Routing Pressure: A Channel-Related and Traffic-Aware Metric of Routing Algorithm

Minghua Tang, Xiaola Lin, and Maurizio Palesi, *Member, IEEE*

Abstract—How to precisely measure performance of routing algorithm is an important issue when studying routing algorithm of network-on-chip (NoC). The degree of adaptiveness is the most widely used metric in the literature. However, our study shows that the degree of adaptiveness cannot precisely measure performance of routing algorithm. It cannot account for why routing algorithm with high degree of adaptiveness may have poor performance. Simulation has to be carried out to evaluate performance of routing algorithm. In this paper, we propose a new metric of routing pressure for measuring performance of routing algorithm. It has higher precision of measuring routing algorithm performance than the degree of adaptiveness. Performance of routing algorithm can be evaluated through routing pressure without simulation. It can explain why congestion takes place in network. In addition, where and when congestion takes place can be pointed out without simulation.

Index Terms—Routing pressure, routing algorithm, metric of routing, channel pressure, network-on-chip

1 Introduction

NoC is presented as a scalable communication architecture for system-on-chip which will integrate several hundred even thousands of processing cores in the near future [1], [2], [3]. The communication efficiency of NoC which is affected by a lot of factors is most important for the whole system.

After the topology of NoC is fixed, routing algorithm plays an important role in determining NoC performance. If one and only one path is selected from source node to destination node by a routing, it is called deterministic routing. An example for mesh topology is Dimension Order Routing (DOR) [4]. Packets are transferred along one dimension then take turns when they do not need to proceed in that dimension. The counterpart of DOR routing for 2D mesh is the XY routing.

On the other hand, adaptive routing may provide more than one path for the packets to reach their destination nodes. If any shortest path can be used to deliver packets, the routing is fully adaptive routing. It might require appropriate virtual channels (VCs) to implement fully adaptive routing in network topologies with cycles such as mesh topology [5], [6]. Without support of VCs, some shortest paths have to be prohibited to implement partial adaptive routing to avoid deadlock. Examples of partial adaptive routing include turn model [7], [8], odd-even (OE) turn model [9], APSRA routing [10], RABC routing [11], IX/Y [12], ABACUS [13] etc.

A routing algorithm designing method may bring about a large number of routings [14], [15]. It takes too much time through simulation to measure performance of those routings. Therefore, a metric that can be computed in a static fashion and which is correlated well with performance is necessary.

The number of paths from source node to destination node is usually used to characterize routing algorithm. On the one hand, it is used to denote the degree of fault tolerance of a routing algorithm. On the other hand, it is used to describe the capability of routing algorithm to avoid congestion. In this paper, we take the second meaning of that concept. The degree of adaptiveness which is the number of paths from source to destination node is taken as a metric of routing [7], [8], [9], [10]. High degree of adaptiveness means that packets are provided more paths to reach their destinations. Thus they would have more chances to avoid congested nodes and arrive at destinations in shorter time.

Nevertheless, our study shows that the degree of adaptiveness might not be a suitable choice as a metric of routing algorithm performance in many cases.

First, the degree of adaptiveness cannot accurately measure performance of routing algorithm. A group of routing algorithms may significantly vary in performance although they have the same degree of adaptiveness. A routing algorithm has higher degree of adaptiveness may have poorer performance under certain traffic patterns. The degree of adaptiveness cannot account for why it happens.

Second, it cannot explain why congestion takes place. Routing algorithm improves performance by avoiding or reducing congestion. However, the relationship between degree of adaptiveness and congestion is not clear. It is difficult to identify where and when congestion will occur.

- *M. Tang is with the Department of Computer Science and Technology, GuangDong University of Finance, Guangzhou 510521, China. E-mail: fractal218@126.com.*
- *X. Lin is with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510275, China. He is also with the Key Laboratory of Digital Life (Sun Yat-sen University), Ministry of Education, Guangzhou 510275, China. E-mail: linxl@mail.sysu.edu.cn.*
- *M. Palesi is with Kore University, Italy. E-mail: maurizio.palesi@unikore.it.*

Manuscript received 31 Dec. 2012; revised 16 May 2013; accepted 21 July 2013. Date of publication 28 July 2013; date of current version 6 Feb. 2015.
Recommended for acceptance by E. Leonardi.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPDS.2013.184

Third, it is difficult to compare the degree of adaptiveness of two routing algorithms. It is nearly impossible for a routing algorithm to have more paths than another for every source-destination pair. Usually, an averaged value of paths from source nodes to destination nodes represents the degree of adaptiveness of a routing [10]. However, much information is lost by the averaged value.

Finally, we cannot know what the maximum packet injection rate is, which can be taken under a routing algorithm with certain degree of adaptiveness.

Under a given topology, the bandwidths of channels are fixed. As routing algorithm varies, the amount of packets passing through a certain channel also changes accordingly. That amount of packets is considered as the pressure imposed on that channel by the routing algorithm.

In this paper, we propose routing pressure to act as the new metric to measure performance of routing algorithm. Difference in performance of routing algorithms can be explained by routing pressure. Routing pressure can account for why congestion takes place and predict where and when congestion is going to occur. It is then meaningful to compare the routing pressures of two routing algorithms. The relationship between the maximum packet injection rate and channel pressure is clarified by a formula.

The rest of the paper is organized as follows: The related work is summarized in the next section. In Section 3, we analyze the degree of adaptiveness as a metric. Then we propose the new metric of routing pressure in Section 4. In Section 5, we exemplify how to measure routing performance by routing pressure. The application of routing pressure in designing routing algorithm is shown in Section 6. Finally, we conclude our paper in the last section.

2 Related Work

Traffic non-uniformity observed on-chip has great impact on system performance. There are lots of causes which could lead to traffic non-uniformity, such as traffic pattern, routing biases, topological artifacts, long range dependence, etc. Numerous attempts have been made to quantify the traffic and its impact on NoC design, performance and optimization [16], [17], [18], [19], [20]. In this paper, we study how to make traffic uniform keeping our focus on routing algorithm.

Given a routing algorithm R, the degree of adaptiveness is defined as the number of shortest paths that can be used to deliver packets from source node to destination node [7], [8]. The degree of adaptiveness of the proposed routing algorithms is thus computed. It has to be calculated one by one for each source-destination pair.

The same definition for degree of adaptiveness is adopted in [9]. The degree of adaptiveness for OE routing in [9] is calculated. Since the degree is independently computed for every source-destination pair, it only draws the conclusion that OE provides more even adaptiveness than turn model.

In the above paper, the degree of adaptiveness is calculated for each source-destination pair. No single metric for the routing algorithms is defined.

In paper [10], the degree of adaptiveness for a source node to its destination node is defined as the ratio of the number of allowed minimal paths to the total number of minimal paths. The adaptiveness for the routing algorithm is the average of the degree of adaptiveness for all the communications. Thus a concrete value of degree of adaptiveness for routing algorithm is calculated. Different routing algorithms can be compared according to the exported value.

