

A Comparative Study on NoC Transpose Traffic

Minghua Tang^{1,*} and Jing Lin²

¹School of Internet Finance and Information Engineering, Guangdong University of Finance, Guangzhou, 510521, China

²School of Information Management, Shanghai Lixin University of Accounting and Finance, Shanghai 200000, China

*Corresponding author

Abstract—In the research of Network-on-Chip (NoC), network performance is evaluated by a lot of simulations. In these simulations, traffic scenarios play an important role. A wide range of traffic scenarios, such as uniform random, transpose, tornado, etc. have been considered. However, there is lack of comparative study of these traffic scenarios. In this paper, simulations are carried out under a wide range of traffic scenarios. The simulation results show that if a routing has good performance under both transpose1 and transpose2 traffics then it will not have poor performance under any other traffic scenario, with high probability.

Keywords—NoC; routing algorithm; traffic scenario

I. INTRODUCTION

With the rapid improvement of semiconductor technology, a huge number of processing cores could be integrated into a single chip. For those complex systems, the communication interconnects are the key to improve the system performance. Toward this end, Network-on-Chip has been proposed [1, 2, 3, 4] in order to handle the SoC communication requirements.

NoC architectural parameters consist of both architecture parameters (i.e., network topology and links) and communication parameters (i.e., switching and routing strategies) as well. For the mostly preferred 2D mesh topology, a great number of routing algorithms have been proposed. When no VCs are used, a part of turns have to be prohibited in order to avoid deadlock.

Consequently, the routing algorithms are referenced by the turn model based routings. Examples of turn model based routing algorithms include Dimension Order Routing (DOR) [5, 6], turn model [7,8], Odd-Even (OE) turn model [9], RTM [10], etc. Apart from these well-known routing algorithms, other routing algorithms could be constructed by taking advantage of Divide-Conquer approach [11].

The performance of routing algorithms is usually evaluated by simulations. During these simulations, a lot of traffic scenarios have to be considered in order to compare different routing algorithms. The considered synthetic traffic scenarios include random, transpose1, transpose2, tornado, bitcompare, bitreversal, asymmetric, hot-spot, etc. The considered realistic traffic includes multimedia system [12].

Although a huge number of traffic scenarios have been considered, the considered traffic scenarios only are a tiny part of the possible traffic scenarios. When a routing algorithm is asserted to have good performance under the well-known traffic scenarios after simulations, it still cannot be assured that

it will not have poor performance under other undetected traffic scenarios.

In this paper, a large number of simulations are conducted to study the effectiveness of traffic scenarios in evaluating routing algorithms. The simulations are carried out more than one hundred traffic scenarios. Simulation results show that if a routing has better performance under both transpose1 and transpose2 traffics then it will not have poor performance under a large number of traffics, with high probability.

II. RELATED WORK

Avoiding deadlock is one of most important tasks of designing routing algorithms. In 2D mesh topology, if no VCs are used, prohibiting a proper group of 90 degree turns is needed to avoid deadlock. The number and location of the prohibited turns which have significant impact on system performance are the main research topic of routing algorithms.

In Dimension Order Routing (DOR) [5, 6], half of the allowed turns are prohibited, which makes that only one deterministic path is left for every communication pair. It is pointed out that prohibiting a quarter of the allowed turns is sufficient to avoid deadlock [7, 8]. Later, it is discovered that prohibiting different turns at odd and even columns could improve the routing performance [9]. The distributions of the prohibited turns are systematically studied in research [10]. After all possibilities are detected, the best one is proposed as RTM routing. With Divide-Conquer approach [11], all kinds of routing algorithms could be constructed. Furthermore, all routing algorithms for 2D mesh networks that are smaller than 5×5 could be constructed.

III. TRAFFIC ANALYSIS

In this paper we focus on 2D mesh topology which is abstracted as a 2D coordinate system. For an $n \times n$ 2D mesh network, the number of nodes is n^2 . The traffic scenarios for $n \times n$ 2D mesh network are defined as following:

Definition 1. Traffic matrix(Λ) : Any doubly-stochastic matrix that has row and column sums of exactly one, where entry λ_{ij} represents the fraction of traffic traveling from node i to node j .

Definition 2. Permutation matrix(Γ) : A traffic matrix whose entries are either 0 or 1.

