Fully Adaptive Thermal-aware Routing for Runtime Thermal Management of 3D Network-on-Chip

Xin. Jiang, Xiangyang. Lei, Lian. Zeng and Takahiro. Watanabe

***Abstract*—Thermal problem is an essential issue which must be taken into account in the 3D Network-on-Chip (NoC) design, because it has a great impact on not only the network performance, but also the reliability of the message transmission. In this work, we present a fully adaptive runtime thermal-aware routing algorithm, which combines the distance, traffic state, path diversity and the thermal impact in the path determination. By simultaneously considering all these factors, the routing algorithm can effectively balance the traffic load while keeping high adaptivity and routability, which also results in an even distribution of temperature across the network. Instead of collecting the topology information of the whole network, we utilize a 12 bits register to reserve the router state for one hop away, which saves the hardware cost largely and decreases the network latency. The simulation results show our proposed routing algorithm can improve the latency and energy consumption by comparing with other previously proposed thermal-aware routing schemes, and the improvement is more remarkable in large scale networks.**

***Index Terms*—Thermal-aware, routing algorithm, visual-servoing, fully adaptive, 3D NoC**

I. Introduction

With the development of high performance semiconductor industry, the multi-core systems have been widely applied, while the interconnection is a major bottleneck that slows down the performance improvement. Three-Dimensional Network-on-Chip (3D NoC) with the benefits of smaller layout footprint, shorter physical distance and hop count and more directions per router has replaced bus architecture and 2D NoC for the on-chip interconnection to achieve lower power and smaller form factor for on-chip data exchange [1].

The thermal issue is a significant challenge in 3D NoC, because it impacts not only the network performance but also the chip reliability. Higher temperature will lead to longer propagation delay, slower circuit switching and larger leakage power. Thermal variations can also result in timing uncertainty, and circuit reliability depends exponentially on operating temperature [2]. On the other hand, the thermal problem on 3D NoC is more serious than that of 2D due to the higher switching activity, longer heat conduction path, larger cross-sectional power density and varying cooling efficiency [3].

Runtime thermal management (RTM) [4] is a popular way to control the router temperature under the safe mode. The monitor senses the temperature of each router and reports it to the temperature-aware controller. When the temperature rises to the thermal threshold, the controller throttles some routers to reduce the traffic load of the overheated routers, and then lower the temperature in the network. When the temperature falls back below the threshold, the controller stops throttling. There are several kinds of throttling schemes, global throttling (GT) to throttle all the routers, distributed traffic throttling (DTT) [2] to only throttle the overheated ones and vertical throttling (VT) [4] to throttle the routers in the vertical direction.

The major problems in RTM includes how to collect the throttling information in the network and how to route the message in the throttled network. Previous works utilize a time-variant topology to synchronize the throttling data to each router across the network, which introduces additional physical channels for data transmission and the synchronization time is increased exponentially with the topology size. In addition, the trade-off between routability and routing time is a big challenge in designing the routing algorithms in the thermal-aware networks. In this work, instead of getting the information for every router node, we apply a data delivery scheme that only collects the throttling information in two hops. Based on the local thermal-aware network, we design a fully adaptive routing algorithm which selects the routes according to the distance, traffic state and path diversity by passing around the throttled nodes. The routing algorithm can select a relatively short path with low traffic load, and the high path diversity results in a high routability. The routing computation is very simple, and there is no routability check before routing, which lead to short routing time and then an improvement for the network latency.

The rest of the paper is organized as follows: Section 2 presents previous research related to the study, Section 3 details the proposed 3D fully adaptive runtime thermal-aware routing algorithms including inter-layer and intra-layer routing, Section 4 illustrates the simulation results and Section 5 concludes the paper.

Manuscript received December 7, 2015.

Xin. Jiang is with Graduate School of Information, Production & Systems, Waseda University (e-mail: jiangxin@ruri.waseda.jp).

Xiangyang. Lei is with Graduate School of Information, Production & Systems, Waseda University (e-mail: leixiangyang@ruri.waseda.jp).

Lian. Zeng is with Graduate School of Information, Production & Systems, Waseda University (e-mail: lian_zeng_ips@ruri.waseda.jp).

Takahiro. Watanabe is with Graduate School of Information, Production & Systems, Waseda University (e-mail: watt@waseda.jp).

