International Journal of Parallel Programming (2019) 47:433–450
https://doi.org/10.1007/s10766-018-0614-6

HARE: History-Aware Adaptive Routing Algorithm for Endpoint Congestion in Networks-on-Chip

Kang Jin[1] · Cunlu Li[1] · Dezun Dong[1] · Binzhang Fu[2]

Received: 19 September 2018 / Accepted: 16 November 2018 / Published online: 12 December 2018

Abstract

Endpoint congestion is one of the most challenging issues when designing low latency and high bandwidth on-chip interconnection networks. Tree saturation and head-of-line blocking caused by the endpoint congestion seriously decrease system throughput and increases network latency, leading to overall performance degradation. Adaptive routing algorithms utilize dynamic network states to route packets around congestion areas and potentially mitigate network congestions, but still cannot deal with endpoint congestions. Existing adaptive routing algorithms mainly take the current route information into account, and rarely use the route information of past packets. In this paper, we explore the route information of past packets, and led to the following novel observations that the virtual channel (VC) allocations of prior packets can be collected as useful information, and the tree saturation can be isolated through better VC selection strategy based on the past route information. Based on this observation, a novel history-aware adaptive routing algorithm for endpoint congestion, HARE, is proposed to improve network performance. We implement HARE based on the state-of-the-art routing algorithm, Footprint, and conduct extensive simulation experiments to compare it with our algorithm. The evaluation results show that our design alleviate the impact of tree saturation consistently and achieve high throughput on both synthetic and trace-driven workloads.

✉ Dezun Dong
dong@nudt.edu.cn

Kang Jin
jinkang17@nudt.edu.cn

Cunlu Li
cunluli@nudt.edu.cn

Binzhang Fu
fubinzhang@huawei.com

[1] National University of Defense Technology, Changsha, China

[2] HuaWei, Beijing, China

Keywords History-aware · Adaptive routing · Endpoint congestion · Networks-on-chip

1 Introduction

Due to the scalable and modular feature of Networks-on-Chip (NoCs), it has been treated as a promising substitute to traditional bus-based architecture for inter-core communication [13]. The design of an efficient NoCs has become an evolving field of research. Many topologies [3,20,21] have been proposed to improve the performance of NoCs. In this study, we prefer two-dimensional (2D) mesh topology because of its regularity and scalability, and focus on routing optimization to improve the performance of NoCs.

Given the topology of the network, routing algorithm is a main factor that affects network performance (throughput and latency). Generally, routing algorithms can be divided into two classes: oblivious and adaptive routing algorithms, based on whether using the network state information or not [8]. In addition, routing algorithms can also be categorized into minimal and non-minimal based on whether choosing the shortest paths or not. In this study, we focus on adaptive minimal routing algorithms as they could distribute traffic across network in case of network congestion while maintaining low implementation complexity.

Network congestion is a major performance inhibitor in NoCs. Generally, network congestion can be classified as fabric congestion and endpoint congestion [18]. Fabric congestion is created when the offered load on a channel is greater than its bandwidth, while endpoint is created when some network nodes are over-subscribed, forming hot-spots in the network [2]. Fabric congestion can be efficiently relieved by balancing load across network channels through adaptive routing algorithm. However, the adaptivity provided by adaptive routing algorithm can aggravate the impact of endpoint, by spreading congestion across other ports and channels [7]. Moreover, a single hot-spot endpoint can spread the congestion through the network and create an effect called tree saturation [27], which can affect the wAe network performance.

Many congestion control mechanisms have been proposed to resolve network congestion. SRP [17], CRP [24] and SMSRP [18] share the similar strategy that reserving network resources for each flow transmission to avoid congestion, but their implementation is too complex. Other class of congestion control is mainly about adaptive routing algorithm. The CBCM [19] is one of the first work to address the interaction between adaptive routing and endpoint congestion. Based on the observation of CBCM, Footprint [12] is proposed to relieve the impact of tree saturation caused by endpoint congestion. Footprint operates on the principle that the next packet follows the path (footprint) of the current routed packet to the same destination when endpoint congestion occurs. However, most of these works, including Footprint, do not utilize the path information of past packets, resulting in bad isolation for tree saturation or inaccurate recognition of endpoint congestion.

In this work, we propose HARE, a History-aware Adaptive Routing for Endpoint congestion in Networks-on-Chip. HARE targets to relieve the impact of tree saturation caused by endpoint congestion. HARE extends routing algorithm in time-series; it

not only takes the current packet information into account but also considers the information of multiple packets that have been routed in past cycles, while performing routing decision. We implement HARE based on Footprint and it uses the two-level routing adaptiveness similar as Footprint. HARE restricts the saturation tree to take up the least VCs by giving higher priority to the deepest footprint VCs. The depth of a footprint VC is defined as the total length of packets sent to the destination node in the most recent period in a VC. During the routing process of HARE, the deepest footprint VC will be prioritized when congestion occurs, by which the tree saturation will be isolated more efficiently and consistently.

To summarize, the main contributions of this paper are as follows:

1. We propose to utilize history information of past packets to alleviate the impact of endpoint congestion, which extends the routing algorithm in time dimension.
2. We propose a new VC selection strategy that prioritizes the deepest VC to isolate the congestion tree efficiently.
3. Based the strategy above, we propose a novel adaptive routing algorithm, HARE, to address endpoint congestion.

The rest of the paper is organized as follows. In Sect. 2 we describe tree saturation, HoL blocking and the previously studied Footprint routing algorithm in detail. The theory and implementation of HARE are described in detail in Sect. 3. Network simulator configuration used for evaluation is described in Sect. 4. In Sect. 5, we present the performance comparison of the routing algorithms. More in-depth studies relating to the implementation and scalability of HARE are presented in Sect. 6. Other works relating to network congestion management are discussed in Sect. 7. We conclude the study in Sect. 8.