In the literature, a number of adaptive routing algorithms have been proposed [13], [22], [23], [24], [25], [26], [27]. Although adaptive properties of their routings are analyzed over and over, the degree of adaptiveness is not quantified.

In paper [21], a runtime selection strategy is proposed to select the optimal output channel. A fitness function which has similarity with channel pressure proposed in this paper is calculated to measure channel status.

3 Motivations

In this paper we focus on two-dimensional mesh topology which is abstracted as a 2D coordinate system. Origin of the coordinate system locates at the top left corner of mesh. The X axis is horizontal direction, and the Y axis is vertical direction. The positive direction of X axis points to east. The positive direction of Y axis points to south.

Every node is identified by its coordinate of a vector (x, y) ($0 \leq x < M$, $0 \leq y < N$), where x and y are its X and Y coordinates, respectively. Furthermore, each node has a unique identifier (ID) which is computed from its coordinates: $ID = y * M + x$.

The simulation environment of this paper is introduced here. We take packet injection rate (*PIR*) as the payload parameter. As an example, *PIR* of 0.2 means that a node injects two packets into the network in ten cycles, on average.

The simulator adopted in this paper is the Noxim simulator [28]. It is an open source simulator and based on SystemC.

The performance metrics of simulation are *average packet latency* and *throughput* which are defined as follows respectively:

$$Average\ packet\ latency = \frac{1}{K}\sum_{i=1}^{K} lat_i$$

where K refers to the total number of packets reach their destinations and lat_i is the latency of *ith* packet.

$$throughput = \frac{total\ received\ flits}{(number\ of\ nodes) * (total\ cycles)}$$

where *total received flits* represents the number of flits which are received by all destinations, *number of nodes* is the number of network nodes, and *total cycles* is the span of the simulation.

In uniform traffic, packets generated at a node are sent to other nodes with the same probability. For a symmetrical network of size N, packets generated at node (i, j) are all sent to node $(\mathrm{N}-1-\mathrm{j}, \mathrm{N}-1-\mathrm{i})$ in transpose1 traffic scenario. In transpose2 traffic, node (i, j) only sends packets to node (j, i).

Each simulation runs for 20000 cycles after 1000 cycles of initialization. Basing on a negative exponential distribution,

TABLE 1
Simulation Configurations

Simulator	*Noxim*
Topology	Mesh-based
Network size	7×7
Port buffer	Four flits
Switch technique	Wormhole switching
Arbitration	Round-Robin
Selection strategy	Random
Traffic scenario	Transpose1, Transpose2 ,uniform
Packet size	Eight flits
Simulation length	20000 cycles
Virtual Channel (VC)	No

a packet is injected into the network. The simulation at each PIR is iterated a number of times to guarantee accuracy. The configurations of simulation are shown in Table 1.

In this paper, we adopt the following definition of degree of adaptiveness.

Definition 1

The adaptiveness for a source-destination pair is the number of shortest paths allowed by a certain routing. For instance, in deterministic routing, there is only one path for any source-destination pair. Therefore, the adaptiveness of every source-destination pair is one.

Definition 2

The adaptiveness of a routing is the summation of adaptiveness for all source-destination pairs.

According to the method proposed in [15], a lot of routing algorithms can be constructed. Consequently, it is possible to completely study routing algorithm due to the abundant research material.

Four routing algorithms are constructed for mesh topology of size 7×7, which are named R1, R2, R3 and R4, respectively. The four routing algorithms have the same adaptiveness.

Fig. 1 shows latency variation of the four routing algorithms, under uniform traffic scenario. Although they have the equivalent adaptiveness, there exists huge difference in their performance. For example, average packet latency is 42, 61, 112 and 1153 for routing R1, R2, R3 and R4 respectively, when PIR is 0.014.

Unfortunately, the degree of adaptiveness cannot account for the huge difference of routing performance.

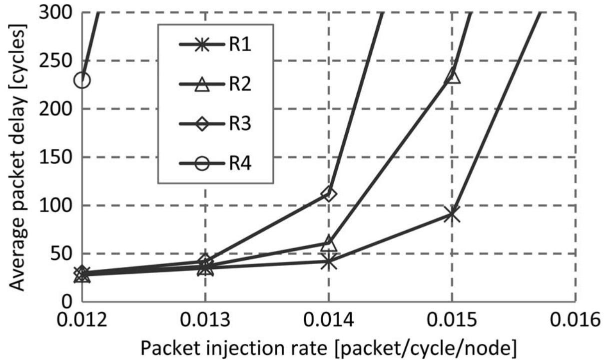

Fig. 1. Latency variations for routings R1, R2, R3, and R4, which have the same adaptiveness.

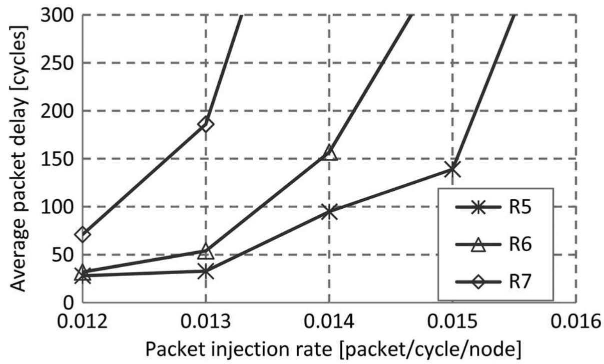

Fig. 2. Latency variations for routings R5, R6, and R7, which have different adaptiveness.

Another three routing algorithms are also obtained by the method in paper [15], which are labeled R5, R6 and R7, respectively. The degree of adaptiveness for R5, R6 and R7 are 16196, 16708 and 17742, respectively. The average packet latency variations for the three routings are depicted in Fig. 2.

High degree of adaptiveness does not bring about high performance. On the contrary, although the adaptiveness of R7 is 9.5 percent higher than R5, performance of R7 is significantly lower than R5.

The degree of adaptiveness still cannot explain why this may happen.

If the routing does not distribute traffic uniformly in the network, congestion will take place in the network. However, the degree of adaptiveness cannot explain why congestion happens in the network and what kinds of paths from source to destination node would lead to congestion.

Although routing algorithms try to make traffic in the network as uniformly as possible, part of the network will accommodate more traffic. However, the degree of adaptiveness cannot predict which part of the network is prone to congestion.

Usually, there is a threshold of PIR for a given routing. Congestion begins to occur when PIR is larger than that threshold. The degree of adaptiveness cannot estimate the value of PIR for a certain routing.

It is difficult to compare the degree of adaptiveness for different routings if adaptiveness is defined at the granularity of each communication pair. Since it is nearly impossible for a routing to have larger degree of adaptiveness than another routing for every communication pair. Only the averaged value is comparable. However, much important information was lost due to the averaged value.

If routing algorithms were not comparable in the meaning of performance metric, the performance gain of a newly designed routing would not be known.

In short, the degree of adaptiveness is not a suitable metric to measure performance of routing algorithm. It is necessary to design a new metric.