For an $n \times n$ 2D mesh network, its permutation matrix is $n^2 \times n^2$ matrix. The total number of permutation matrix is $n^2!$.

For example, for 5×5 2D mesh network, its total number of permutation matrix is 25!.

According to [13], any doubly-stochastic traffic matrix Λ could be expressed as a weighted combination of a group of permutation matrices. Consequently, we only consider permutation matrix in this paper.

In 2D mesh NoC, there are a number of well-known and widely referenced traffic scenarios, such as random, transpose, tornado, bitcompare, bitreversal, asymmetric, hot-spot, etc. These considered traffic scenarios only occupy a tiny part of the permutation matrices. When a routing algorithm is asserted to have good performance under the well-known traffic scenarios after simulations, it still cannot be assured that it will not have poor performance under other undetected traffic scenarios.

In 2D mesh NoC, some traffic scenarios, such as butterfly, can be applied only on particular network sizes (power of 2). While for certain traffic, for example uniform, a routing algorithm that has good performance under this traffic may have very poor performance under other traffics. For example, XY routing has the best performance under uniform traffic. However, XY routing has very poor performance under other traffic scenarios.

However, for transpose1 and transpose2 traffics, they contain the southeast, northwest, northeast and southwest communications. Furthermore, for most of the real traffics, packets travel in those four directions. Consequently, if a routing works well for transpose1 and transpose2 traffics, it may not work poorly under any other traffic. Then we can expect good performance routings under lots of traffics.

We take a communication pair (1, 34) as an example shown in Figure 1. Although it is not a communication pair in transpose1 traffic, it will not have poor performance if the routing has good performance under transpose1 traffic. Most of the real traffics are composed of a large number of such communication pairs. Although they are not communication pairs of transpose1 or transpose2 traffics, they will not have poor performance if the routing has good performance under both transpose1 and transpose2 traffics, with high probability.

0	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	32	33	34
35	36	37	38	39	40	41
42	43	44	45	46	47	48

FIGURE I. A COMMUNICATION PAIR EXAMPLE

IV. PERFORMANCE EVALUATION

The port buffer size is four flits. The network payload traffic is regulated by the packet injection rate (PIR). For example, each node under PIR of 0.1 injects 10 packets into network every 100 cycles, on average. Each packet has eight flits. The simulator of Noxim [14] supports a wide range of synthetic traffic scenarios, such as Uniform, Transpose, bitcomp, bitrev, diagonal, asymmetric, tornado, etc.

By using Divide-Conquer method [11], six routing algorithms are created. They are referenced by R1, R2, R3, R4, R5, and R6. Their packet delay variations under transpose1 and transpose2 traffic scenarios are depicted in Figure 2 and Figure 3, respectively. Figure 2 and Figure 3 show that the performance of R1, R2, R3, R4, R5, and R6 degrade gradually. Routing R1 has the best performance and Routing R6 has the worst performance.

