Bi-Routing: A 3D Bidirectional-channel Routing Algorithm for Network-based Many-core Embedded Systems

Wen-Chung Tsai[1] Yi-Yao Weng[2] Chun-Jen Wei[3]
Sao-Jie Chen[2,3] Yu-Hen Hu[4]

[1] Department of Information and Communication Engineering, Chaoyang University of Technology
Taichung 413, Taiwan, ROC
azongtsai@cyut.edu.tw

[2] Graduate Institute of Electronics Engineering, National Taiwan University
Taipei 106, Taiwan, ROC
niles90221@gmail.com, csj@cc.ee.ntu.edu.tw

[3] Department of Electrical Engineering, National Taiwan University
Taipei 106, Taiwan, ROC
d92921022@ntu.edu.tw, csj@cc.ee.ntu.edu.tw

[4] Department of Electrical and Computer Engineering, University of Wisconsin, Madison
Madison 53706, WI, USA
hu@engr.wisc.edu

Received 23 August 2013; Revised 10 November 2013; Accepted 16 December 2013

Abstract. Network-on-Chip (NoC) is an emerging technology designed for the communication of IPs in an embedded system. This paper proposes a 3D (Three-Dimensional) model for a Bi-directional NoC (BiNoC). This three-dimensional model inspires the development of a new routing algorithm for BiNoC, called Bidirectional Routing (Bi-Routing). Bi-Routing is a fully adaptive routing algorithm using different layers in the proposed three-dimensional model to avoid deadlock without prohibiting the use of any path. As such, Bi-Routing can improve the load balance and reduce the packet latency of an NoC. Compared with existing routing methods, experimental results demonstrated Bi-Routing's superior performance with admissible area, power, and timing overheads.

Keywords: Three-Dimensional (3D), Network-on-Chip (NoC), Bidirectional Channel, Routing Algorithm

1 Introduction

System-on-Chip (SoC) uses numerous kinds of Intellectual Properties (IPs) and interconnections to form an embedded system in a single chip. As the technology progresses, the number and operating frequency of IPs are increasing. The bottleneck has transferred from IPs to interconnections. For example, with the deep sub-micron integrated circuit technology, crossing a chip with a highly optimized interconnects takes between six to ten clock cycles and only one set of IPs can use the traditional bus-based interconnection to transact data, such that the rest numerous IPs are waiting for the using right. Therefore, a new approach to designing the communication subsystem between IPs, Network-on-Chip (NoC), has been proposed in the past years to meet the design productivity and signal integrity challenges of next-generation system designs [1], [2], [3].

Routing is to decide which path a packet is to deliver. In other words, given a source and a destination, routing directs a packet where to go. A bad routing algorithm will let numerous packets pass through the same path or choose a longer path. We realize that the same route will lead to the lack of path diversity, and it creates a large load imbalance in the network. So path diversity provided by the adopted routing algorithm determines the performance of an NoC greatly. Most important of all, a deadlock will cause the on-chip interconnection crashed. Thus, routing algorithms must be deadlock-free.

A Bidirectional-channel Network-on-Chip (BiNoC) architecture was proposed to enhance the performance, quality-of-service, and fault-tolerance of on-chip communications [4], [5], [6], [7]. BiNoC allows each communication channel to be dynamically self-configured to transmit flits in either direction in order to better utilize on-chip hardware resources. However, the conventional routing methods adopted in these BiNoC studies cannot fully exploit the path diversity of the BiNoC architectures. Accordingly, we present a three-dimensional model of BiNoC and a new routing algorithm for BiNoC called Bidirectional Routing (Bi-Routing). Bi-Routing can reduce packet latency and achieve higher bandwidth utilization due to its high path diversity by conditionally making channel bidirectional. Moreover, deadlock-freedom is provided which will be introduced in detail in this paper.

The rest of this paper is organized as follows. In Section 2, we will first introduce the background about BiNoC and some deadlock-free routing algorithms. Section 3 will describe a 3-dimensional model of BiNoC and a routing algorithm based on the 3-dimensional model will be presented. In Section 4, we will show the experimental results with analysis of performance simulation and implementation overhead. Finally, Section 5 will draw a conclusion.

2 Background

First, we will introduce Bidirectional-channel Network-on-Chip (BiNoC) in Section 2.1. Next, Section 2.2 will compare several deadlock-free routing algorithms.

2.1 Bidirectional-channel Network-on-Chip

In a conventional router, all channels are unidirectional, thus it may lead to the following scenario where one output channel is busy or in congestion and another channel is idle because the direction of the idle channel is an input channel. BiNoC was proposed to overcome this problem by make all channels bidirectional and to allow each communication channel to be dynamically self-configured to transmit flits in either direction. For example, as shown in Fig. 1(a), every vertex represents a task with a value t_j of its computation time, and every edge represents the computing dependence with a value of communication volume. A mesh NoC with most optimized mapping is shown in Fig. 1(b). We can find that the NoC only use three unidirectional channels with the other directional channels being idle. However, if we make the same mapping solution on BiNoC which can dynamically change the direction of each channel between each pair of routers as shown in Fig. 1(c), the bandwidth utilization will be improved and the total execution time can be reduced.