4 Routing Pressure

The bandwidth of network channels is fixed when the network is constructed. If packets passing through one channel are beyond its bandwidth, congestion arises.

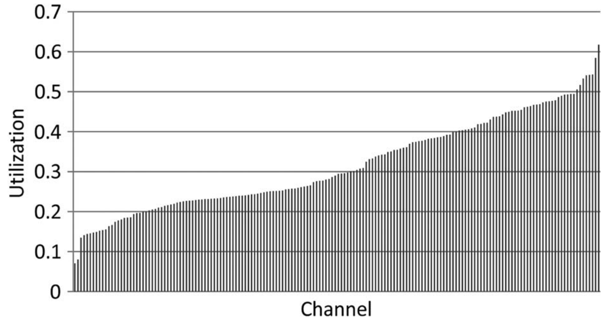

Fig. 3. Utilization ratio of channels under OE routing algorithm and uniform traffic.

4.1 Utilization Ratio of Channels

The traffic load across the NoC is not uniform, although routing algorithm tries to make it uniform [29], [30].

The following simulation adopts the well-known OE (odd-even turn model) routing and uniform traffic scenario. Through the simulation, we write down the number of packets passing through every channel. Then the ratio of that number over the largest processing capability of channel is calculated. The ratio represents utilization of each channel. Fig. 3 shows the value of computed ratios.

The max ratio is 0.6176, the corresponding channel is busier than others during the simulation. The minimum value is 0.0704, the corresponding channel is freer than others during the simulation. The utilization ratios of other channels lie in between the maximum and minimum value.

This example shows that the channel utilization is not uniform under OE routing although traffic is uniform.

4.2 Impact of Traffic Scenario on Routing Performance

A routing algorithm has different performance as traffic scenario varies. Usually, a routing algorithm has good performance under a certain traffic. However, it may have poor performance under another traffic.

Under uniform traffic and without VC support, XY routing performs better than other partial adaptive routings [7], [10]. However, under nonuniform traffic XY routing has poor performance.

The degree of adaptiveness of OE routing is more even than turn model by which three routings are created: west-first, north-last and negative-first. OE routing has better

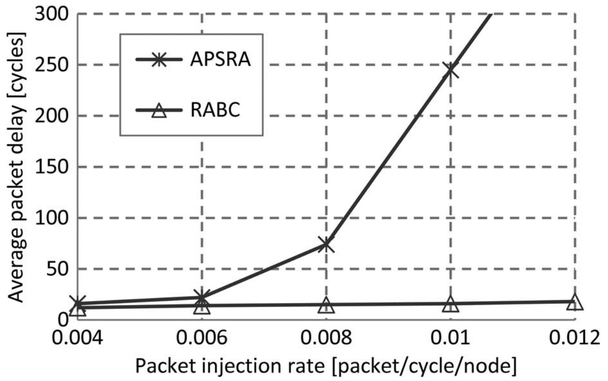

Fig. 4. Latency variation of APSRA and RABC routing algorithm under transpose1 traffic.

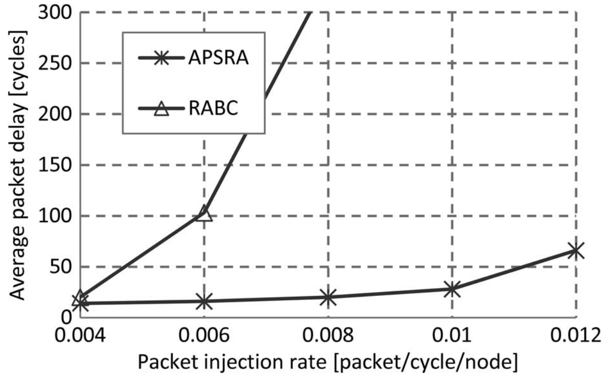

Fig. 5. Latency variation of APSRA and RABC routing algorithm under transpose2 traffic.

performance than negative-first routing under transpose2 traffic. However, it is outperformed by negative-first routing under transpose1 traffic [9].

RABC routing has better performance than APSRA routing under transpose1 traffic as Fig. 4. However, as transpose2 traffic is taken by the network, RABC is outperformed by APSRA, Fig. 5.

Consequently, when measuring performance of NoC routing algorithm, the traffic scenario should be taken into consideration.

4.3 Definition of Channel Pressure

In this paper, we use channel pressure to measure the number of packets each channel has to process. Two motivational examples are presented before we define channel pressure. Fig. 6a depicts a 2 $\times$ 2 network, its routing algorithm is depicted as two prohibited turns.

Suppose there is only one communication in the traffic. Node 0 and node 3 are its source and destination node respectively. Packets from source node are taken as one unit. There is only one path from node 0 to node 3, 0-2-3 labeled by node ID. Consequently, the channel pressures for channels 0-2 and 2-3 is 1, shown in Fig. 6b. The channel pressures of other channels are zero.

Fig. 6c illustrates another routing for 2 $\times$ 2 network Fig. 6a. There is also one communication in the traffic. Under this routing, there are two paths from source node 0 to destination node 3. Packets generated at source will be uniformly distributed between the two paths. The channel pressures of the four channels 0-1, 0-2, 1-3 and 2-3 are 0.5, Fig. 6d.

For network larger than 2 $\times$ 2, channel pressure cannot be computed straightforwardly. The critical step to compute channel pressure is to get the amount of shares

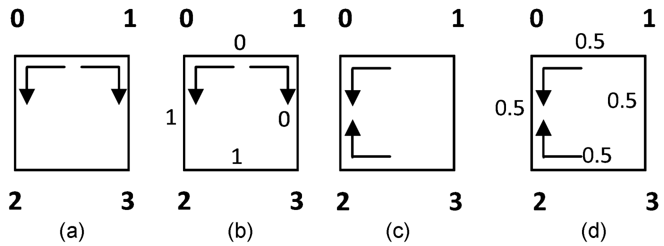

Fig. 6. Example of channel pressure.

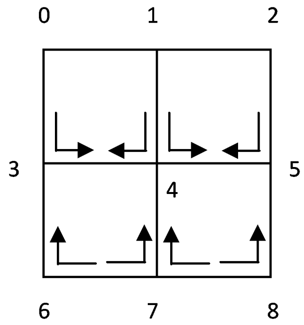

Fig. 7. Network of size 3 × 3 with the prohibited turns.

imposed by one communication pair. After shares imposed by every communication pair on a channel are calculated, channel pressure can be obtained by just adding all those shares.

Given a communication, we create a searching tree to compute shares of packets it impose on each channel. The root of the searching tree is the source node of the given communication. The leaves are the destination node of the communication. A group of continuous branches from root to a leaf represents a path from source node to destination. Consequently, the number of leaves in the tree equals that of paths from source to destination node.

We exemplify searching tree by a 3 × 3 mesh network which routing is shown by the prohibited turns, Fig. 7. The only communication is from node 0 to node 8.

According to the given routing, three paths are allowed from source node 0 to destination node 8. The constructed searching tree is depicted in Fig. 8. Two of the three paths pass through channel 7-8.