2 Motivation

2.1 Congestion Analysis

Endpoint congestion usually occurs when multiple source nodes send packets to the same destination node. Traffic that can not be handled opportunely will occupy the router buffer, causing endpoint congestion. More seriously, the congestion in the oversubscribed destination node will propagate back to upstream routers, creating tree saturation. Figure 1 shows simple tree saturation scenario with different routing algorithm. They use the same traffic pattern: $\{f_1, f_2\} = \{n_0 \rightarrow n_{14}, n_{13} \rightarrow n_{14}\}$. The topology of the network is a 4×4 2D mesh with 3 VCs per physical channel.

With deterministic routing, node n_0 and node n_{13} are contending for the hot-spot destination n_{14}. Initially, the adjacent nodes of n_{14} are congested because buffers are occupied by blocked packets. Then, such congestion effect continuously spreads upstream until it reaches the source nodes n_0 and n_{13}. The resulting congestion tree is shown in Fig. 1a with red arrow. In the second example shown in Fig. 1b, minimal adaptive routing is adopted. However, the network condition becomes even worse. This is because adaptive routing spreads congestion to other alternative paths, which resulting in a much larger congestion tree.

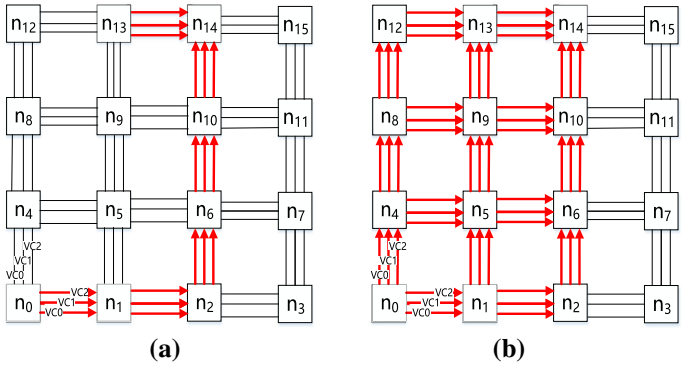


Fig. 1 Examples of tree saturation with dimension order routing and minimal adaptive routing. **a** DOR. **b** Adaptive (Color figure online)

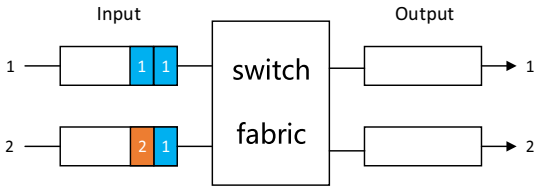


Fig. 2 An example of HoL blocking

Another unexpected impact of endpoint congestion is head-of-line (HoL) blocking. Endpoint congestion associated with adaptive routing can result in more serious HoL blocking. The packets contributing to the saturation tree will be congested in the head of VC, which makes other uncongested packets cannot be routed to the down-stream router. Figure 2 shows an example to demonstrate the impact of HoL blocking in an on-chip router. In this example, the first and second input ports of the router are contending to send packets to a congested output port. The packets at the head of Input 1 and Input 2 are all congested due to endpoint congestion because they contain the same destination. In this case, the uniform packets that are not at the head of VCs will also experience congestion.

2.2 Existing Routing Algorithm Solving Endpoint Congestion

Prior works (e.g., XORDET [5], DBAR [23], CBCM [19], etc.) propose to utilize a dedicated resource to isolate the impact of tree saturation. The recently proposed routing algorithm, Footprint, shares similar idea, but it takes virtual channels (VCs) adaptiveness into consideration to avoid the spreading of congestion through multiple VCs. Footprint has three main steps, including determining available outputs, selecting output port, selecting VC. In the first step, it generates a set of alternative output ports and records the number of idle VCs, footprint VCs respectively. In the second step, Footprint selects the port with more idle VCs. If equal, the number of footprint VCs is compared then. If still equal, port will be selected randomly. In the third step, different class of VC are assigned different priority when congestion occurs. Congestion is estimated by comparing the number of idle VCs with a predefined threshold. If the

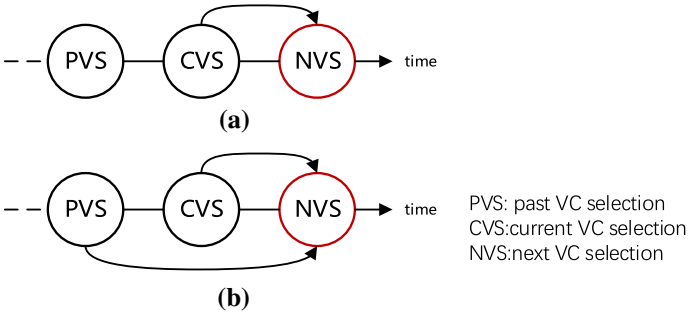


Fig. 3 An overview of different routing design for footprint which uses current VC info and HARE which uses all previous info about VC allocation. **a** Footprint. **b** HARE

network is congested, idle VC are requested with highest priority. Then footprint VCs are requested with relative low priority. Lastly, others are requested with lowest priority. Else, all adaptive VCs are requested.

Footprint is able to relieve the impact of tree saturation to some extent through limiting the available VCs during routing computing. However, Footprint only considers the path information of current packet for a VC which is not sufficient to efficiently isolate congestion tree caused by endpoint congestion. It should be noted that the HoL blocking shown in Fig. 2 still can exist when adopting Footprint routing. This is because different footprint VCs have the same priority in Footprint routing, and congestion can spread between footprint VCs. Hence, Footprint routing algorithm does not actually achieve expected performance-minimizing the impact of tree saturation. Since all footprint VCs are requested with equal probability, tree saturation will still significantly degrade the network performance. Such impact will be even worse if the majority of VCs are footprint VCs. To better isolate the saturation tree, it is needed to distinguish between different footprint VCs.

3 History-Aware Adaptive Routing Design

Based on the analysis of shortcomings of Footprint, we develop HARE to isolate the impact of tree saturation and HoL blocking more effectively.

3.1 History-Aware Adaptive Routing Theory

HARE exploits a novel history-aware adaptive routing theory that uses history routing information to optimize the current routing operation. Different from prior routing theories, it extends packets routing in time-series. Figure 3 illustrates the contradistinction of fundamental ideas of Footprint and our work. PVS means VC allocations of all packets in the VC. CVS means VC allocation of first packet in the VC. NVS means VC allocation of the coming packet. According to the idea of Footprint that packets follow the path of previous packet to the same destination, it only utilizes CVS info for the NVS operation as shown in Fig. 3a. With HARE, however, VC path info for all previous packets in a router is used to optimize the VC selecting for the next packet as depicted in Fig. 3b.