II. Related Works

Recent researches have paid much attention to thermal-aware routing algorithms. Most of these routing algorithms are based on a traffic and thermal-aware RTM scheme [4], in which a downward routing algorithm and a vertical throttling strategy are applied to achieve runtime thermal safety. Then many works were explored to improve the downward routing and vertical throttling. In [5], a Traffic- and Thermal-Aware Routing (TTAR) algorithm was proposed to balance the network traffic load by selecting the paths away from the neighbors of the throttled nodes in the horizontal layer and avoiding selecting the paths downward to the bottom layer in the vertical layer, aiming to improve the throughput and latency. Ref. [6] presented a Transport Layer Assisted Routing (TLAR) algorithm in which the topology information is used to assist the determination of routing. TLAR was composed of three kinds of lateral routing algorithms in the horizontal layer and downward routing in the vertical layer, called Downward-Lateral Deterministic Routing (DLDR), Downward-Lateral Adaptive Routing (DLAR) and Downward-Lateral Adaptive-Deterministic Routing (DLADR), using deterministic, adaptive and combination of both in the horizontal layer respectively. To improve the routability of TLAR in the horizontal layer and alleviate the traffic load in the bottom layer, a Topology Aware Adaptive Routing (TAAR) algorithm was proposed in [3]. Different from TLAR, TAAR used a Lateral Cascaded Routing to reroute the message when it is unroutable by Lateral Deterministic Routing (LDR) and Lateral Adaptive Routing (LAR) in the horizontal layer. Moreover, it applied a queuing analysis theory to balance the traffic in the vertical layer, and finally selected uniform random downward layer selection for the vertical routing.

The above routing algorithms are based on the awareness of the varying topology in the throttling mode, and the topology is got through a synchronization process in which the throttling information is collected in z-direction, x-direction and y-direction in sequence. To implement this process, additional physical channels for each I/O port in each direction and a register for storage of the delivery data are needed, which lead to an increase of the hardware overhead. To store the throttling state of each node, the table size grows with the topology size, and the time consumed on looking up table increases the network latency. Before using TLAR and TAAR, a routability check is required, and the time complexity for the check is very high, which also results in an increase of the network latency. In TLAR, the routability and adaptivity of the lateral routing algorithms are limited, and the downward routing brings about traffic congestion in the bottom layer. In TAAR, there are two routing modes to implement different routing algorithm, which requires two different memories and two separate channels, leading to an increase of the hardware overhead.

To overcome these drawbacks, in this paper, we design a fully adaptive thermal-aware routing algorithm for 3D NoC, which can achieve high routability and adaptivity with low hardware overhead.

III. Design Algorithms

To effectively control the temperature in the network, we use RTM and VT throttling strategy to throttle the overheated routers in the emergency mode. The assumptions in our model are: First, the source node and the destination node are not throttled. Second, if a router is throttled, then all the routers above it are throttled; otherwise, all the routers below it are not throttled. Third, the routers in the bottom layer are never throttled.

A. Overview of the Algorithm

The flow chat of the proposed routing algorithm is shown in Fig. 1, where S and D are source and destination nodes respectively, D_i is the corresponding destination (the node that has the same X and Y coordinates with the destination node) in layer i, and C_i is the current node in layer i. The routing algorithm is composed of two parts: fully adaptive intra-layer routing in the horizontal layer and downward routing in the vertical layer. To avoid deadlock, the routing algorithm firstly routes the packet in the horizontal layer until the corresponding destination node, and then to the destination node in the vertical direction. The turns from up to the four horizontal directions are prohibited. If the packet can't be routed in the current layer, it is transferred downward to the next layer and routed to the corresponding destination again by the intra-layer routing algorithm. This process is repeatedly executed until it is down to bottom layer. Since the routers in the bottom layer are not throttled, there is at least one path to the corresponding destination. Finally, the packet can be routed from the corresponding destination in the bottom layer to the destination node.

