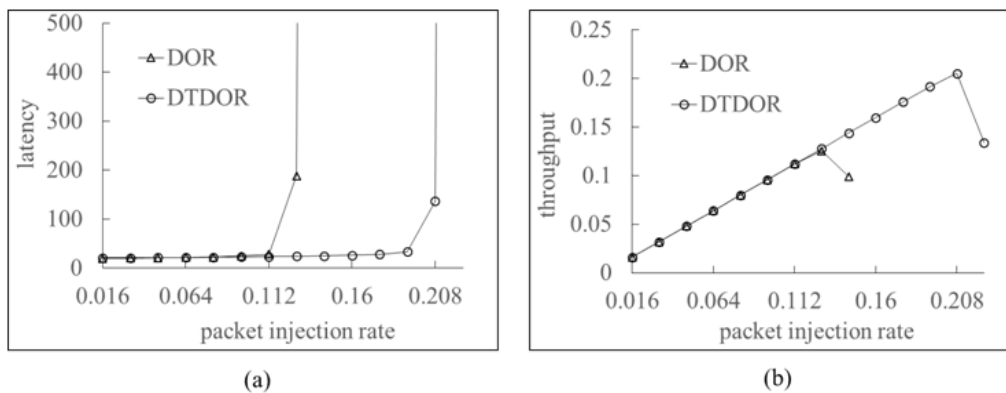
in NoCs. The input of the simulator are parameterizable allowing us to specify parameters such as network size ( $nz$ ), traffic pttren ( $tp$ ), the number of virtual channels per physical channel ( $N_{vc}$ ), buffer depth ( $bd$ ), packet injection rate ( $pir$ ), the number of flits per packet ( $N_{flits}$ ). In all runs for this paper, we choose typical values for some of the parameters:  $nz = 16$  for 1-D torus and  $nz=8 \times 8$  for 2-D torus,  $N_{vc}=2$ ,  $bd=4$  flits, and  $N_f = 5$  flits.

The routing algorithm compared is the classical dimension order routing in torus adopt dateline scheme, in this paper, we label it with DOR. Besides, the two algorithms are compared in both 1-D torus and 2-D torus to show the effectiveness of DTDOR. Since DOR in torus requires at least two virtual channels to avoid deadlock, for the sake of fairness, two virtual channels are supplied to both DOR and DTDOR.

#### 4.1 Traffic Pattern and Evaluation Metrics

It is common to evaluate the performance of a proposed algorithm under a range of synthetic traffic patterns. In this paper, we evaluate the proposed DTDOR by using uniform and hotspot traffic patterns. In uniform traffic, communication is evenly distributed in the whole network. While in the hotspot traffic, some nodes are designated as hotspots, which will receive more traffic than other nodes. Usually, hotspot traffic is considered to be a more realistic traffic model as, in most applications, processes communicate frequently with only some of the total number of other processes (*e.g.*, Memory resources, I/O resources) [8].

As for performance metrics, we choose throughput (flits / node / cycle) and latency (cycles / packet). We indicate with packet injection rate ( $pir$ ) the rate at which packets are injected into the network. For a given packet injection rate, a simulation is conducted to evaluate the average packet latency and throughput in a network adopting DOR or DTDOR. Each simulation run for a synthetic traffic is 100,000 clock cycles long. For all runs, a warm-up period of 10,000 clock cycles is assumed to lead the network to a stable state.

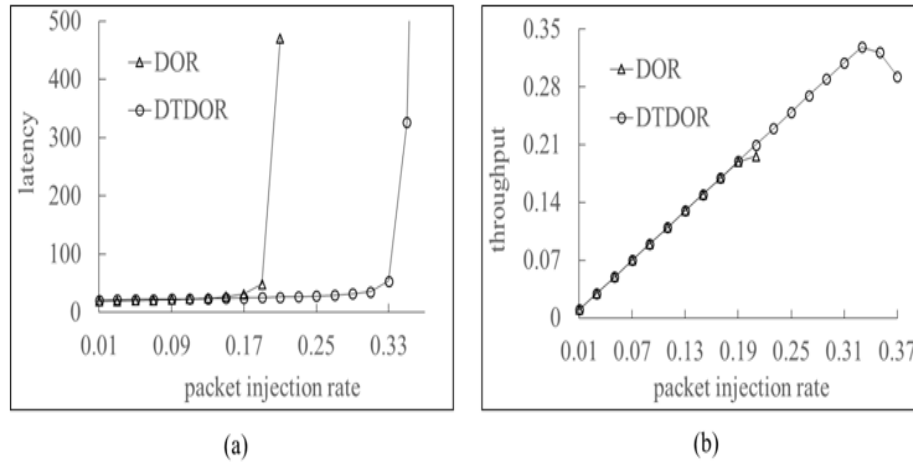


**Figure 5. Performance Comparison between DTDOR and DOR in 1-D Torus under Uniform Traffic: (a) Pir-latency Curve, (b) Pir-throughput Curve**