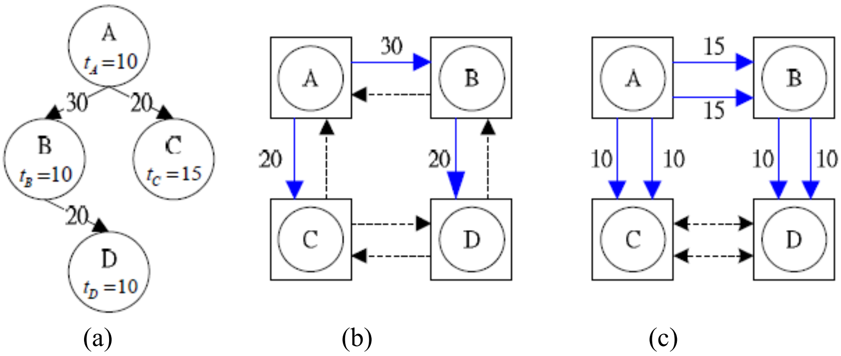


Fig. 1. Example of (a) Task Graph Mapping to (b) a Conventional NoC, and (c) a BiNoC

Fig. 2 indicates a timing analysis that conventional NoC needs 80 cycles to execute in Fig. 1(a). However, if we make the same mapping solution on BiNoC as Fig. 1(c) shows. Due to the improved bandwidth utilization by BiNoC, the total execution time can be reduced to 55 cycles.

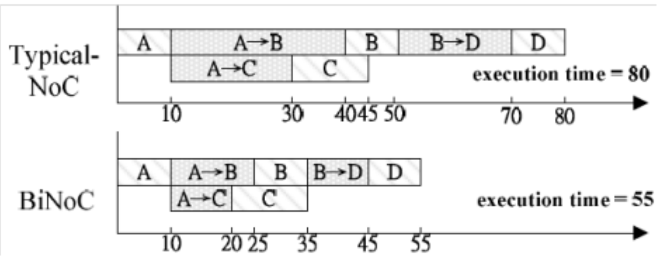


Fig. 2. Detailed Execution Schedules of Typical NoC and BiNoC

BiNoC reconfigures direction of channels by a request-based design as shown in Fig. 3, where the bidirectional channels are connected with two tri-state buffers to control the data conveyed from the bidirectional channel. Furthermore, we also need a Finite State Machine (FSM) to control the availability of each channel, or conflict will occur when both of the routers want to deliver packets. Once the availability signal is asserted, the Switch Controller can use the channel to deliver packets. The FSM will cooperate with another FSM in the neighboring router such that only one direction is used to convey packet.