3.2 History-Aware Adaptive Routing Implementation

To actually constrain the spreading of congestion, we need to limit the number of available footprint VCs. How to differentiate the VCs and make the optimal selection is one of the keys of the paper, since the selection strategy in adaptive routing algorithm has a significant impact on network performance. An efficient selection strategy could realize high adaptivity when fabric congestion occurs and dynamic isolation when endpoint congestion occurs. In this work, we present a new VC selection strategy that implements history-aware adaptive routing theory. The full routing algorithm is summarized in Algorithm 1.

Algorithm 1 HARE Routing Algorithm

1: Determine legal output ports and compute the number of idle VCs, footprint VCs and its depth;
2: Determine output port. Select the port with more idle VCs. If equal, compare the number of footprint VCs then. If still equal, randomly select a port;
3: Determine VC requests. If no congestion, randomly select a VC. Else prefer idle VC, then deepest footprint VC, then other footprint VC, and finally other busy VC.

Our implementation creatively exploits the VC depth to minimize the congestion tree and maximize buffer utilization. To compute the VC depth, we need additional bits to store the destinations of all packets and the number of packets to the same destination for each VC. This only introduces little buffer overhead to the control plane of a router (see Sect. 5.3), and their is no additional buffer overhead in datapath. Whats more, since we use similar port selection strategy as Footprint, the only change is adding a priority for deepest footprint VC between idle VCs and common footprint VCs in the process of determining VC requests. In addition, HARE is deadlock free, since it is based on Footprint and the we just limit the number of available footprint VCs as mentioned above.

3.3 History-Aware Adaptive Routing Example

The detailed VC selection strategy is shown in Fig. 4. R0 and R1 are two independent routers in congestion tree caused by endpoint congestion. Packets with different color means they belong to different flows. In the first case as shown in Fig. 4a, there are three footprint VCs (VC0, VC1, VC3) and one busy VC (VC2) for packets to R0 with red color. With HARE, we calculate the depth for each VC. Then we compare the depth of footprint VC to select the deepest footprint VC. The reason why we select the deepest footprint VC rather other footprint VC is that packets to the same destination are allocated to the same VC as much as possible so as to reduce HOL blocking. In contrast to Footprint, which selects among all footprint VCs with equal possibility, HARE always selects the deepest VC. This improvement of selection strategy makes the isolation of tree saturation more thorough.

In another case as shown in Fig. 4b, since there is one idle VC (VC3) in R1, HARE will choose the idle VC, like Footprint, to keep high adaptivity. However, the key difference with Footprint is how to route afterwards. Assuming that in the next clock

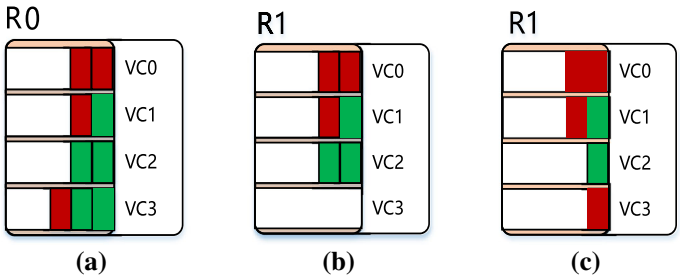


Fig. 4 VC selection strategy of HARE in different situations

cycle the VC occupancy condition of R1 is illustrated as Fig. 4c. For the subsequent packet to R1, HARE will still select the deepest footprint VC (VC0) no matter how many footprint VCs, while Footprint equally requests all three footprint VCs with one footprint VC (VC3) newly generated in the previous step. Hence, Footprint not only doesnt constrain the impact of tree saturation, but also expands the size of congestion tree in this situation.

3.4 History-Aware Adaptive Routing Effect

Figure 5 shows the effect comparison of Footprint and our work, using the same traffic pattern as Fig. 1: $\{f_1, f_2\} = \{n_0 \rightarrow n_{14}, n_{13} \rightarrow n_{14}\}$. We assume node n_{14} is oversubscribed and becomes a hot-spot. With ordinary adaptive routing that does not consider endpoint congestion, the resulting congestion tree is depicted in Fig. 1b. The size of congestion tree becomes a little bit small with Footprint, which is because Footprint restricts the congestion to footprint VCs. Figure 5a shows a possible resulting congestion tree, and the number of congested VCs between nodes could be different. This is because Footprint does not limit the number of footprint VCs. In contrast, HARE limits the congested packets to one footprint VC, minimizing the size of congestion tree. Figure 5b is a possible congestion tree with HARE. Noting that the deepest footprint VC could be VC0, VC1 or VC2, this ensure that load is balanced across all VCs. In a word, Footprint could isolate the congestion tree to some degree, while HARE achieves the ideal goal of Footprint that minimize the impact of congestion tree.

4 Evaluation Methodology

We evaluate the proposed HARE routing algorithm using a cycle accurate simulator Booksim [16]. We compare the performance of HARE with an adaptive routing algorithm,Footprint, and an oblivious routing algorithm, dimension order routing (DOR).

The baseline network topology, unless otherwise specifically stated, is a 8×8 2D mesh. The baseline VC number per physical channel is presumed to be 10. Meanwhile, 4, 8, 16 VCs are also appraised to study the impact of different VC number.