#### 4.2 Results and Discussion

We run the simulation for DOR and DTDOR in both 1-D and 2-D torus networks. With only one virtual channel, deadlock-free routing algorithm cannot be supported by the previous routing schemes, so we set the number of virtual channels to two

associated with one physical link (channel). Figures 5 (a) and (b) show the average latency and throughput versus packet injection rate in 1-D 16-node torus under uniform traffic. From the Figure we can see that, the average latency of the two routing schemes are comparable in low load rate network ( $pir < 0.112$  flits / node / cycle). When  $pir$  is higher than 0.112 (flits / node / cycle), the proposed routing algorithm DTDOR has lower latency than DOR. Also, the proposed DTDOR performs better than DOR in the respect of throughput. Although the traffic under uniform is already well balanced, the gain of DTDOR is still apparent.

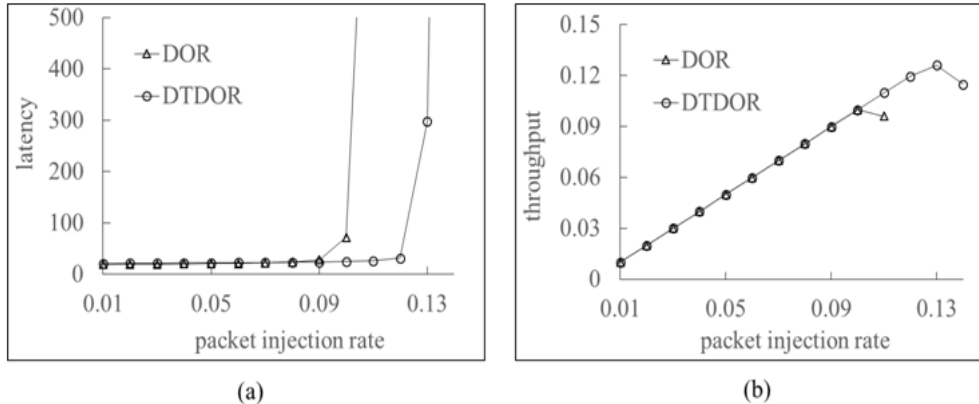


**Figure 6. Performance Comparison between DTDOR and DOR in 2-D Torus under Uniform Traffic: (a)  $Pir$ -latency Curve, (b)  $Pir$ -throughput Curve**

Figure 6 and Figure 7 show the performance comparisons of the two routing algorithms in 2-D  $8 \times 8$  torus networks. In uniform traffic, as shown in Figure 6, it is observed that, for a  $pir$  value of 0.19 (flits / node / cycle), the proposed DTDOR gives an reduction of 47.6% in average latency as compared to DOR. The proposed routing scheme performs better than DOR from both the saturation and peak throughput viewpoints. In hotspot traffic, as shown in Figure 7, the saturation point of DTDOR and DOR are 0.13 (flits /node /cycle) and 0.11 (flits /node /cycle), respectively. As for metric of the peak throughput, DTDOR and DOR gain 0.126 (flits /node /cycle) and 0.100 (flits /node /cycle), respectively. The proposed routing algorithm DTDOR outperforms DOR by 18.2% and 26% in terms of saturation point and peak throughput enhancement.

Why does the proposed DTDOR beat DOR beautifully in both average latency and throughput? Now we give a brief analysis. In DOR scheme, the two virtual channels are strictly constrained, and the utilization of them is extremely uneven. However, in DTDOR scheme, the two virtual channels are always supplied, and they are used with no engagement. Therefore, the proposed routing scheme releases the resource allocation of virtual channels, and makes it more even for the utilization of the two virtual channels, which directly translates directly to a higher throughput and lower average delay.





**Figure 7. Performance Comparison between DTDOR and DOR in 2-D Torus under Hotspot Traffic: (a) Pir-latency Curve, (b) Pir-throughput Curve**

## 5. Conclusion

A good routing scheme should make sufficient use of the resources and achieve good performance. In this paper, we have proposed a new routing algorithm based on draining scheme, which produces a series of deadlock-free routing algorithms DTDOR for n-D torus. By relieving the restriction of using virtual channels, our proposed DTDOR can make more effective use of virtual channel resources, therefore yields lower latency and higher throughput. Simulation results show that DTDOR can effectively enhance virtual channel utilization, and brings > 15% improvement when compared to DOR scheme. Moreover, the proposed DTDOR in this paper is simple and easy to implement, and can be extended to n-D torus networks.