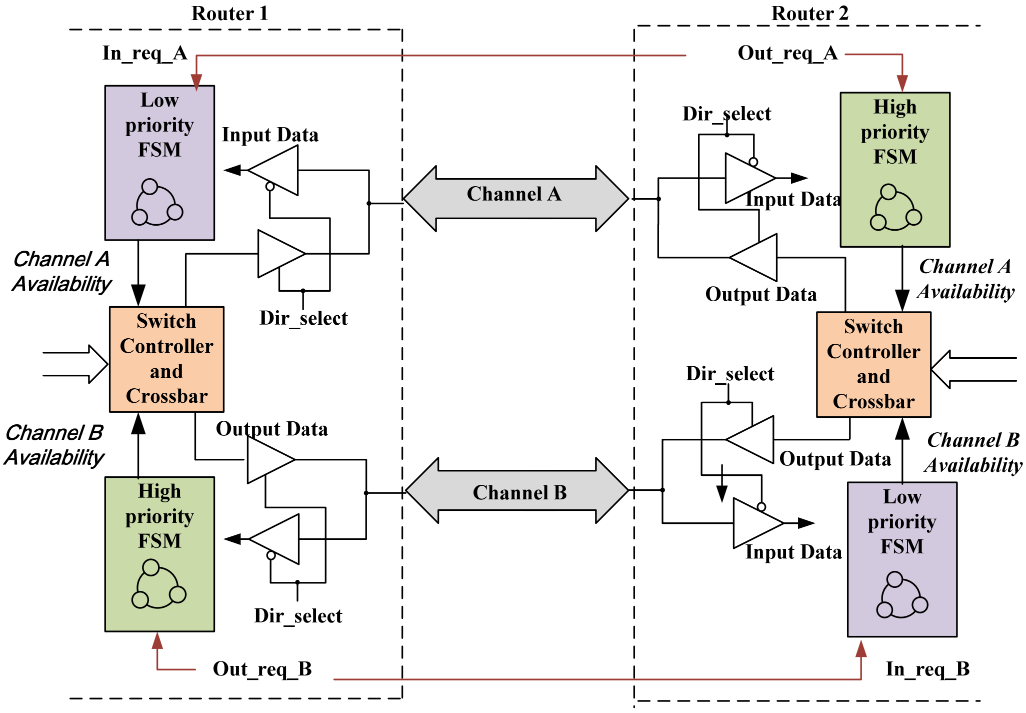


Fig. 3. Bidirectional Channel Interface between Routers

2.2 Related Routing Algorithms

Considering load balance, we prefer to use an adaptive routing algorithm that has more paths to choose for a packet routing delivery. Glass and Ni presented an elegant concept of *turn model* [8]. The basic idea of turn model is to prohibit the minimum number of turns that break all of the deadlock cycles such that routing algorithms based on turn model can be deadlock-free. Three adaptive routing algorithms, namely west-first, north-last, and negative-first were designed based on turn model. We show that the four cases of prohibited turns of the three routing algorithms in Fig. 4. Note that the solid lines indicate the allowed turns and the dash line indicate the prohibited turns. For example, Case two uses the turn model that prohibits S-W turn and N-W turn. According to this turn model, west-first routing delivers all the packets to west first if packets need to be delivered to west. Similar with the west-first routing, negative-first routing and north-last routing were designed according to their own turn models. Turn model provides a simple way to design a deadlock-free adaptive routing. Nevertheless, there is a highly uneven routing path use problem in a global view. That is at least half of the source-destination pairs are limited to having only one minimal path, while full adaptive is provided for the rest of the pairs.

To solve the uneven routing path use problem, an odd-even turn model was presented by Chiu in [9]. With the odd-even turn model, any packet is not allowed to take an E-N turn or an E-S turn at any nodes located in an even column, and any packet is not allowed to take an N-W turn or an S-W turn at any nodes located in an odd column. This odd-even turn model restricts certain turns based on the locations such that none of the turns are eliminated in an NoC. Although the odd-even still restricts some turns for a packet to use, these restricted turns are unobvious in a global view. Therefore, the odd-even turn model has higher path diversity than other turn models. Based on the odd-even turn model, we can design an OE-Routing algorithm, which will compare with our proposed Bi-Routing algorithm in Section 4.

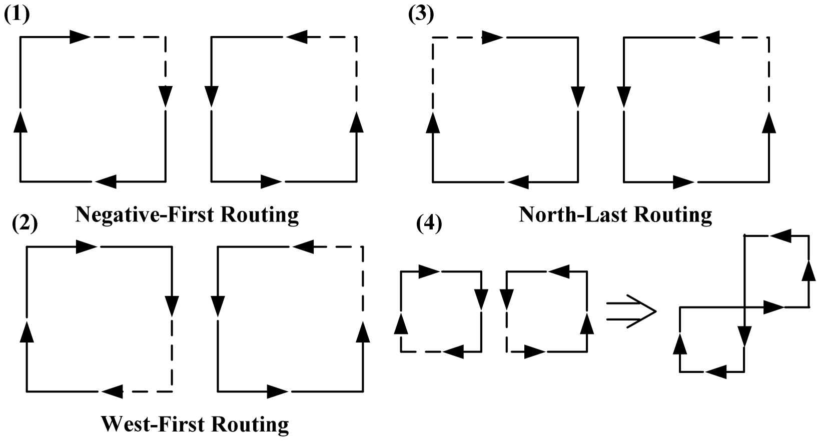


Fig. 4. Four Cases of Turn Models

3 Methodology

In this section, we present the design methodology of our proposed Bi-Routing routing algorithm to exploit the characteristics of bidirectional channels and provide higher path diversity.

3.1 Three-Dimensional Model of BiNoC

Since the original model of mesh NoC cannot show the behavior of BiNoC, we have to represent these four kinds of bidirectional channel patterns as a three-dimensional model in Fig. 5(a). The new Z-dimension is time related, which shows the channel diversity during time changed. The three-dimensional graph as shown in Fig. 5(a) is not a physical three-dimensional IC, but a conceptual model to represent the behavior of a BiNoC. Moreover, as shown in Fig. 5(a), the odd-even turn model in BiNoC can also be represented in our three-dimensional model.

3.2 Bidirectional Routing Algorithm

The three-dimensional model of BiNoC mentioned in Section 3.1 indicates that BiNoC has higher path diversity than the original unidirectional NoC. We use this path diversity to develop a Bi-Routing algorithm for BiNoC in this section. The Bi-Routing idea is shown in Fig. 5(b). On a unidirectional NoC, a deadlock cycle formed by the paths on the same layer can be broken by using another layer of channel (in the Z-dimension). Therefore, we need not prohibit any turn and all paths can be included in the feasible routing set of Bi-Routing. We develop the Bi-Routing based on Theorem 1 brought up in [10].

Theorem 1: A connected and adaptive routing function R for an interconnection network I is deadlock-free, if there are no cycles in its channel dependency graph.

A channel dependency graph D for a given interconnection network I and routing function R is a directed graph, D = G(C, E). The vertices of D are the channels of I. An arc (e_k) in D is a pair of channels (c_i, c_j) where there exists a direct dependency from c_i to c_j. The meaning of connected routing function is that for any packet, the connected routing function can find a path to deliver the packet to the destination. Therefore, from Theorem 1, if we can break the cycle in a channel dependency graph, the routing algorithm is deadlock-free. Hence, three rules are brought up for our Bi-Routing algorithm.

Rule 1: Packets use reverse channel at the E-S turn and the E-N turn.