Shares of the channel on the searching tree are calculated after the searching tree is set up. The calculated shares for every channel along with the searching tree are shown in Fig. 9. At each level of the searching tree, the summation of shares equals one.

We then define channel pressure formally. Let R be a deadlock-free routing algorithm for a topology TG, and a traffic tr consisting of $COMMs$ communication pairs. Since each communication pair has only one source node and one destination node, src_i and $dest_i$ are used to denote the source node and destination node of the *ith* communication $comm_i$, respectively. Similarly, the packet injection rate for source node src_i is referred to as PIR_i.

Suppose there are p_i paths for the *ith* communication and the packets generated at source node src_i are uniformly distributed across those p_i paths. Then each channel on a path will process PIR_i/p_i of packets coming from the source node. If k paths pass through a channel, that channel will assume $k * PIR_i/p_i$ of the packets from source node.

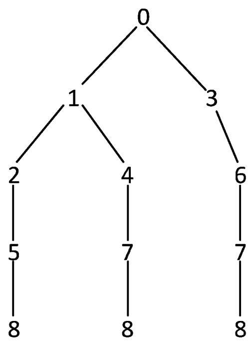

Fig. 8. Example of searching tree.

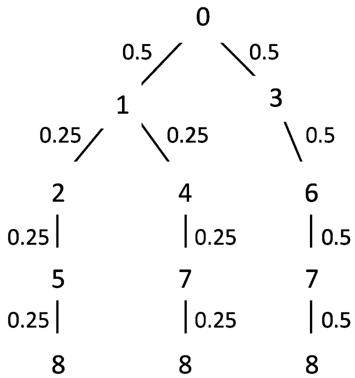

Fig. 9. Shares of packets for channels on searching tree.

For a channel ch, the *ith* communication will impose a certain share of packets on it. That share is referred to as shr_i, where $shr_i = \rho_i * PIR_i$, $\rho_i = k * PIR_i/p_i$.

Definition 3

Given a channel ch, the summation of shares of packets imposed by all communication pairs in traffic tr is called its channel pressure, $prss_{ch}$,

$$prss_{ch} = \sum_{i=1}^{COMMs} shr_i = \sum_{i=1}^{COMMs} \rho_i * PIR_i \tag{1}$$

If all source nodes have the same packet injection rate PIR, channel pressure of ch becomes:

$$prss_{ch} = PIR * \sum_{i=1}^{COMMs} \rho_i \tag{2}$$

Let $\rho = \sum_{i=1}^{COMMs} \rho_i$. Then we have the channel pressure of ch:

$$prss_{ch} = PIR * \rho \tag{3}$$

In this case, channel pressure for channel ch can be denoted ρ for simplicity.

The intuition behind this definition is that a channel with large channel pressure will be imposed more packets. In other word, it is under higher pressure.

4.4 Computation of Channel Pressure

The pseudo-codes for calculating channel pressure is depicted is shown in the supplemental file which is available in the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.184.

The functions need to traverse all paths of a given communication pair. If the communication pair has n paths, their computational complexities are both $O(n)$. Suppose the source node and destination node of the communication pair are (src_x, src_y) and (dst_x, dst_y) respectively. Let $dx = |src_x - dst_x|$ and $dy = |src_y - dst_y|$. If the routing is fully adaptive, we have, $n = (dx + dy)!/(dx! * dy!)$.

For instance, there are 924 paths from source node 0 to destination node 48 in 7×7 network (Fig. 14) under fully adaptive routing. If routing is not fully adaptive, the total number of paths belongs to a communication pair is significantly smaller than the largest allowable value. For example, there are 84 paths from source node 0 to destination node 48 under OE routing.

4.5 Routing Pressure of Routing Algorithm

In the previous section, we propose the definition and calculation of channel pressure for every single channel. In this section, we propose the definition of pressure for a routing algorithm.

The channel with the largest channel pressure will receive the maximum volume of traffic under the given routing algorithm. If the received amount of the messages is beyond its bandwidth capacity, congestion occurs at that channel. Otherwise, congestion will not occur at that channel.

If congestion does not take place at the channel with the largest channel pressure, it will not happen at other channels. Consequently, the largest channel pressure can be used to estimate whether congestion will occur in the network or not. Furthermore, it can be used to measure the performance of routing algorithm.

Definition 4

Given a routing algorithm and a traffic pattern, the routing pressure refers to the largest channel pressure among all channels under the given traffic.

If all source nodes in the traffic have the same packet injection rate (PIR) and a routing has routing pressure of $\rho * PIR$, the routing pressure is represented by ρ for simplicity in the rest of this paper.

The routing pressures for some state-of-the-art routing algorithms are depicted in the supplemental file available online.

5 Using Routing Pressure to Measure Routing Performance

5.1 Correlation Between Routing Pressure and Average Packet Latency

Using the method of paper [15], any specific routing of a mesh network, even all deadlock-free routings, can be created. When two turns are restricted in each 2×2 subnetworks, there are 2529 deadlock-free routing algorithms for 3×3 network. Table 2 shows the correlation coefficient between average packet latency, routing pressure and routing adaptiveness, respectively, in which RP refers to routing pressure, RA refers to routing adaptiveness, APL refers to average packet latency, three traffics are considered. Under each traffic, correlation coefficient between routing pressure and latency is higher than that of routing adaptiveness and latency.

TABLE 2
Correlation Coefficient

	Uniform	Transpose1	Transpose2
RP VS APL	0.56	0.81	0.82
RA VS APL	0.22	0.62	0.64

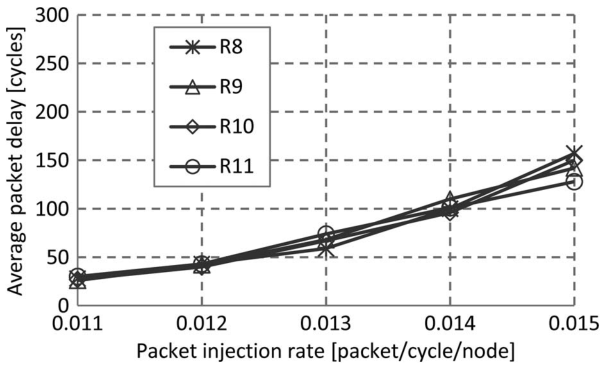

Fig. 10. Latency variations of routings with the same routing pressure.

5.2 Routing Pressure and Performance

Four routing algorithms with the same routing pressure for transpose1 traffic are created for 7×7 mesh network. They are labeled as R8, R9, R10 and R11. The latency variations are shown in Fig. 10. As can be observed in Fig. 10, there is no difference in the performance of the four routings with the same routing pressure.

In the second case, other four routing algorithms with different routing pressure for transpose1 traffic are constructed for 7×7 mesh network. They are labeled as R12, R13, R14 and R15, having routing pressure 5.42, 6.31, 7.15 and 8.28, respectively. Fig. 11 shows the latency variations for the four routings. As routing pressure increases, its performance decreases accordingly.