In our future work, we would like to develop an efficient mapping scheme for different special links, and we aim to develop a selecting scheme to evenly distributed special links in the network, research about that will also be considered in our future work.

## Acknowledgement

This work is now supported by National Natural Science Foundation of China (No 71002094). The idea of this work came out when the author (X. Wang) was pursuing her PhD degree in Tsinghua University. She would like to thank her professor, and the related 863 and NSFC projects when she was studying in Tsinghua University. Besides, the valuable suggestions from the PhD students Zhigang Yu and Kele Shen in Tsinghua are extremely valuable, the authors really appreciate these suggestions, and also would like to thank both of them here. Also, the authors would like to express their thanks to anonymous reviewers for this manuscript; their constructive comments contribute to the improvement of the quality of this work.

## References

- [1] N. R. Adiga, M. A. Blumrich, D. Chen, P. Coteus, A. Gara, M. E. Giampapa, P. Heidelberger, S. Singh, B. D. Steinmacher-Burow, T. Takken, M. Tsao and P. Vranas, "Blue Gene/L torus interconnection network", IBM J. Research and Development, vol. 49, no. 3, (2005).
- [2] D. Abts, "The cray XT4 and seastar 3-D torus interconnect", Encyclopedia of Parallel Computing, vol. 6, (2011).

- [3] Y. Ajima, R. Inoue, S. Hiramoto and T. Shimizu, "Tofu: A 6-D mesh/torus interconnect for exascale computers", J. Fujitsu Science and Technology, vol. 48, no. 3, (2012).
- [4] W. J. Dally, "Virtual-channel flow control", IEEE Trans. on Parallel and Distributed Systems, vol. 3, no. 2, (1992).
- [5] C. J. Glass and L. Ni, "The turn model for adaptive routing", ACM SIGARCH Computer Architecture New, vol. 20, no. 2, (1994).
- [6] G. M. Chiu, "The odd-even turn model for adaptive routing", IEEE Trans. Parallel and Distributed Systems, vol. 11, no. 7, (2000).
- [7] X. Wang and D. Xiang, "A cost-effective inter connect architecture for interconnection network", IETE J. of Research, vol. 59, no. 2, (2013).
- [8] G. Ascia, V. Catania, M. Palesi and D. Patti, "Implementation and analysis of a new selection strategy for adaptive routing in networks-on-chip", IEEE Trans. on Computers, vol. 57, no. 6, (2008).
- [9] J. Flich and D. Bertozzi, "Designing network-on-chip architectures in the nanoscale era", Chapman and Hall/CRC, (2011).
- [10] J. Duato, S. Yalamanchili and L. Ni, "Interconnection networks: an engineering approach", IEEE Press, (1997).
- [11] W. J. Dally and B. Towles, "Principles and practices of interconnection networks", Morgan Kaufmann Press, San Francisco, (2004).
- [12] B. Grot and S. W. Keckler, "Scalable on-chip inter connect topologies", Proceedings of International Workshop on Chip Multiprocessor Memory Systems and Interconnects, Phoenix, Arizona, USA, (2008) June 21-25.
- [13] J. Hu and R. Marculescu, "DyAD-smart routing for networks-on-chip", Proceedings of ACM/IEEE Design Automation Conference, San Diego, CA, United states, (2004) June 7-11, pp. 260-263.
- [14] B. Fu, Y. H. Han and H. W. Li, "An abacus turn model for time/space efficient reconfigurable routing", Proceedings of International Symposium on Computer Architecture, San Jose, CA, United states, (2011) June 4-8.
- [15] J. Flich, P. L. Pedro, M. Malumbres, J. Duato and T. Rokicki, "Combining in-transit buffers with optimized routing schemes to boost the performance of networks with source routing", Proceedings of International Symposium on High Performance Computing, Tokyo, Japan, (2000) October 16-18.

## Authors



**Xinyu Wang**, Female, born in 1985, she received her doctoral degree from the Department of Computer Science and Technology in Tsinghua in July, 2013, and her research areas include parallel and distributed computing, and routing design in network on chip. Email address: Distribute\_2008@163.com.



**Nan Li**, Female, born in 1982, she received her doctoral degree from the College of Management Science and Engineering, Dongbei University of Finance and Economics in June, 2012, and her research areas include Science Evaluation, Technical Innovation, and Risk Management. Email address: linan\_123456@163.com.