To escape from deadlock in a BiNoC by using another layer to route, as shown in Fig. 5(b), we choose E-S turn and E-N turn as a breaking position (using reverse channel in another layer) in clockwise and counter-clockwise cycles.

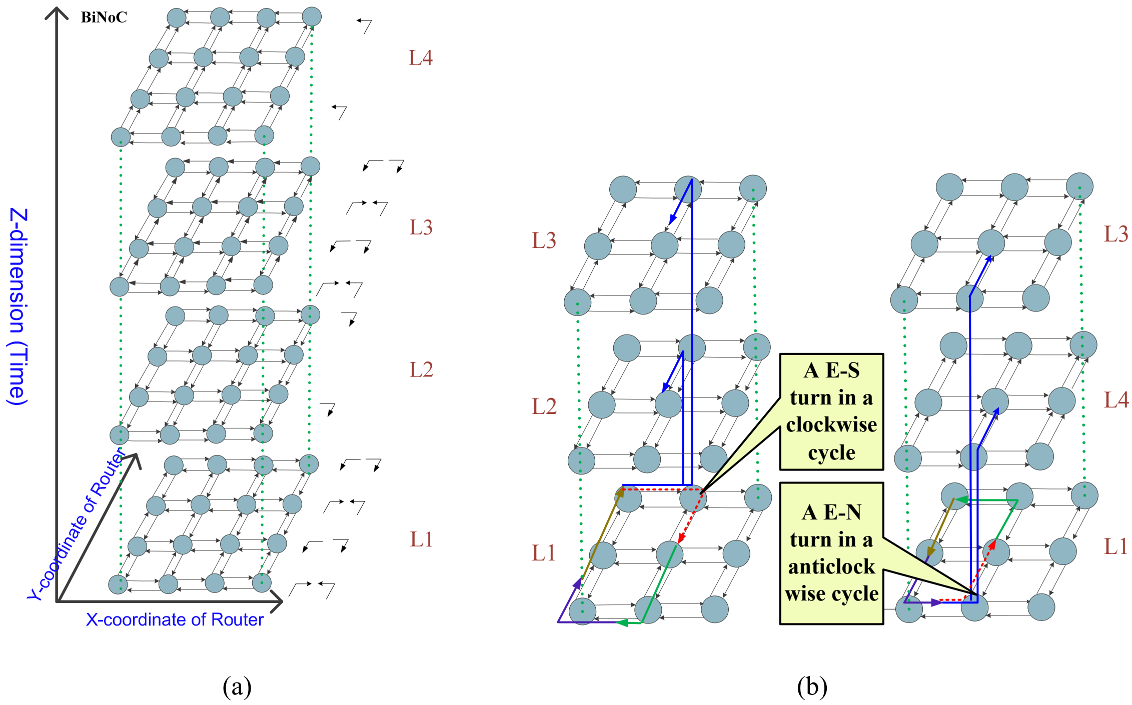


Fig. 5. (a) BiNoC Three-Dimensional Model and (b) Cycles Breaking Example (Rule 1)

Rule 2: Packets from south (north) reverse channel and delivered to north (south) must use reverse channel.

An inter-layer deadlock will appear without Rule 2. Rule 2 indicates that packets should keep using a reserve channel in south or north such that an inter-layer deadlock can be removed as shown in Fig. 6(a). In which, red dotted lines represent paths violating Rule 2 and lead to inter-layer deadlock conditions.

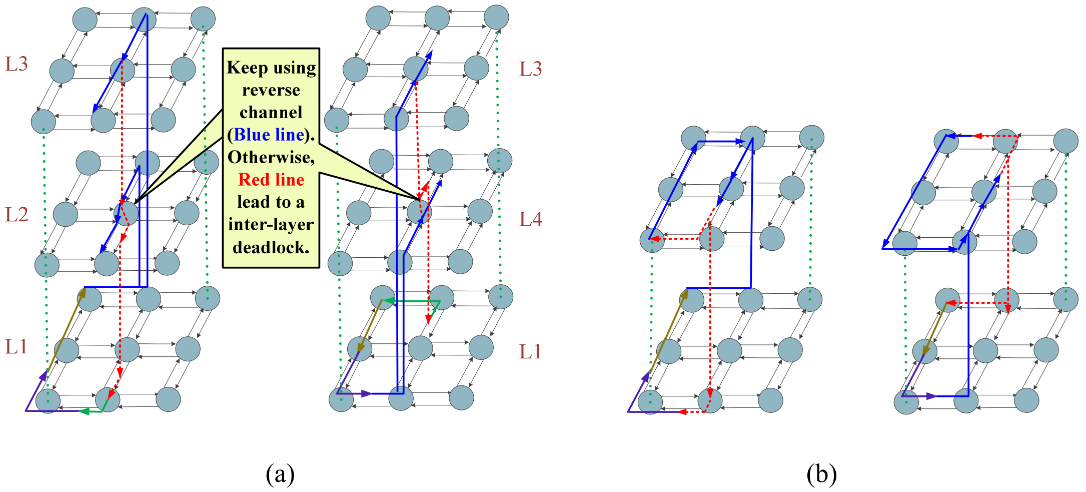

L3
L2
L1
Keep using reverse channel (Blue line). Otherwise, Red line lead to a inter-layer deadlock.
L3
L4
L1
(a)
(b)

Fig. 6. (a) Example of Rule 2 and (b) Example of Rule 3

Rule 3: Packet form reverse channel cannot take S-W or N-W turn.