Several state-of-the-art routing algorithms are chosen as examples to study the relationship between routing pressure and routing performance. They are XY, OE, turn model (negative-first) and APSRA.

For transpose1 traffic, routing pressures for the four routings are 6, 4.81, 6 and 8.86, respectively. The simulation results are shown in Fig. 12. OE has the best performance because its routing pressure is smallest. XY and negative-first (NF) have the same routing pressure. There is no difference in their performance. APSRA has the worst performance due to its largest routing pressure.

For transpose2 traffic, routing pressures for the four routings are 6, 4.81, 2.41 and 5.44, respectively. Fig. 13 shows the simulation results for them. As expected, negative-first has the best performance. Under this traffic,

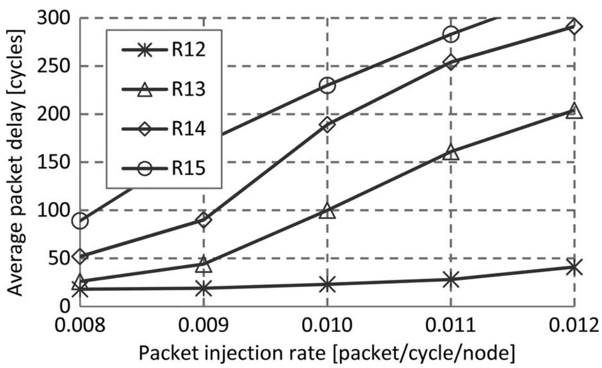

Fig. 11. Latency variations of routings with different routing pressure.

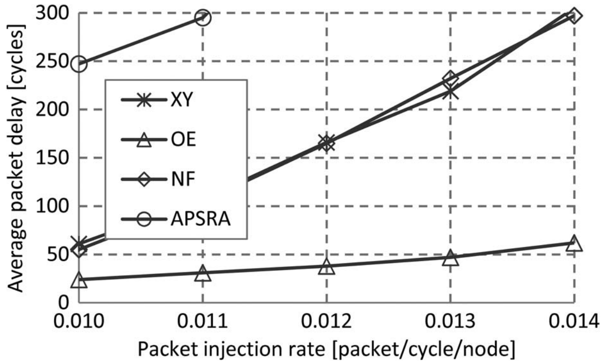

Fig. 12. Latency variations of routings XY, OE, NF, and APSRA for transpose1 traffic.

XY has the worst performance because of its largest routing pressure.

These examples indicate that routing pressure is highly related with its performance. Consequently, routing pressure is a better metric to measure performance of routing algorithm.

5.3 To Address Why Congestion Takes Place

Since the channels are frequently referenced in this section, we depicts a mesh of size 7×7 as an example in Fig. 14. Each node is identified by its ID and each channel is labeled by its start and sink node, for example channel 0-1.

To analyze the relationship between congestion and channel pressure, we take routing R12 when $PIR = 0.012$ as an example. The channel with the largest pressure of 5.42 is channel 17-24.

The average channel utilization ratio of all the network channels is 24 percent. However, the utilization ratio of channel 17-24 is as large as 81.4 percent. It shows that packets passing through channel 17-24 are larger than the other channels. The channel utilization ratio variations along with channel pressure are shown in the supplemental file available online.

In transpose1 traffic, one communication is from node 10 to node 26. Packets generated at node 10 reach destination node 26 after only passing through four channels. However, node 10 generated the largest latency packet among all the packets in the traffic. When computing average packet latency for every single communication pair we get the average packet latency for each communication pair. The average packet latency for communication between node 10 and 26 is still the largest.

Under routing algorithm R12, there is only one path for communication between node 10 and 26, which is 10-17-24-25-26 labeled by node ID. All packets of the communication have to pass a congested channel. It is not surprised packet latency of this communication is large.

Consequently, congestion takes place at the channel which has the largest pressure.

5.4 Using Routing Pressure to Predict Packet Injection Rate

Given a routing pressure, the largest packet injection rate not getting to congestion can be computed.

To avoid congestion the amount of packets imposed on a channel cannot be larger than its bandwidth. Suppose all source nodes in the traffic have the same packet injection rate of PIR, routing algorithm has routing pressure of $\rho * PIR$, network channel can process $frate$ flits per cycle, all packets have the same length of len flits. The simulation runs $time$ cycles. A channel can then process at most $frate * time/len$ packets during the simulation. The number of packets imposed on it is $\rho * PIR * time$. We have,

$$\rho * PIR * time \leq frate * time/len. \quad (4)$$

Then following inequality holds,

$$\rho * PIR \leq frate/len. \quad (5)$$

In this paper, the simulator is Noxim in which channel needs two cycles to forward a flit, that is $frate = 0.5$. Each packet has eight flits, $len = 8$. Given a routing with routing pressure of $\rho * PIR$, to avoid congestion PIR has to satisfy,

$$PIR \leq 0.0625/\rho. \quad (6)$$

For transpose1 traffic, APSRA routing pressure is 8.86. The maximum packet injection rate is 0.007 as shown in Fig. 15. After that congestion begins to occur in the network and system performance degrades significantly.

XY and negative-first routing have the same routing pressure of 6 for transpose1 traffic. The maximum packet injection rate is 0.0104 to avoid congestion, as depicted in Fig. 16.

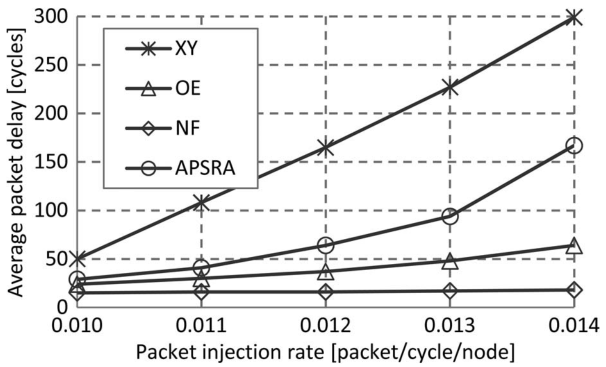

Fig. 13. Latency variations of routings XY, OE, NF, and APSRA for transpose2 traffic.

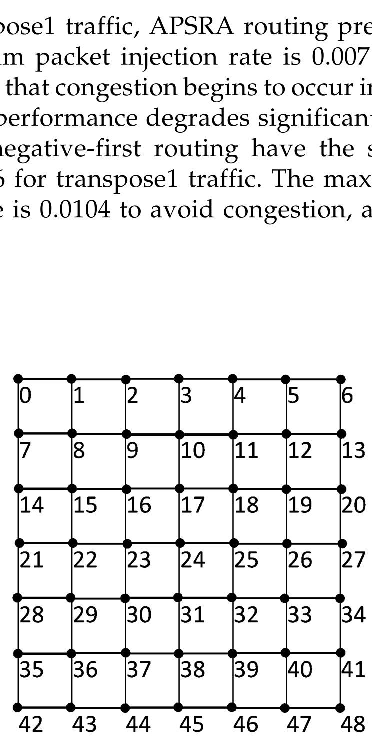

Fig. 14. Mesh topology.