The simulated router uses credit-based virtual channel flow control with input-queued, wormhole switching microarchitecture. Network data packets comprise single flit as baseline but different packet sizes are also evaluated to examine the impact. The

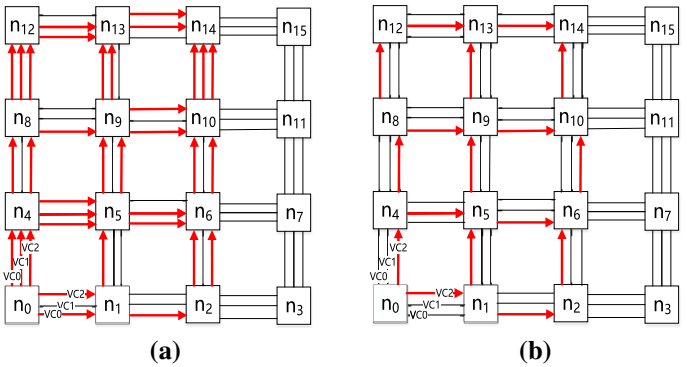


Fig. 5 Impact of endpoint congestion with footprint and HARE. **a** Footpirnt. **b** HARE

Table 1 Network simulation parameters

Parameters	Values
Traffic pattern	Uniform, shuffle, bitrev, transpose, hotspot, trace-driven workloads
Flow control mechanism	Virtual channel, credit-based
Packet size	**single-flit packets**, 1-6-flits uniform packets
Allocator	Priority-based
Speedup	Internal_speedup=2
Routing algorithm	HARE, footprint, DOR
VC	4, 8, **10**, 16 VCs per physical channel

The default values are marked in bold

input buffer size per VC is 4 flits. The router has 2× internal speedup to guarantee nearly 100% router throughput for any traffic. VC and switch allocation are performed using priority-based allocator.

Four representative types of synthetic traffic patterns, uniform random (UR), shuffle, bitrev and transpose, are used in the experiments. In addition, hotspot traffic is used to generate endpoint congestion to evaluate the ability of routing algorithm in restricting congestion tree. Besides synthetic workload, we use traces from PARSEC [4] workloads to evaluate the performance. The detailed configurations are listed in Table 1.

5 Evaluation

5.1 Synthetic Workload Result

5.1.1 Fixed Packet Size

Figure 6 shows the average latency of different routing algorithms with UR, shuffle, bitrev and transpose traffic pattern respectively. For UR traffic pattern (Fig. 6a),

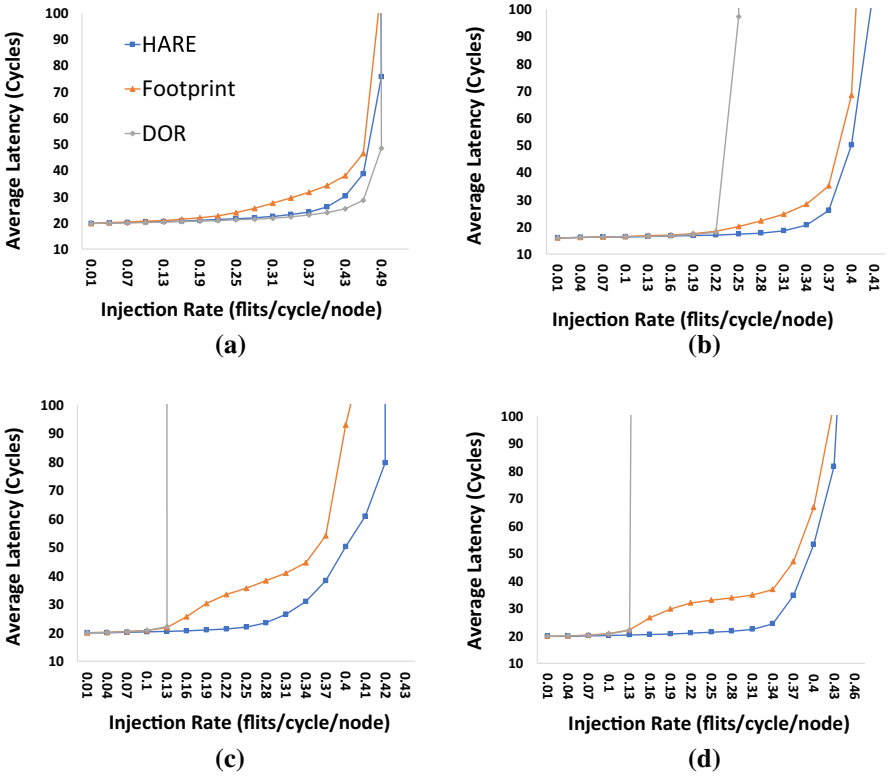


Fig. 6 Latency-throughput comparison with single-flit packet size. **a** Uniform. **b** Shuffle. **c** Bitrev. **d** Transpose

DOR provides the best performance since the load is already balanced. Adaptive routing introduces additional latency because of path selection function, non-minimal routing and increasing HoL blocking discussed in Sect. 2.1. Since HARE and Footprint are minimal adaptive routing, the reduction in saturation throughput is marginal. For non-uniform traffic patterns such as shuffle, bitrev and transpose traffic patterns, adaptive routing algorithms (HARE and Footprint) achieve higher saturation throughput compared to deterministic routing (DOR). This is because the load is serious imbalance and adaptive routing would distribute traffic across the entire network.

At low load, the latency of HARE is nearly identical to that of Footprint, which is because all adaptive VCs are requested under benign traffic. As network load increases, the occurrence of endpoint congestion among the network results in tree saturation. Footprint can somehow relieve the impact of tree saturation by prioritizing footprint VCs. However, HARE outperforms Footprint under all traffic pattern as shown in Fig. 6 since our method prefers to select the deepest footprint VC instead of all footprint VCs when endpoint congestion occurs so as to restrict the congestion tree and reduce HoL blocking further.