The essence of Rule 3 is similar to the conventional turn model; it eliminates a turn in just one layer. In other words, reverse channels will make up a cycle, if we do not prohibit packet be routed to a lower layer. With Rule 3, in three-dimensional model, packets cannot take S-W turn and N-W turn when packets are not in the L1 layer as shown in Fig. 6(b). Where the red dotted lines represent prohibiting turns in a higher layer and lead to inter-layer deadlock conditions. With the three rules, Bi-Routing can we provide a fully adaptive routing, which can spread traffic loads to the whole network instead of keeping some parts of the network in heavy congestion.

4 Experimental Results

Our simulation environment comprised an 8x8 mesh. Three synthetic traffic patterns were used, including uniform, transpose, and hotspot traffics. In the uniform traffic, a node receives a packet from any other node with equal probability. Every node transmits packets to a randomized destination with a probability based on the injection rate. In the transpose traffic, a node at a source with coordinate (*i, j*) will sent a packet to a destination with coordinate (*j, i*). In the hotspot traffic, 20% of the packets change their destination to some selected hotspots while the remaining 80% of the traffic keep uniform. In this work, we chose (3, 3), (3, 2), (3, 1), (3, 0) as hotspots. In addition to synthetic traffics, we used E3S benchmarks [11] to demonstrate the performance variations in real traffics.

The presented router architectures were implemented in TSMC 90nm technology using Synopsys Design Compiler with topographical mode under typical operating conditions. The topographical mode of Design Compiler utilizes the Galaxy Design Platform physical implementation technologies to derive accurate interconnect delay data, which allows the Design Compiler solution to predict many post-layout parameters, such as timing, testability, and area during synthesis.

4.1 Performance Evaluation in Synthetic Traffic Patterns

We simulated XY-Routing, west-first routing algorithm (WF-Routing), odd-even routing (OE-Routing), and our proposed Bi-Routing algorithm. The packets in our experiments were composed of 16 flits with one header flit and one tail flit. The capacity of the buffer in each of the 5 directions of channels was 8 flits using wormhole switching. We simulated our network by injecting loads, from 20 flits per clock cycle to 500 flits per clock, at each node. For each injection rate, the simulation time was 25000 clock cycles. The results of latency in three traffic patterns are shown in Fig. 7(a), and the results of throughput are shown in Fig. 7(b).

The simulation results show that our bidirectional routing, Bi-Routing, has the best performance among the four algorithms. XY-Routing outperforms OE-Routing and WF-Routing because XY-Routing can distribute packets evenly in the uniform traffic condition. This part of results is the same as in [8], [9]. Our bidirectional routing algorithm still had better saturation throughput than XY-Routing, about a 6.9% improvement, as shown in Figs. 5(a) and 5(b). However, the throughput of bidirectional routing decreases much more than XY-Routing in high injection rate.

The transpose and hotspot traffic patterns are close to the real-case embedded system traffics, because in a SoC most IPs have communications with the main CPU core. Adaptive routing algorithms perform better than XY-Routing in transpose and hotspot traffic patterns, because adaptive routing algorithms have more paths to route. The Bi-Routing method had 14.78% and 16.51% improvements over the OE-Routing one, in transpose traffic and hotspot traffic, respectively. The reason is our proposed Bi-Routing algorithm can spread traffic loads to relief local traffic congestion. Fig. 8 shows the flits distribution in the network.

Fig. 8 shows the flits distribution at 0.288 injection rate in uniform traffic pattern. The coordinates in the graph is the position in the network. We can observe that 4500 to 5000 flits were routed to Column 3 and Column 5 under OE-Routing algorithm, but our bidirectional routing actually routed packets much evenly than OE-Routing in the network. That is, resources were utilized efficiently under our bidirectional routing.