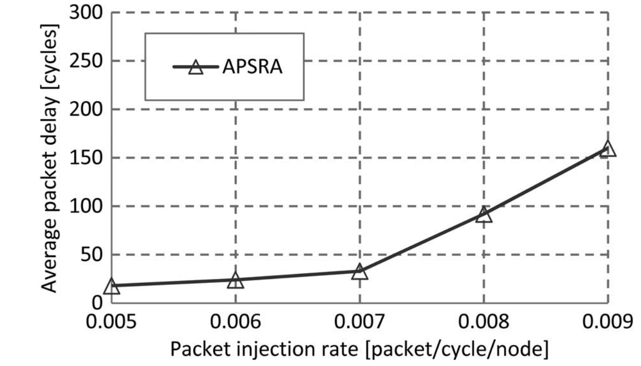

Fig. 15. Maximum packet injection rate under APSRA routing for transpose1 traffic.

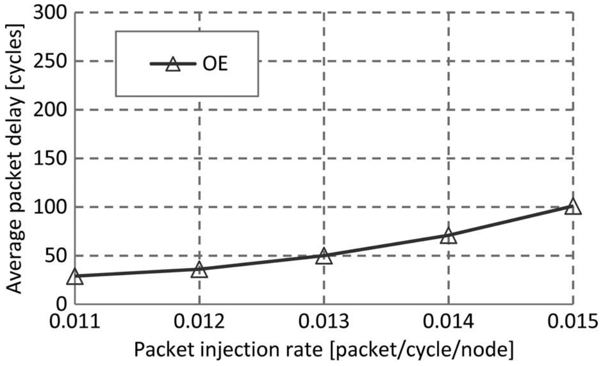

Fig. 17. Maximum packet injection rate under OE routing for transpose1 traffic.

The routing pressure of OE routing is 4.81 for transpose1 traffic. Fig. 17 shows its latency variation. The maximum packet injection rate is 0.013.

The routing pressure can account for why routing algorithms perform differently when PIR is larger than the maximum value. We refer the reader to the supplemental file available online for the detail.

6 Application of Routing Pressure

6.1 Application in Designing Routing Algorithms

It is a quite complex task to study or design an efficient routing algorithm. As pointed out in paper [15], there are as many as 12^{36} routing candidates for network of size 7×7. One should have to consider those huge searching space while studying or designing routings. It may not be feasible to evaluate routing algorithm using simulation method.

Routing pressure could be useful when studying or designing routing algorithm. With this new metric, a routing can be evaluated by computing its routing pressure, avoiding time-consuming simulation.

A routing which has good performance under one traffic may have poor performance under the other traffics. To evaluate a routing, every traffic has to be taken into consideration. It is impossible due to the huge variations of traffics.

Under 2D mesh topology, communications can be roughly classified into four categories according to the relative position of source and destination nodes. In the first category of southeast communication, destination is at the southeast direction of source node. The second one is northwest where destination is at the northwest direction of source node. Similarly, southwest and northeast communications are defined respectively.

In most cases, if a routing has high performance for communications of all the four categories it may have high performance for most traffics. Transpose1 and transpose2 traffics just have the four communications.

Consequently, as our study shows that if a routing performs well under both transpose1 and transpose2 traffic, it may have good performance for a wide range of traffics. This can be further verified by the following examples.

A routing labeled R18 in these examples is build using the method [15]. Its routing pressure is 4.19 under both transpose1 and transpose2 traffics. Under these two traffics, OE has routing pressure of 4.81. R18 outperforms OE for the two traffics. Moreover, it has better performance than OE for a lot of traffics which are exemplified in the following.

The performance comparison between R18 and OE under transpose1 and transpose2 traffics is depicted in Figs. 18 and 19, respectively. The average packet latency under R18 is significantly lower than that of OE. The throughput variation is depicted in the supplemental file in this section available online.

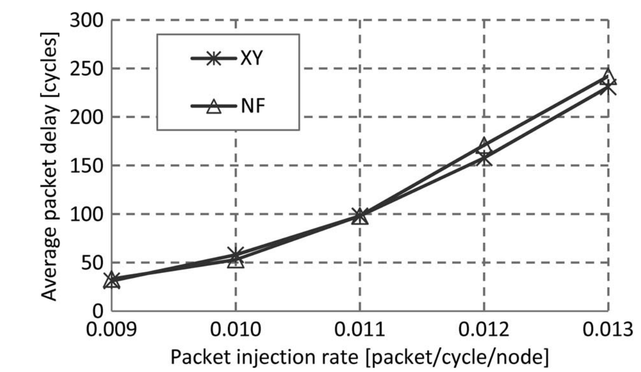

Fig. 16. Maximum packet injection rate under XY and negative-first routing for transpose1 traffic.

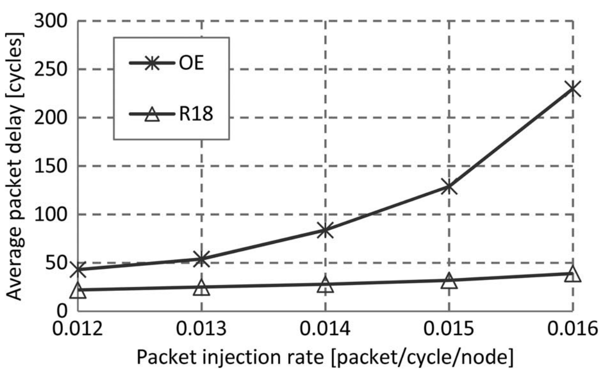

Fig. 18. Latency variation under transpose1 traffic scenario of routing R18.

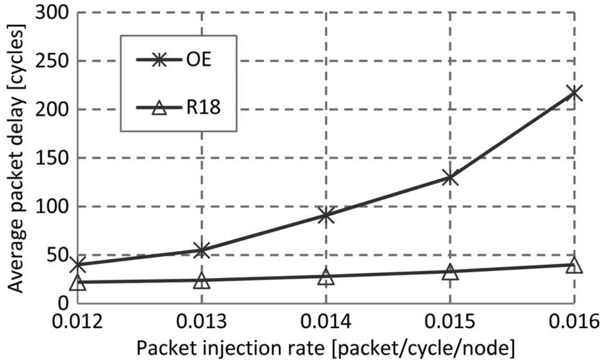

Fig. 19. Latency variation under transpose2 traffic scenario of routing R18.

The simulation results for random traffic are shown in Fig. 20. R18 also has better performance than OE.

A more realistic traffic of *hotspot* is also examined in this paper. We refer the reader to the supplemental file available online for the detail.

Although routing algorithm R18 only has smaller routing pressure than OE under transpose1 and transpose2 traffic, it has better performance than OE for random and *hotspot* traffics all the same.