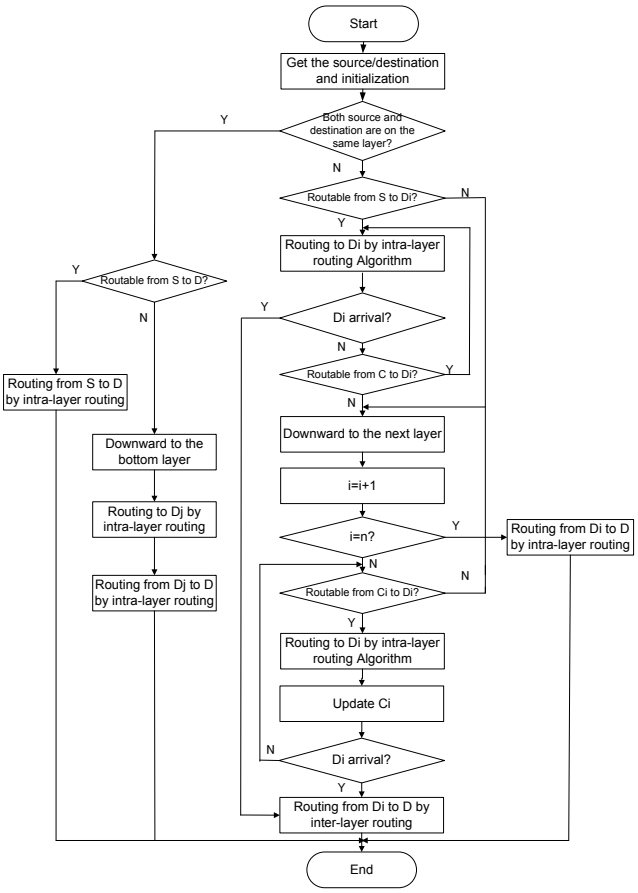


Fig. 1. Flow chat of the proposed routing algorithm

ISBN: 978-988-14047-6-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

We employ a throttling information collection mechanism that each router transfers the throttling state to the four neighbor nodes in the horizontal layer, through which the local router is informed of the node status in two hops. In according to the second assumption, there is no need to collect the throttling information in the vertical direction. In each router, there is a 12-bit register for storage the throttling information of the neighbor nodes. The throttling information is represented in a one bit signal, with "1" representing "throttle" and "0" as "normal". An example of the hardware architecture and the representation for the throttling status of the neighbor nodes is illustrated in Fig. 2. Fig. 2 (a) is the architecture of the data delivery for one pair of nodes, and Fig.2 (b) is the status for the neighbor nodes of C, where the shadow nodes are the throttled nodes. The throttling status is encoded as a sequence of binary number corresponding to each neighbor node respectively. The throttling information is used for path computation, and the throttled routers can be seen as faulty routers and are unavailable at present.

B. *Fully Adaptive Intra-layer Routing*

The intra-layer routing is a 2D fully adaptive routing algorithm, in which the route of the next hop is determined by distance, traffic state, path diversity, and throttling status of the neighbors together. It also includes livelock avoidance and deadlock recovery schemes. We denote C, D and C_i as the current node, destination node and the neighbor node in direction i (North, South, East, West in 2D) respectively, and use the following objective function to evaluate the routing condition of each neighbor node.

$$f_i = \alpha l_i + \beta s_i, i = 1,2,3,4 \quad (1)$$

where l_i denotes the distance (number of hops) from C_i to D, and s_i denotes the occupied buffer size in direction i, and α and β represent the weights for the two objectives. Each objective is normalized to the average value of the current node by using:

$$f_i' = \alpha l_i / l_0 + \beta s_i / s_0, i = 1,2,3,4 \quad (2)$$

where l_0 is the minimum number of hops from C_i to D, and s_0 is the total buffer size of C_i. The node C_i with the minimum function value will be selected as the next node on the routing path. If there are more than one candidate neighbor nodes, we will make decisions according to its path diversity. Diversity (d_i) is defined as the number of shortest path from the current node to the destination node [7]. In this case, the neighbor with high diversity will be selected.

The throttling information of the neighbor nodes is input to the current router, which has an impact on each parameter of the function. The routing function is calculated again by using the changed parameters, and then the candidate node on the path will be determined according to the function value. There are four cases:

-- (Case 1) If the direct neighbor nodes of the current nods are throttled, the corresponding l_i is set to be ∞, and s_i is set to be ∞. In this way, the throttled nodes can never be selected. An example of case 1 is shown in Fig. 3 (a), in which the east node of C is throttled, and it becomes an unavailable router.