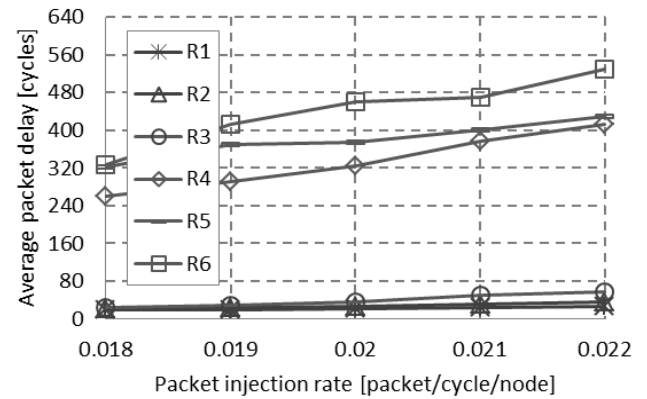


FIGURE II. DELAY VARIATIONS UNDER TRANSPOSE1 TRAFFIC SCENARIO

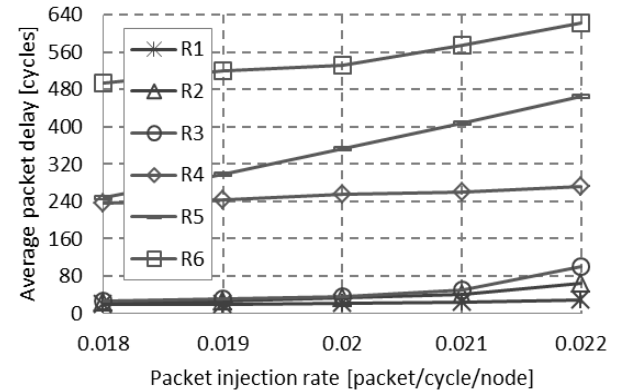


FIGURE III. DELAY VARIATIONS UNDER TRANSPOSE2 TRAFFIC SCENARIO

Figure 4 shows the results for asymmetric traffic scenario. R1 still has the best performance than other five routing algorithms. R5 has the worst performance under this traffic.

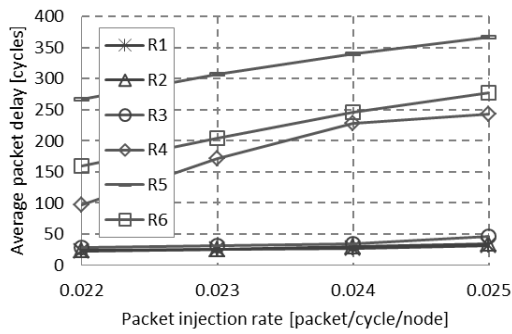


FIGURE IV. DELAY VARIATIONS UNDER ASYMMETRIC TRAFFIC SCENARIO

Figure 5 shows the results for bitcompare traffic scenario. Under this traffic scenario, the performance difference of the six routing algorithms is very trivial.

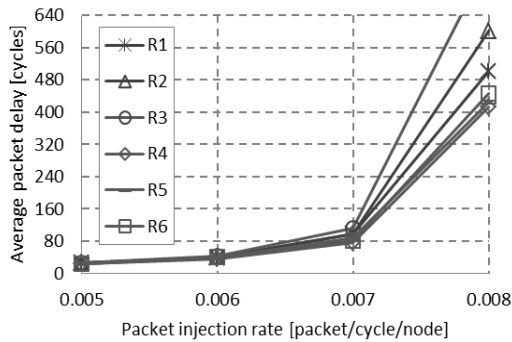


FIGURE V. DELAY VARIATIONS UNDER BITCOMP TRAFFIC SCENARIO

The results for bitreversal are depicted in Figure 6. Under this traffic scenario, both routing R4 and R5 have the best performance. The performance of routing R1 is worse than that of routing R4 and R5.

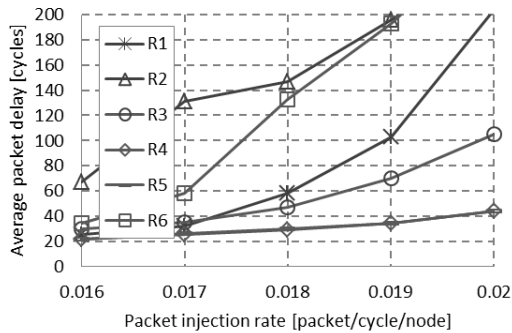


FIGURE VI. DELAY VARIATIONS UNDER BITR TRAFFIC SCENARIO

The results for hs-c are depicted in Figure 7. Under this traffic scenario, the performance difference of the six routing algorithms is very trivial.

Figure 8 shows the results for hs-tr traffic scenario. Routing R5 has the best performance. Routing R3 has the worst performance. Although routing R1 does not has the best performance, it does not has the worst performance either.