Transpose1 and transpose2 traffics consider the situations where destination node situates at the southeast, northwest, southeast and northeast of the source node. Traffic mainly consists of the four types of traffics. If a routing works well under both transpose1 and transpose2 traffic, it can still work well under the other traffics. Consequently, after a new routing algorithm is obtained, its routing pressures under both transpose1 and transpose2 traffics can be used to measure its performance. It is no longer necessary to measure performance of routing for every traffic one by one, which would be infeasible.

6.2 Exploit the Best Performance Routing for Uniform Traffic

Uniform traffic is a special traffic in which the non-adaptive routing algorithm of XY has the much better performance compared with the adaptive routing [8], [9], [10], [13], [23]. However, our study shows that the performance difference between the XY routing and adaptive routing can be diminished with the help of routing pressure.

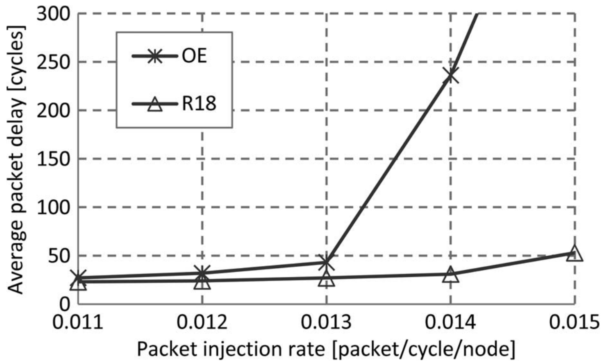

Fig. 20. Latency variation under uniform traffic scenario of routing R18.

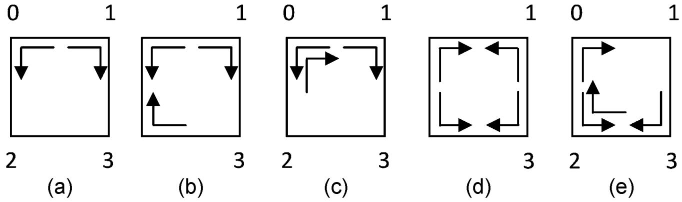

Fig. 21. Three categories of routing in 2×2 network.

We first show that XY and YX routings are the best routings for uniform traffic. Then we build a routing whose performance can be close to that of XY or YX routing.

In the 2×2 network, at most four turns can be prohibited to guarantee minimal path between any node pair, and at least two turns should be prohibited to avoid deadlock. The routing algorithms for 2×2 network can be classified into three categories. In the first category, only two turns are prohibited as shown in Fig. 21a. In the second category, three turns can be prohibited as shown in Figs. 21b and 21c. Figs. 21d and 21e depict the third category, four turns are prohibited.

Using the routing algorithm designing method [15], we can get all the routing algorithms for network of size 3×3. Four cases are considered as follows respectively.

1. Two turns are prohibited in each 2×2 subnetwork. There are 2529 deadlock-free routings for 3×3 network. The minimal routing pressure for uniform traffic is eight. One of such routing labeled 2-turns is selected for performance simulation.
2. Three turns are prohibited in each 2×2 subnetwork. Totally, 119582 deadlock-free routings are obtained for 3×3 network. The minimal routing pressure for uniform traffic is also eight. One of such routing labeled 3-turns is selected for performance simulation.
3. Four turns are prohibited in each 2×2 subnetwork. We get 24226 deadlock-free routings this time. The minimal routing pressure is six. Only two routings have this minimal routing pressure, they are XY and YX routings. The minimal routing pressure for the remaining 24224 routings is eight. One of such routing labeled 4-turns is selected for performance evaluation.

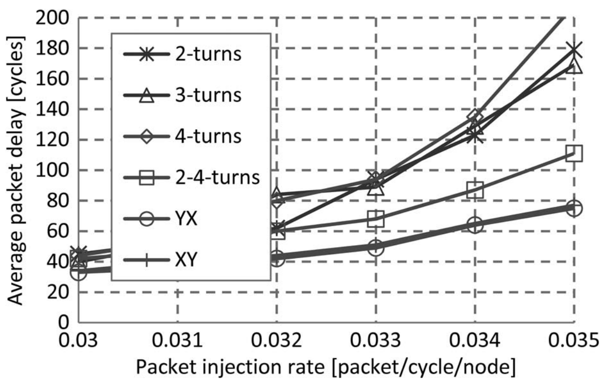

Fig. 22. Performance comparison for routings of 3×3 network under uniform traffic.

4. Two out of four turns are prohibited in each 2×2 subnetwork. There are 2259989 deadlock-free routings for 3×3 network, including XY and YX routings. Excludes XY and YX routings, the minimal routing pressure is seven. One of such routing labeled 2-4-turns is selected for performance evaluation.

Fig. 22 shows the performance of the six routings 2-turns, 3-turns, 4-turns, 2-4-turns, XY and YX. XY and YX routings have better performance than the other routings.

We claim that XY and YX routing has the best performance under uniform traffic since we have exhaustively searched all effective routings for 3×3 network.

We study to design adaptive routing algorithm whose performance is as close as possible to XY or YX routing under uniform traffic, offering satisfied performance under nonuniform traffic. The detail is shown in the supplemental file available online.

7 Conclusions

The degree of adaptiveness is often used to measure performance of routing algorithm. The intuition behind it is that more paths provide more chances for packet transmission to avoid congested nodes in the network. However, our study shows that high degree of adaptiveness may not be necessarily result in high performance. We have proposed a new metric of routing algorithm to more actually measure routing performance than the degree of adaptiveness. Performance of routing algorithm can be evaluated through routing pressure without simulation.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grants No. 60773199, U0735001, and 61073055 and the Scientific Research Foundation for Talent Introduction (NO. 2012RCYJ013).