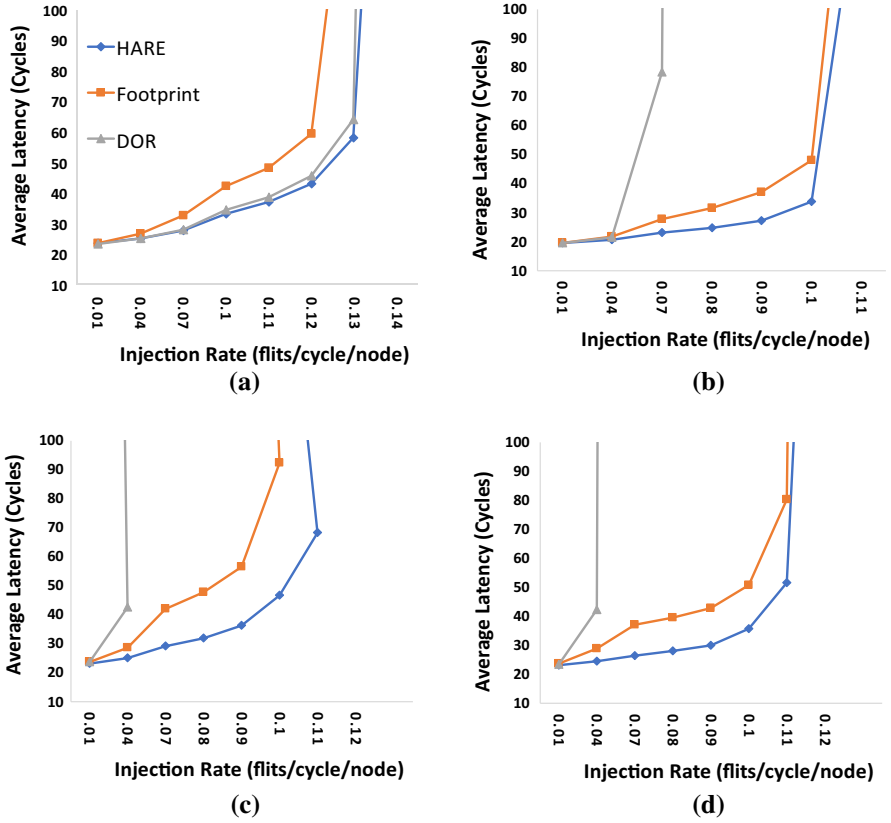


Fig. 7 Latency-throughput comparison with variable packet size. **a** Uniform. **b** Shuffle. **c** Bitrev. **d** Transpose

5.1.2 Varied Packet Size

In addition to fixed packet size, we also evaluate the proposed routing algorithm with different packet sizes. In this experiment, packets with size ranging from 1 to 6 flits are randomly generated in source node. For uniform traffic, DOR still provides slightly higher throughput than Footprint, while HARE achieves almost the same throughput as DOR. This is because adaptive routing would degrade network performance further when it comes to endpoint congestion as discussed in Sect. 2.1 and the adaptiveness is limited by restricting available VCs with HARE. For non-uniform traffic (shuffle, bitrev and transpose), adaptive routing algorithms (HARE and Footprint) outperform obvious routing (DOR), since load is extreme imbalance and the locality of DOR makes it worse. For all four traffic patterns, HARE provides higher throughput than Footprint as shown in Fig. 7. This demonstrates that HARE is able to isolate the impact of endpoint congestion more effectively.

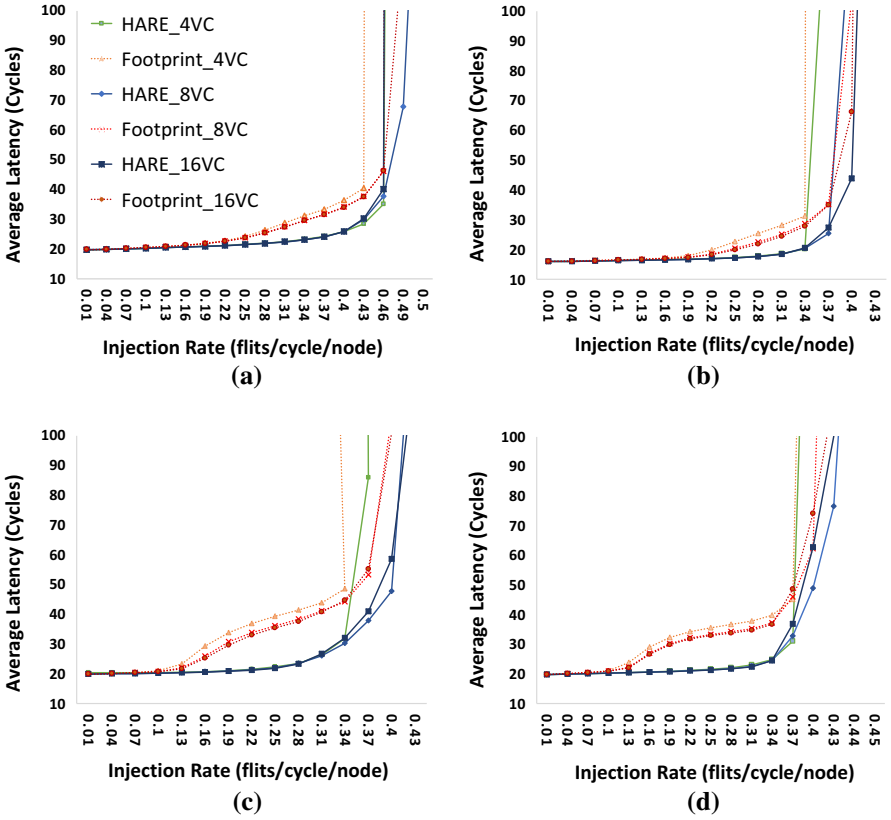


Fig. 8 Latency-throughput comparison with different number of VCs. **a** Uniform. **b** Shuffle. **c** Bitrev. **d** Transpose

5.1.3 Impact of Number of VCs

In this experiment, we evaluate the impact of number of VCs on Footprint and HARE. Since Duatos theory [9] is adapted to avoid deadlock, the number of required VCs is no less than 2. We perform experiment with 4 VCs, 8 VCs and 16 VCs respectively and the results are shown in Fig. 8.

For Footprint, the throughput increases as the number of VCs increases, as it is designed to take up as much resources as possible. In contrast, the throughput of HARE is nearly unchanged for different number of VCs, which is because increasing the number of VCs engages slight throughput improvement if there are already too many VCs. In addition, increasing the number of VCs means increasing the router delay due to complex VC allocation. An interesting insight offered by the graphs in Fig. 8 is that with the same number of VCs, HARE always offers higher saturation throughput and lower latency. Similarly, this is the result of the usage of history VC Allocation information and better VC selection strategy.