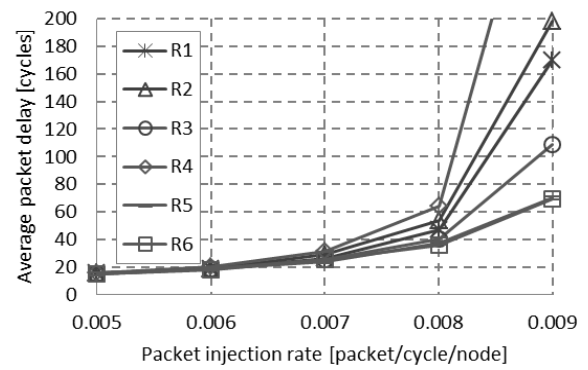


FIGURE VII. DELAY VARIATIONS UNDER HS-C TRAFFIC SCENARIO

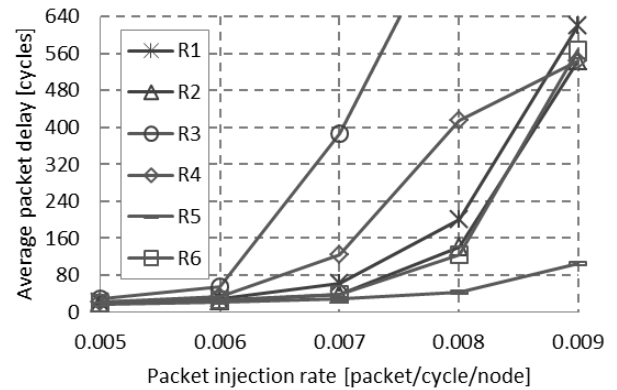


FIGURE VIII. DELAY VARIATIONS UNDER HS-TR TRAFFIC SCENARIO

Figure 9 depicts the results for random traffic scenario. Routing R1 has the best performance. Routing R6 has the worst performance.

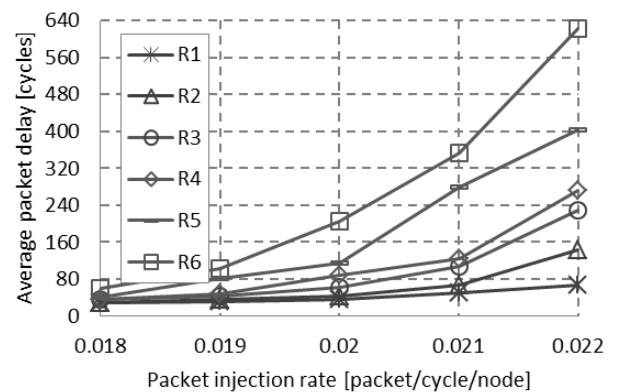


FIGURE IX. DELAY VARIATIONS UNDER RANDOM TRAFFIC SCENARIO

The results for tornado traffic are shown in Figure 10. Routing R1 still has the best performance. Routing R5 has the worst performance.

The results for multimedia traffic are shown in Figure 11. Routing R4 has the worst performance. The performance difference for the remaining five routings is trivial.

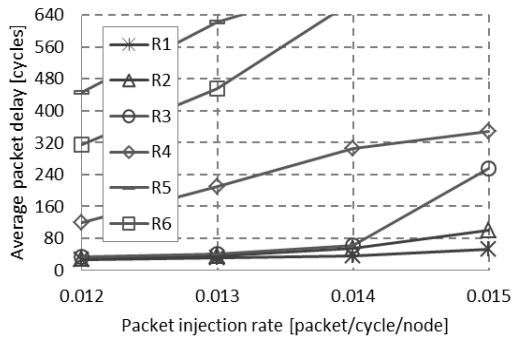


FIGURE X. DELAY VARIATIONS UNDER TORNAO TRAFFIC SCENARIO

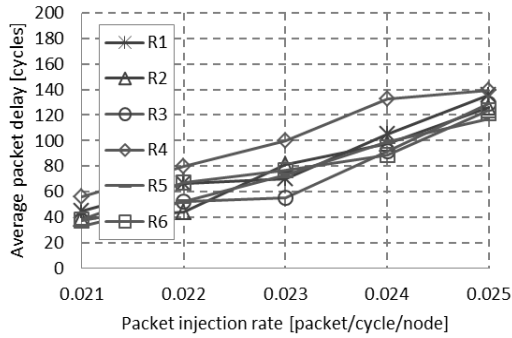


FIGURE XI. DELAY VARIATIONS UNDER MULTIMEDIA TRAFFIC SCENARIO

Apart from these well-known traffic scenarios, 100 permutation matrices are randomly created. Simulations are conducted under the six routing algorithms and the 100 traffics and packet delay is recorded, respectively. Then for each routing algorithm, the average packet delay under the 100 simulations is computed. Figure 12 depicts the average packet delay for the six routing algorithms. It shows that routing R1 has the smallest average packet delay and routing R6 has the largest average packet delay.