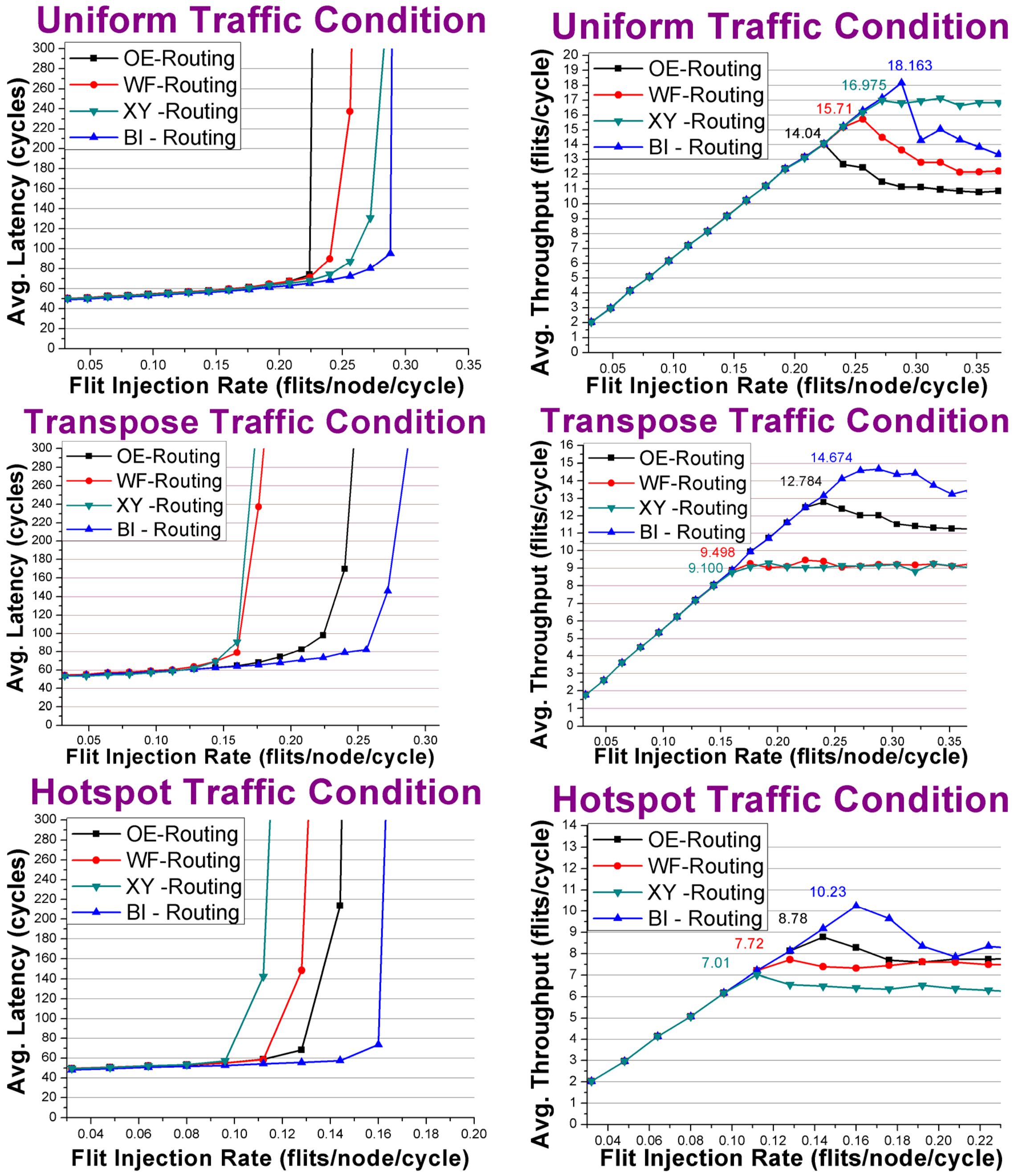
4.2 Performance Evaluation in Real Traffic Patterns

In addition to evaluating Bi-Routing performance with synthetic traffics, we used E3S benchmarks [11] to demonstrate the performance variations in real traffics. The experiments used the same setting as synthetic traffics, but for each of the three adopted real traffic patterns: Auto-indust, Consumer, and Telecom. Real traffic simulation results are shown in Fig. 9. Our Bi-Routing is better than the other algorithms in most situations. However, WF-Routing outperforms Bi-Routing in the Consumer traffic pattern.

4.3 Implementation Overhead

Fig. 10 illustrates the area breakdown on the BiNoC with Bi-Routing scheme. Since the scheme implementation of BI-Routing is more complex than that of OE-Routing, the area of BI-Routing is larger than OE-Routing due to the added routing controls in the Routing Computation unit of the BiNoC router as shown in Figure 10.