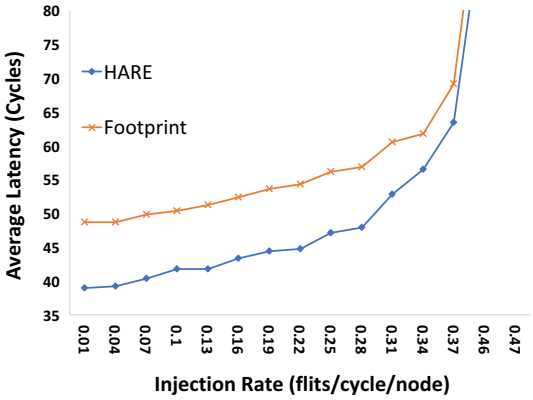


Fig. 9 Latency-throughput comparison with hotspot traffic

5.1.4 Hotspot Traffic

We evaluate the performance of HARE using a 16:12 hotspot traffic pattern. Under this traffic pattern, we select 16 nodes in the network to send traffic to 12 destination nodes, while other nodes run background uniform traffic with constant injection rate of 0.35. Figure 9 shows average latency of background traffic as the hotspot traffic injection rate increases. As illustrated in the figure, endpoint congestion created by hotspot traffic will damage the the background traffic. The performance of HARE is compared with Footprint and the results show that HARE achieves lower latency and higher saturation throughput. This is because we use VC allocation information of past packets and require packets to follow the deepest footprint VC, hence isolating the congestion tree and reducing HoL blocking.

5.2 Application Traces

In this section, we compare the performance of HARE and Footprint using traces from PARSEC 2.0 workloads. The results are shown in Fig. 10 where the improvement percentage is depicted. Similarly, HARE achieves better performance for all cases except for ferret and swaptions. Among all the network traces, HARE outperforms Footprint by up to 18% and the average improvement is 1.8%. For some traces with low traffic, the benefit is small. While for traces that generate heavy network traffic (i.e., multiregion), the improvement is significantly higher. This is because endpoint congestion is more likely to occur in heavy traffic and there is more room for HARE to improve. In this experiment, we can see that HARE is more effective in isolating the hotspot congestion.

5.3 Cost

There is little overhead for HARE compared with Footprint as only the statistical data of past packets is added for each VC. Beyond that, the improvement of VC selection

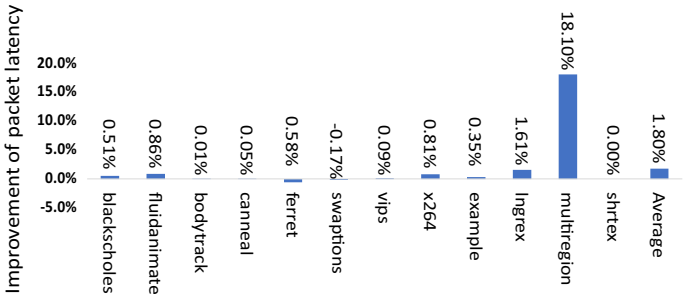


Fig. 10 Improvement of packet latency using traces from PARSEC workloads

strategy does not introduce any expend. To prioritize the deepest Footprint VC over usual Footprint VC, we need add a $\log_2 n$ bits of register for each VC to record the corresponding depth, Where n is number of packets to the same destination packets as the current routing packet for a VC, namely depth of VC. Usually, n remains in single digits and we assume the value of n is 8 for simplicity. For a 8×8 2D mesh topology with 8VCs per physical channel, the corresponding overhead is 24 bits per port which is negligible considering general flit size (e.g., 128 or 256 bits).

6 Discussion

6.1 VC Selection Strategy Variants

The VC selection strategy we use to implement the history-aware adaptive routing is to select the deepest footprint VC when endpoint congestion occurs and experiment results show that it works well. However, there are many other implementations of the history-aware routing theory, such as prioritizing the first two, three, or even n deepest footprint VC, where n is an integer no more than the number of VCs per physical channel. We have implemented a variant of HARE that prefers to select the first two deepest footprint VC and name it HARE2. Then, we evaluate the performance of HARE2 with uniform and non-uniform traffic pattern. The network configuration is the same as that of evaluations in Sect. 5.1.2. With HARE used as baseline, the performance comparison is shown in Fig. 11. The throughput of HARE2 is nearly identical to that of HARE with uniform traffic pattern, while HARE2 achieves higher saturation throughput with shuffle traffic pattern. This is because the load is extremely unbalanced and allowing two VCs for a flow distributes the load in some extend. Apparently, there is a tradeoff between congestion isolation and load balancing. There may be an optimal digit that we should limit the number of footprint VCs to, and we leave it as a future work. In addition, we evaluate the performance of HARE2 with hotspot traffic pattern, using the same configuration as that of evaluation in Sect. 5.1.4. The result is shown in Fig. 12. HARE2 achieves the same throughput as HARE, which is because the impact of endpoint congestion is effectively minimized through limiting congestion to a few VCs.

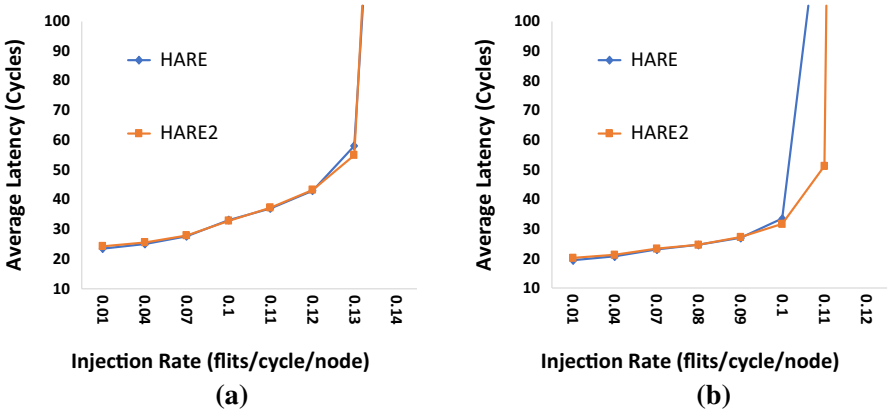
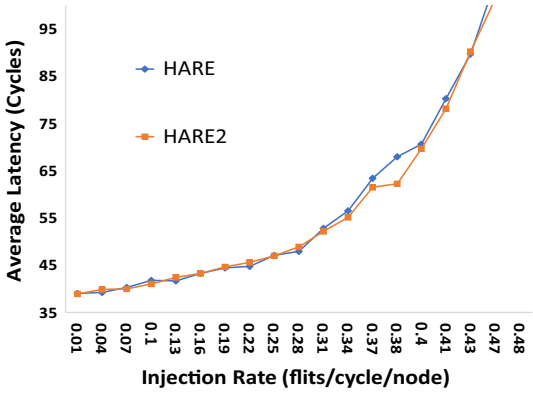