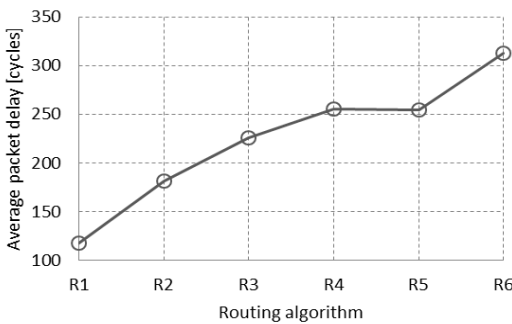


FIGURE XII. THE AVERAGE PACKET DELAY UNDER 100 TRAFFIC SCENARIOS

Routing algorithm R1 has the best performance under both transpose1 and transpose2 traffic scenarios among the six routing algorithms. It still has the best performance under asymmetric, random, and tornado traffics scenarios. Under the other traffics, although it does not have the best performance, it

does not have the worst performance either. When 100 random traffic scenarios are considered, routing R1 still has the best performance. Consequently, from the simulations in this section, one important observation could be made that if a routing has good performance under both transpose1 and transpose2 traffics, it will not have poor performance under any other traffic scenario, with high probability.

V. CONCLUSIONS

In this paper, a wide range of simulations are conducted to compare the performances of six routing algorithms under a large number of traffic scenarios. The considered traffic scenarios include some well-known traffic scenarios and 100 randomly generated traffic scenarios. Simulation results show that if a routing has good performance under both transpose1 and transpose2 traffics, it will not have poor performance under any other traffic scenario, with high probability.

ACKNOWLEDGMENT

This work is supported in part by the Natural Science Foundation of Guangdong under Grants No. 2017A030313391.

REFERENCES

- [1] H. G. Lee, N. Chang, U. Y. Ogras and R. Marculescu, "On-Chip Communication Architecture Exploration: A Quantitative Evaluation of Point-to-Point, Bus, and Network-on-Chip Approaches," *ACM Transactions on Design Automation of Electronic Systems*, vol. 12, pp. 1-20, Aug. 2007.
- [2] W. J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," *Proc. ACM/IEEE Design Automation Conf.*, pp. 684-689, 2001.
- [3] L. Benini and G. D. Micheli, "Networks on Chips: A New SoC Paradigm," *IEEE Computer*, vol. 35, no. 1, pp. 70-78, Jan. 2002.
- [4] F. Martti, "A Scalable High-Performance Computing Solution for Networks on Chips," *Micro*, IEEE. 2002, pp. 46-55.
- [5] W. Dally and B. Towles, "Principles and Practices of Interconnection Networks," Morgan Kaufmann, 2004.
- [6] M. Palesi and M. Daneshmand, "Routing Algorithms in Networks-on-Chip," Springer, ISBN 1461482739, Edition 2014.
- [7] C. J. Glass and L. M. Ni, "The Turn Model for Adaptive Routing," *Proc. 19th Ann. Int'l Symp. Computer Architecture*, pp. 278-287, 1992.
- [8] C. J. Glass and L. M. Ni, "The Turn Model for Adaptive Routing," *J. Assoc. for Computing Machinery*, vol. 41, pp. 874-902, 1994.
- [9] G.-M. Chiu, "The Odd-Even Turn Model for Adaptive Routing," *IEEE Trans. Parallel and Distributed Systems*, vol. 11, no. 7, pp. 729-738, 2000.
- [10] M. H. Tang, X. L. Lin and M. Palesi, "The Repetitive Turn Model for Adaptive Routing," *IEEE Transactions on Computers*, vol. 66, no. 1, pp. 138-146, 2017.
- [11] M. H. Tang, X. L. Lin and M. Palesi, "An Offline Method for Designing Adaptive Routing Based on Pressure Model," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 2, pp. 307-320, 2015.
- [12] G. Varatkar and R. Marculescu, "Traffic Analysis for On-Chip Networks Design of Multimedia Applications," *Proc. 39th Design Automation Conf. (DAC '02)*, pp. 510-517, June 2002.
- [13] G. Birkhoff, "Tres observaciones sobre el algebra lineal," *Univ. Nac. Tucum'an Rev. Ser. A*, vol. 5, pp. 147-151, 1946.
- [14] V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti, "Cycle-Accurate Network on Chip Simulation with Noxim," *ACM Transactions on Modeling and Computer Simulation*, Vol. 27 No. 1, 2016.