-- (Case 2) If one node next to the neighbor nodes of C (for example node EE in Fig. 3 (b)) is throttled, l_i and d_i are changed according to the locations of the destination nodes. If there are still shortest paths to D, l_i remains unchanged. If there is no shortest path, l_i is recalculated by constructing a shortest path from the current node to D to bypath throttled nodes. If one of the nodes on the shortest path is throttled, d_i is reduced to 1/2 of the current value; if all nodes on the shortest path direction are throttled, d_i is reduced to 0. An example of case 2 is shown in Fig. 3 (b), where the east node of E is throttled. In this example, if the destination node is D_1, l_i of node E becomes $l_i + 2$, and d_i becomes 1/2 d_i.

-- (Case 3) If two nodes next to the neighbor nodes of C are throttled, l_i and d_i are changed according to the locations of the destination nodes. An example of case 3 is shown in Fig. 3 (c), where the east and north nodes of neighbor E are throttled. In this example, if the destination node is D_2, l_i of node E becomes $l_i + 4$, and d_i becomes 0.

-- (Case 4) If three nodes next to the neighbor nodes of C are throttled, the neighbor nodes can't be selected unless they are the destination nodes. An example of case 4 is shown in fig. 3 (d), where E will not be selected unless it is the destination.

To avoid livelock, we use a small amount of state in packet to record the number of times the packet has been misrouted (N_m). Once N_m reaches a threshold N_0, no more misrouting is allowed [8]. To deal with deadlock, we use a deadlock recovery scheme DISHA [9] to handle the problem. In this case, each router is equipped with an extra flit buffer to store the header flit of one of the deadlock engaged packets.

An example of the proposed fully adaptive intra-layer routing is illustrated in Fig. 4. In this example, the packet is routed from S to D. The shadow nodes are the nodes that are throttled, the value above each edge is the occupied buffer size (s_i) , and the value above each node is the diversity for each node to D (d_i). We set $\alpha = 0.7$, $\beta = 0.3$, and the buffer size is 4. Firstly, $C = S$, $l_0 = 5$, $s_0 = 4$, and $d_0 = 3$. We calculate the function value of each candidate node (node 1, node 5, node 7

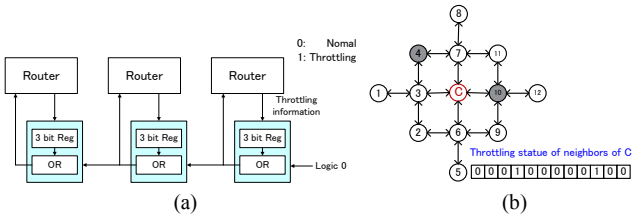


Fig.2 An example of throttling collection for one router

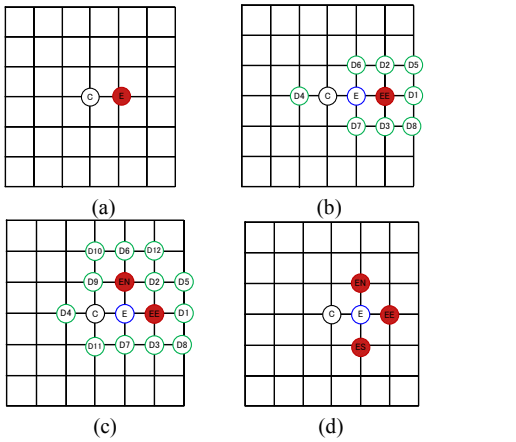


Fig.3 An example of variations of parameter value in the throttling mode

and node 11). Since node 1 is throttled, l_3 is ∞. Since node 12 is throttled, $l_1 = 6$, $l_2 = 4$, $l_4 = 6$, $d_2 = 1/2 * 4 = 2$. The function values of each neighbor node are: $f_{node1} = 1.065$, $f_{node5} = 1.065$, $f_{node7} = \infty$, $f_{node11} = 0.71$. By comparing the function value, the node 11 is selected as the next node on the path. In the same way, the route from *S* to *D* can be determined as *S* -> 11 -> 16 -> 17 -> 18 -> *D*.