Fig. 11 Latency-throughput comparison with uniform and non-uniform traffic pattern. **a** Uniform. **b** Shuffle

Fig. 12 Latency-throughput comparison with hotspot traffic



6.2 Scalability

HARE is more than an adaptive routing algorithm, but an effective adaptive routing theory for endpoint congestion. It can be combined with most routing algorithms previously proposed, since the only change is better VC selection strategy. In addition, the theory is topology-agnostic. It can be applied to any topologies theoretically. However, the implementation for specific topology might be a little different and we leave it as a future work.

6.3 Fabric Congestion

HARE is designed for endpoint congestion, which does not mean it can not deal with fabric congestion. Adaptive routing algorithms are born to tackle fabric congestion, but make it worse when the network is suffered from endpoint congestion. Hence, we propose a novel history-aware adaptive routing algorithm, HARE, to address fabric congestion and endpoint congestion simultaneously. Specifically, HARE prioritizes

the port with more idle VCs in the port selection stage and prioritizes idle VC in the VC selection stage, so as to route around fabric congestion.

7 Related Work

The development of HARE is in part motivated by the Footprint routing algorithm proposed in [12]. Together they share many similar operating principles such as two level adaptivity and VC limitation. However, as stated in the original Footprint study and demonstrated in our experiments, Footprint doesnt effectively isolate the congestion tree due to insufficient information. We have shown that by considering the history information in the same work, we can create a more efficient solution that works well for endpoint congestion in Networks-on-Chip.

There are many other endpoint congestion control mechanisms based on the idea of congestion isolation besides Footprint. VOQnet [6] requires as many VCs as destinations in the network for each input queue. Though it is effective to eliminate HoL blocking, it is not scalable to large network. However, VOQsw [1]uses as many VCs as output ports in a router to reduce cost. So, this technique is able to eliminate router-wide HoL blocking, but it doesnt completely remove HoL blocking. Destination-Based Buffer Management (DBBM) [25] argues that VCs are assigned to different destinations evenly. XORDET [5] shares similar principle as DBBM but improved for direct topologies. However, their buffer utilization is very low, as VCs are statically assigned to different end-points in the network. In addition, RECN [10] uses dynamically created set aside queues (SAQs) to eliminate HOL blocking in an efficient way. However, RECN mechanism is limited to source deterministic routing. Whats more, FBICM [11], which can be applied to networks that use distributed deterministic routing, has been proposed, achieving the same effect as RECN, but its implementation is too costly and complex.

The other main class of congestion management is based on reservation such as Speculative reservation protocol (SRP) [17], channel reservation protocol (CRP) [24], Small-Message SRP (SMSRP) and Last-Hop Reservation Protocol (LHRP) [18]. SRP proposed by Jiang, et al. is designed to avoid hot-spot congestion, but the traffic schedule based on network status information is not always available. CRP done by Michelogiannakis et al. is proposed to deal with both fabric and endpoint congestion. However, the reservation scheduling is too complex due to the consistency of multiple network resource.

An alternative technique for congestion control is based on congestion notification, such as explicit congestion notification ECN [28]. ECN sends congestion alarm to the source nodes contributing to its appearance to throttle its injection rate, if the occupancy of queues in the router exceeds a predefined threshold. However, many works have shown that ECN is slow to response and is potentially unstable [17,18,26]. Since our work is orthogonal to these reservation-based and notification-based congestion control mechanisms, the approach of HARE can be combined with these state-of-art solutions and the existing advantages can be retained together.

Furthermore, various adaptive routing algorithms have been proposed to tackle congestion. Both DyAD [15] and DyXY [22] use local network state information, which will lead to non-optimal routing decision. RCA [14] proposed by Gratz et al. is the first work utilizing global information to improve load balancing. However, this algorithm introduces redundant information that may degrade the congestion estimation. To overcome this problem, a technique proposed by Ramanujam et al. uses a dedicated network to sequentially transmit delay information for each network node [29]. However, this technique may be not efficient due to the high delay, for each router, of calculating the estimate for all other nodes in the network.

8 Conclusion

Network endpoint congestion is difficult to address as tree saturation and HoL blocking it creates degrade the whole network performance further. In this study we introduced HARE, a history-aware adaptive routing algorithm, to resolve endpoint congestion. HARE utilizes the past route information of packets to count the depth of the footprint VC. Specifically, HARE favors the deepest footprint VC when congestion occurs. In contrast, Footprint only uses current path info to mark the footprint VCs and select these VCs with equal probability. By prioritizing the deepest footprint VC instead of all footprint VCs, HARE is effective in resolving endpoint congestion. Experiments show that HARE achieves latency reduction and throughput improvement for synthetic and trace-driven workloads. This demonstrates that HARE is able to excel at isolating endpoint congestion and outperform exiting adaptive routing algorithm.

Acknowledgements The work was supported by Core-electronics, High-end-general-Chips, and Infrastructural-software Project of China (2018ZX01028101), National Key Research and Development Program of China under Grant No. 2016YFB0200401, Innovation Platform and Talents Program of Hunan Province under Grant No. 2017RS3047 and FANEDD under Grant No. 201450.