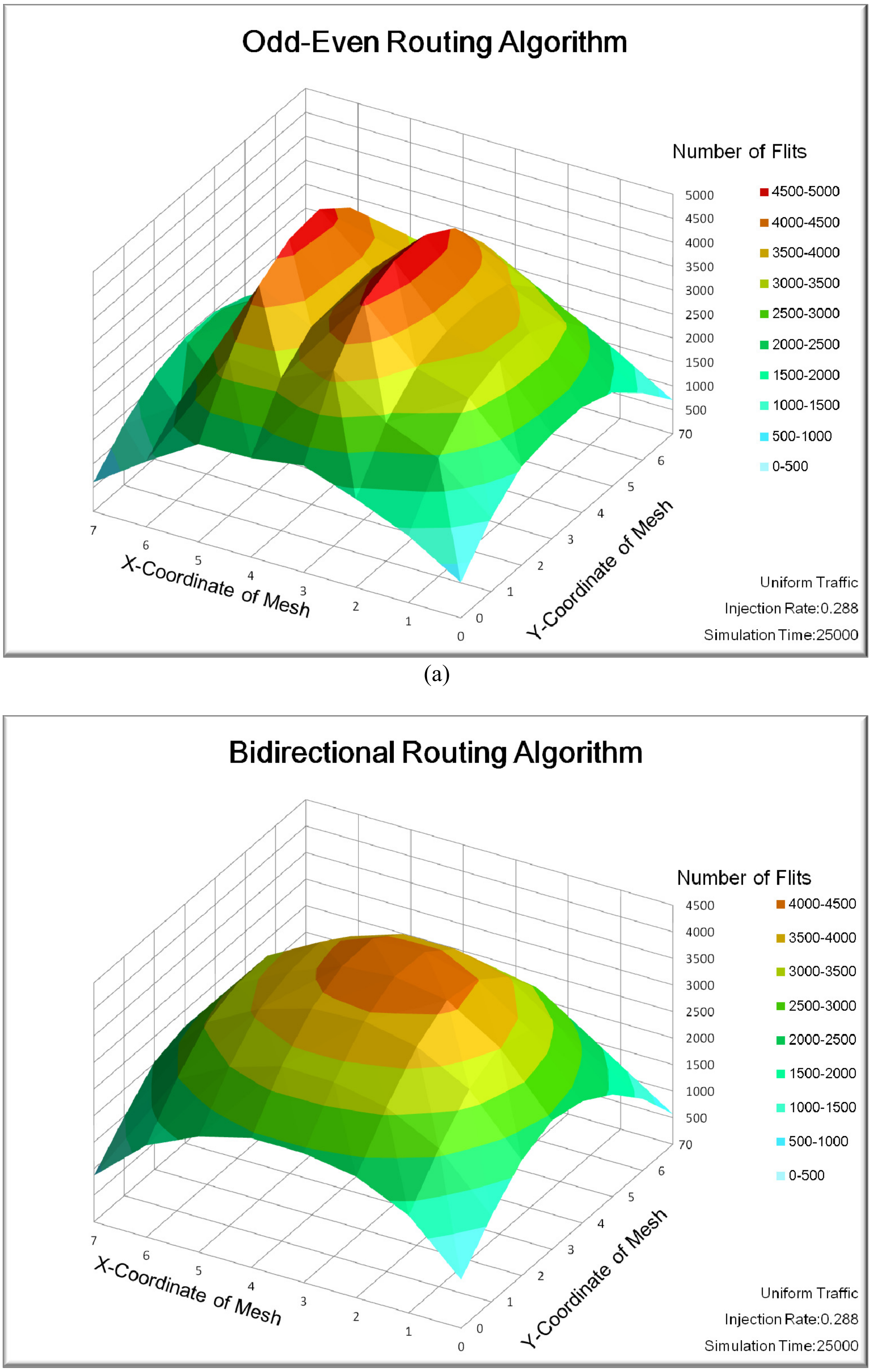
As listed in Table 1, Bi-Routing had 2.94% area overhead because Bi-Routing needs more hardware to implement. Therefore, we propose a design trade-off between area and performance according to the applications. For a low-area design, we can choose OE-Routing as the routing algorithm in BiNoC. For some applications, area of chip is insignificance. In this situation, we can choose Bi-Routing.



(a) (b)

Fig. 7. (a) Latency and (b) Throughput versus Injection Rate under OE-Routing, WF-Routing, XY-Routing, and Bi-Routing

Table 2 lists the power consumptions of OE-Routing and Bi-Routing. Since Bi-Routing is composed of more hardware, its power consumption was higher than OE-Routing. Besides, the additional hardware of Bi-Routing did not increase the delay timing of the critical path of the original BiNoC implementation, which is the path from the shift registers in an input buffer to the switch allocator.



(a)

(b)

Fig. 8. Flit Distribution Graph under (a) Odd-Even Routing Algorithm and (b) Bidirectional Routing Algorithm

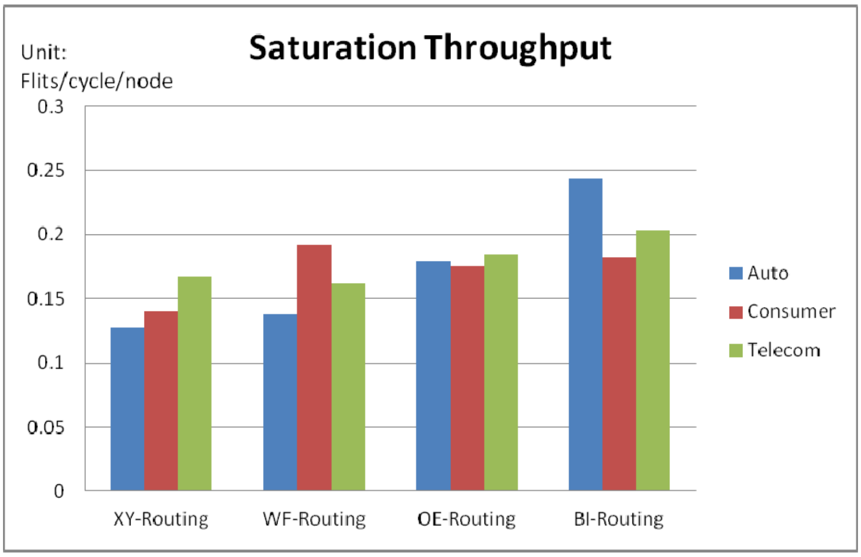


Fig. 9. Saturation Throughputs with XY-Routing, WF-Routing, OE-Routing, and Bi-Routing in Real Traffic Patterns