References

[1] International Technology Roadmap for Semiconductors Interconnect, Semiconductor Industry Association.
[2] W.J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," in *Proc. ACM/IEEE Des. Autom. Conf.*, 2001, pp. 684-689.
[3] L. Benini and G.D. Micheli, "Networks on Chips: A New SoC Paradigm," *Computer*, vol. 35, no. 1, pp. 70-78, Jan. 2002.
[4] W.J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. San Mateo, CA, USA: Morgan Kaufman, 2004.
[5] C.J. Glass and L.M. Ni, "Maximally Fully Adaptive Routing in 2D Meshes," in *Proc. Intl Conf. Parallel Process.*, 1992, pp. 101-104.
[6] L. Schwiebert and D.N. Jayasimha, "Optimal Fully Adaptive Minimal Wormhole Routing for Meshes," *J. Parallel Distrib. Comput.*, vol. 27, no. 1, pp. 56-70, May 1995.
[7] C.J. Glass and L.M. Ni, "The Turn Model for Adaptive Routing," in *Proc. 19th Ann. Int'l Symp. Comput. Architect.*, 1992, pp. 278-287.
[8] C.J. Glass and L.M. Ni, "The Turn Model for Adaptive Routing," *J. Assoc. Comput. Machine.*, vol. 41, pp. 874-902, 1994.
[9] G.-M. Chiu, "The Odd-Even Turn Model for Adaptive Routing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 11, no. 7, pp. 729-738, July 2000.
[10] M. Palesi, R. Holsmark, S. Kumar, and V. Catania, "Application Specific Routing Algorithms for Networks on Chip," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 3, pp. 316-330, Mar. 2009.
[11] M.H. Tang and X.L. Lin, "Network-on-Chip Routing Algorithms by Breaking Cycles," in *Proc. ICA3PP*, 2010, pp. 163-173.
[12] A.M. Shafiee, M. Montazeri, and M. Nikdast, "An Innovational Intermittent Algorithm in NOC," in *Proc. World Acad. Sci., Eng. Technol.*, 2008, pp. 145-147.
[13] B.Z. Fu, Y.H. Han, J. Ma, H.W. Li, and X.W. Li, "An Abacus Turn Model for Time/Space-Efficient Reconfigurable Routing," in *Proc. ISCA*, June 2011, pp. 259-270.
[14] A. Mejia, J. Flich, J. Duato, S. Reinemo, and T. Skeie, "Segment-Based Routing: An Efficient Fault-Tolerant Routing Algorithm for Meshes and Tori," in *Proc. 20th IPDPS*, Rhodos, Grece, Apr. 2006, p. 105.
[15] M.H. Tang and C.H. Wu, "A New Method of Designing NoC Routing Algorithm," in *Proc. 2nd Int'l Conf. Consumer Electron., Commun. Netw.*, 2012, pp. 3044-3047.
[16] P. Bogdan and R. Marculescu, "Non-Stationary Traffic Analysis and Its Implications on Multicore Platform Design," *Trans. Comp.-Aided Des. Integr. Circuits Sys.*, vol. 30, no. 4, pp. 508-519, Apr. 2011.
[17] M. Bakhouya, A. Chariete, J. Gaber, and M. Wack, "A Buffer-Space Allocation Approach for Application-Specific Network-on-Chip," in *Proc. 9th IEEE/ACS AICCSA*, 2011, pp. 263-267.
[18] A. Scherrer, A. Fraboulet, and T. Risset, "Long-Range Dependence and On-Chip Processor Traffic," *Microprocess. Microsyst.*, vol. 33, no. 1, pp. 72-80, Feb. 2009.
[19] P. Bogdan and R. Marculescu, "Statistical Physics Approaches for Network-on-Chip Traffic Characterization," in *Proc. 7th IEEE/ACM Int. Conf. Hardw./Softw. Codesign Syst. Synthesis*, 2009, pp. 461-470.
[20] G. Varatkar and R. Marculescu, "On-Chip Traffic Modeling and Synthesis for MPEG-2 Video Applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 1, pp. 108-119, Jan. 2004.
[21] Z. Qian, P. Bogdan, G. Wei, C. Tsui, and R. Marculescu, "A Traffic-Aware Adaptive Routing Algorithm on a Highly Reconfigurable Network-on-Chip Architecture," in *Proc. 8th IEEE/ACM/IFIP Int'l Conf. Hardware/Software CODES+ISSS*, 2012, pp. 161-170.
[22] P. Gratz, B. Grot, and S.W. Keckler, "Regional Congestion Awareness for Load Balance in Networks-on-Chip," in *Proc. IEEE Int'l Symp. High Perform. Comput. Architect.*, 2008, pp. 203-214.
[23] D. Seo, A. Ali, W. Lim, N. Rafique, and M. Thottethodi, "Near-Optimal Worst-Case Throughput Routing for Two-Dimensional Mesh Networks," in *Proc. Int'l Symp. Comput. Architect.*, 2005, pp. 432-443.
[24] M. Daneshtalab, M. Ebrahimi, T.C. Xu, P. Liljeberg, and H. Tenhunen, "A Generic Adaptive Path-Based Routing Method for MPSoCs," *J. Syst. Architect.*, vol. 57, no. 1, pp. 109-120, Jan. 2011.
[25] J. Wu, "A Fault-Tolerant and Deadlock-Free Routing Protocol in 2D Meshes Based on Odd-Even Turn Model," *IEEE Trans. Comput.*, vol. 52, no. 9, pp. 1154-1169, Sept. 2003.
[26] M. Li, Q.-A. Zeng, and W.-B. Jone, "DyXY—A Proximity Congestion-Aware Deadlock-Free Dynamic Routing Method for Network on Chip," in *Proc. DAC*, 2006, pp. 849-852.
[27] S.M. Natalie, E.J., and Z.Y. Wang, "DBAR: An Efficient Routing Algorithm to Support Multiple Concurrent Applications in Networks-on-Chip," in *Proc. ISCA*, 2011, pp. 413-424.
[28] Sourceforge.netNoxim: Network-on-Chip Simulator, 2008. [Online]. Available: http://noxim.sourceforge.net
[29] P. Bogdan and R. Marculescu, "Quantum-Like Effects in Network-on-Chip Buffers Behavior," in *Proc. 44th ACM/IEEE DAC*, 2007, pp. 266-267.
[30] S. Hollis, C. Jackson, P. Bogdan, and R. Marculescu, "Exploiting Emergence in On-Chip Interconnects," *IEEE Trans. Comput.*, 2012.

Minghua Tang received the BS degree in applied math from Chongqing University, Chongqing, China, in 1999, the MS degree in applied math from Chongqing University, in 2005, and the PhD degree in computer software and theory from Sun Yat-Sen University, Guangzhou, China, 2010. He has been an Assistant Professor at the Department of Computer Science and Technology, Guangdong University of Finance, Guangzhou, China, since 2010. His research fields include the computer architecture and algorithm design for network-on-chip.

Xiaola Lin received the PhD degree in computer science in Michigan State University, East Lansing. He is a Professor in the Department of Computer Science, School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China. He has been doing research and teaching in USA, Hong Kong, and China. He has acted as an Investigator for the projects of millions Hong Kong dollars in Hong Kong and is a Principal Investigator for a number of research projects in China, such as NSFC projects etc. His research interests include parallel and distributed computing, and computer networking.

Maurizio Palesi received the MS and PhD degrees in computer engineering from the Universit di Catania, Catania, Italy, in 1999 and 2003, respectively. Since November 2010, he has been an Assistant Professor at Kore University, Enna, Italy. Dr. Palesi has been serving on the Editorial Board of VLSI Design journal as an Associate Editor since May 2007. He has served as a Guest Editor for the VLSI Design Journal—Special Issue on Networks-on-Chip in 2008, as a Guest Editor for the International Journal of High Performance Systems Architecture—Special Issue on Power-Efficient, High Performance General Purpose and Application-Specific Computing Architectures in 2009, and as Guest Editor for Elsevier MICPRO Journal—Special Issue on Network-on-Chip Architectures and Design Methodologies in 2010. He serves as the Technical Program Committee Member for the following IEEE/ACM International Conferences: RTAS, CODES +ISSS, ESTIMedia, SOCC, VLSI, ISC, and SITIS. He was also the Co-organizer of International Workshops on Network-on-Chip Architectures (NoCArc) from 2008 to 2012. He is a member of the IEEE.