References

1. Anderson, T.E., Owicki, S.S., Saxe, J.B., Thacker, C.P.: High-speed switch scheduling for local-area networks. ACM Trans. Comput. Syst. (TOCS) **11**(4), 319–352 (1993)
2. Benson, T., Anand, A., Akella, A., Zhang, M.: Understanding data center traffic characteristics. In: Proceedings of the 1st ACM Workshop on Research on Enterprise Networking, pp. 65–72. ACM (2009)
3. Besta, M., Hoefler, T.: Slim fly: a cost effective low-diameter network topology. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 348–359. IEEE Press (2014)
4. Bienia, C., Kumar, S., Singh, J.P., Li, K.: The PARSEC benchmark suite: characterization and architectural implications. In: Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques, pp. 72–81. ACM (2008)
5. Cebrian, R.P., Requena, C.G., Requena, M.E.G., Rodrguez, P.L., Marn, J.D.: HoL-blocking avoidance routing algorithms in direct topologies. In: 2014 IEEE International Conference on High Performance

Computing and Communications, 2014 IEEE 6th International Symposium on Cyberspace Safety and Security, 2014 IEEE 11th International Conference on Embedded Software and System (HPCC, CSS, ICESS), pp. 11–18. IEEE (2014)
6. Dally, W., Carvey, P., Dennison, L.: Architecture of the Avici terabit switch/router. pp. 41–50 (1998)
7. Dally, W.J., Aoki, H.: Deadlock-free adaptive routing in multicomputer networks using virtual channels. IEEE Trans. Parallel Distrib. Syst. **4**(4), 466–475 (1993)
8. Dally, W.J., Towles, B.P.: Principles and Practices of Interconnection Networks. Elsevier, Amsteredam (2004)
9. Duato, J.: A new theory of deadlock-free adaptive routing in wormhole networks. IEEE Trans. Parallel Distrib. Syst. **4**(12), 1320–1331 (1993)
10. Duato, J., Johnson, I., Flich, J., Naven, F., Nachiondo, T.: A new scalable and cost-effective congestion management strategy for lossless multistage interconnection networks. In: Null, pp. 108–119. IEEE (2005)
11. Escudero-Sahuquillo, J., Garca, P., Quiles, F., Flich, J., Duato, J.: FBICM: Efficient congestion management for high-performance networks using distributed deterministic routing. In: International Conference on High-Performance Computing, pp. 503–517. Springer, Berlin (2008)
12. Fu, B., Kim, J.: Footprint: regulating routing adaptiveness in networks-on-chip. In: 2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA) pp. 691–702 (2017)
13. Gaur, M.S., Laxmi, V., Zwolinski, M., Kumar, M., Gupta, N.: Network-on-chip: current issues and challenges. In: 2015 19th International Symposium on VLSI Design and Test (VDAT), pp. 1–3. IEEE (2015)
14. Gratz, P., Grot, B., Keckler, S.W.: Regional congestion awareness for load balance in networks-on-chip. In: IEEE 14th International Symposium on High Performance Computer Architecture, 2008. HPCA 2008, pp. 203–214. IEEE (2008)
15. Hu, J., Marculescu, R.: DyAD: smart routing for networks-on-chip. In: Proceedings of the 41st Annual Design Automation Conference, pp. 260–263. ACM (2004)
16. Jiang, N., Balfour, J., Becker, D.U., Towles, B., Dally, W.J., Michelogiannakis, G., Kim, J.: A detailed and flexible cycle-accurate network-on-chip simulator. In: 2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp. 86–96. IEEE (2013)
17. Jiang, N., Becker, D.U., Michelogiannakis, G., Dally, W.J.: Network congestion avoidance through speculative reservation. In: 2012 IEEE 18th International Symposium on High Performance Computer Architecture (HPCA), pp. 1–12. IEEE (2012)
18. Jiang, N., Dennison, L., Dally, W.J.: Network endpoint congestion control for fine-grained communication. In: 2015 SC-International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–12. IEEE (2015)
19. Kim, G., Kim, C., Jeong, J., Parker, M., Kim, J.: Contention-based congestion management in large-scale networks. In: 2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), pp. 1–13. IEEE (2016)
20. Kim, J., Dally, W.J., Abts, D.: Flattened butterfly: a cost-efficient topology for high-radix networks. In: ACM SIGARCH Computer Architecture News, vol. 35, pp. 126–137. ACM (2007)
21. Kim, J., Dally, W.J., Scott, S., Abts, D.: Technology-driven, highly-scalable dragonfly topology. In: ISCA'08. 35th International Symposium on Computer Architecture, 2008, pp. 77–88. IEEE (2008)
22. Li, M., Zeng, Q.A., Jone, W.B.: DyXY: a proximity congestion-aware deadlock-free dynamic routing method for network on chip. In: Proceedings of the 43rd Annual Design Automation Conference, pp. 849–852. ACM (2006)
23. Ma, S., Enright Jerger, N., Wang, Z.: DBAR: an efficient routing algorithm to support multiple concurrent applications in networks-on-chip. ACM SIGARCH Comput. Archit. News **39**(3), 413–424 (2011)
24. Michelogiannakis, G., Jiang, N., Becker, D., Dally, W.J.: Channel reservation protocol for over-subscribed channels and destinations. In: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, p. 52. ACM (2013)
25. Nachiondo, T., Flich, J., Duato, J.: Buffer management strategies to reduce hol blocking. IEEE Trans. Parallel Distrib. Syst. **21**(6), 739–753 (2010)
26. Pfister, G., Gusat, M., Denzel, W., Craddock, D., Ni, N., Rooney, W., Engbersen, T., Luijten, R., Krishnamurthy, R., Duato, J.: Solving hot spot contention using infiniband architecture congestion control. Proc. HP-IPC **2005**, 6 (2005)

27. Pfister, G.F., Norton, V.A.: Hot spot contention and combining in multistage interconnection networks. IEEE Trans. Comput. **100**(10), 943–948 (1985)
28. Ramakrishnan, K., Floyd, S., Black, D.: The addition of explicit congestion notification (ECN) to IP. Tech. rep. (2001)
29. Ramanujam, R.S., Lin, B.: Destination-based adaptive routing on 2d mesh networks. In: Proceedings of the 6th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, p. 19. ACM (2010)