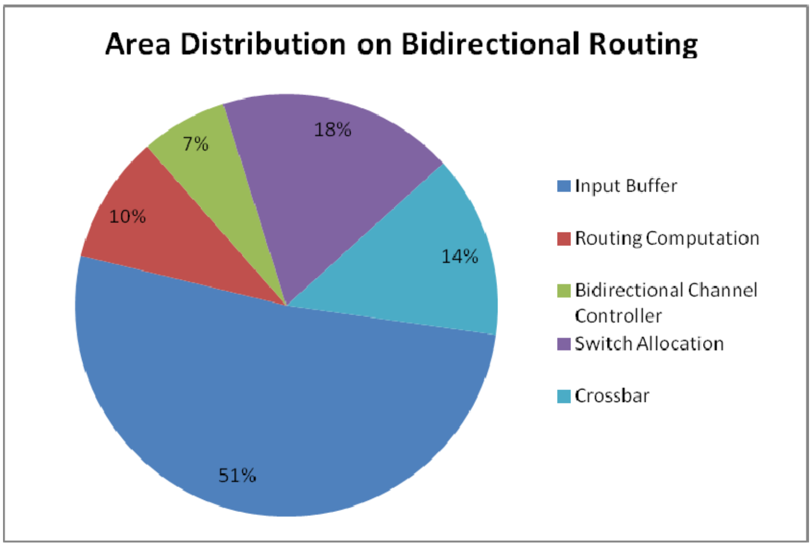


Fig. 10. Area Breakdown of BiNoC Architecture with Bi-Routing Scheme

Table 1. Area Overhead for the Bi-Routing Scheme Implementation

Architecture	Area	Overhead
BiNoC with OE-Routing	71913.35	-
BiNoC with Bi-Routing	74032.27	2.94%

Table 2. Power Overhead for the Bi-Routing Scheme Implementation

Architecture	Power (mW)	Overhead
BiNoC with OE-Routing	7.61	-
BiNoC with Bi-Routing	7.80	2.50%

5 Conclusion

In this paper, we proposed a three-dimensional (3D) model of BiNoC. Based on this 3D model, we developed a new routing algorithm for BiNoC called Bi-Routing. Bi-Routing used the reversed channel to break the deadlock cycle in BiNoC, if any. Experimental results showed that our proposed Bi-Routing delivers better performance over the original BiNoC with OE-Routing because of the increased path diversity and the enhanced load balance provided by Bi-Routing. Moreover, the implementation overhead of area, power, and timing makes minor impacts to the original design.

Acknowledgement

This work was partially supported by National Science Council, ROC, under grants NSC-102-2218-E-324-001 and NSC-102-2218-E-002-011.

References

[1] W.J. Dally, B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," in *Proceedings of the Design Automation Conference*, pp. 684-689, 2011.

[2] L. Benini, G. De Micheli, "Networks in Chips: A New SoC Paradigm," *IEEE Computer*, Vol. 35, No. 1, pp. 70-78, 2002.

[3] A. Jantsch, H. Tenhunen, I. Ebrary, *Networks on Chip*, Kluwer Academic Publishers, 2003.

[4] Y.C. Lan, S.H. Lo, Y.H. Hu, S.J. Chen, "BiNoC: A Bidirectional NoC Architecture with Dynamic Self-Reconfigurable Channel," in *Proceedings of the 3rd ACM/IEEE International Symposium on Network-on-Chip*, pp. 266-275, San Diego, U.S., 2009.

[5] Y.C. Lan, H.A. Lin, S.H. Lo, Y.H. Hu, S.J. Chen, "A Bidirectional NoC (BiNoC) Architecture with Dynamic Self-Reconfigurable Channel," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 20, No. 3, pp. 427-440, 2011.

[6] S.H. Lo, Y.C. Lan, H.H. Yeh, W.C. Tsai, Y.H. Hu, S.J. Chen, "QoS Aware BiNoC Architecture," in *Proceedings of the 24th IEEE International Parallel & Distributed Processing Symposium*, pp. 1-10, Atlanta, U.S., 2010.

[7] W.C. Tsai, D.Y. Zheng, S.J. Chen, Y.H. Hu, "A Fault-Tolerant NoC Scheme Using Bidirectional Channel," in *Proceedings of the 48th Design Automation Conference*, pp. 918-923, San Diego, U.S., 2001.

[8] C.J. Glass, L.M. Ni, "The Turn Model for Adaptive Routing," *Journal of the ACM*, Vol. 41, No. 5, pp. 874-902, 1994.

[9] G.M. Chiu, "The Odd-Even Turn Model for Adaptive Routing," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 11, No. 7, pp. 729-738, 2000.

[10] W.J. Dally, C.L. Seitz, "Deadlock-free Message Routing in Multiprocessor Interconnection Networks," *IEEE Transactions on Computers*, Vol. C-36, No. 5, pp. 547-553, 1987.

[11] R. Dick, "Embedded System Synthesis Benchmark Suites (E3S)," *URL: < http://ziyang.eecs.umich.edu/~dickrp/e3s/* >.