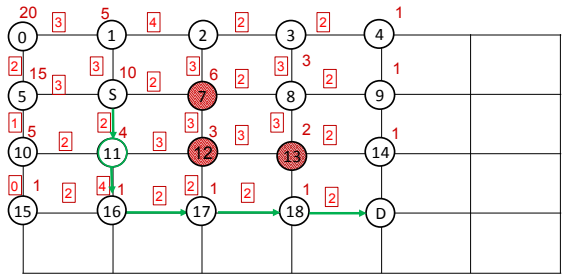


Fig. 4 An example of the intra-layer routing

C. Downward Inter-layer Routing

If there is no path to be selected to go to the destination or N_m reaches the threshold N_0, it is considered as unroutable in the current layer. In this situation, the packet is transported down to the next layer and rerouted by using the intra-layer routing algorithm. If it also can't reach the corresponding destination, it is downward one layer again. If it is unroutable in all of the layers above, it is finally down to the bottom layer. By downward the message one layer each time, the routing algorithm can balance the traffic load in the vertical layer in a certain traffic patterns. At the same time, even though the packet didn't reach the corresponding destination in the above layers , it gets close to the destination in each downward layer, which also alleviates the transportations in the bottom layer. An example of the downward routing is shown in Fig. 5. In this example, the nodes with cross are throttled nodes. When the packet reaches node *T*, it becomes unroutable. Then it is down to layer1, and rerouted to *D'*. Finally it is routed from *D'* to *D* in the vertical direction.

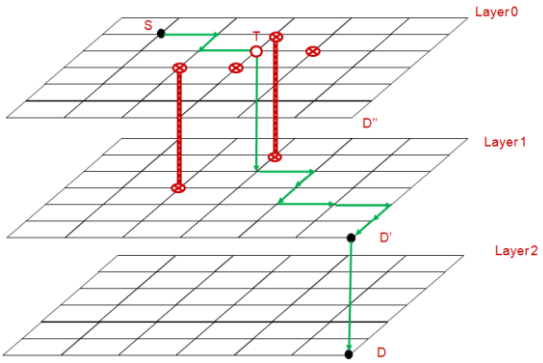


Fig. 5 An example of the inter-layer routing

IV. Experimental Results and Analysis

We implement the proposed routing algorithm by using an open source simulator Access Noxim [10], which is a combination of Noxim simulator and HotSpot simulator, and adopts the power model of Intel's 80-core processor. Noxim is a cycle-accurate SystemC NoC simulator, and HotSpot provides the architecture-level thermal model. We design two kinds of experiments on the 3D mesh topology in different network scales at different traffic patterns. Firstly, we execute the simulation in the normal mode, in which the RTM scheme is used to regulate the temperature in the network. Then we fix the number of throttled nodes and test the performance of the proposed routing algorithm. The parameters for the simulation are shown in Table I. The simulation is implemented on 2 GHz Linux workstation with 4G memory. The proposed routing algorithm is compared with TLAR-DLADR [6] and TAAR [3] on considering the network performance including average latency, energy consumption and throughput.

TABLE I
Parameters for Simulations

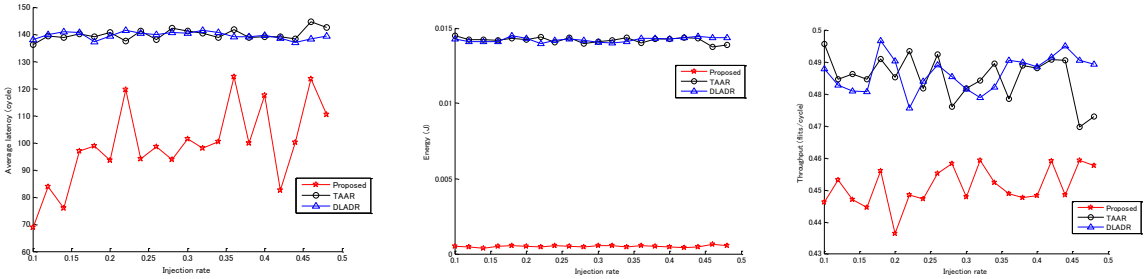
ID	Parameter	Value
1	Packet size	2~10 flits
2	Buffer size	16 flits
3	Simulation time	10^4 cycles
4	Warm up time	2000 cycles
5	α, β	0.75, 0.25
6	Mesh size	4*4*4, 8*8*4
7	Throttling scheme	VT
8	Traffic pattern	Random, shuffle, transpose

A. Results on 4*4*4 Mesh NoC in Normal Mode

We first implement the simulations on a small scale 4*4*4 mesh network. We run three simulations with different traffic patterns. The comparison results including average latency, energy consumption and throughput at random, transpose and shuffle traffic patterns are illustrated in Fig. 6, Fig. 7 and Fig. 8 respectively.

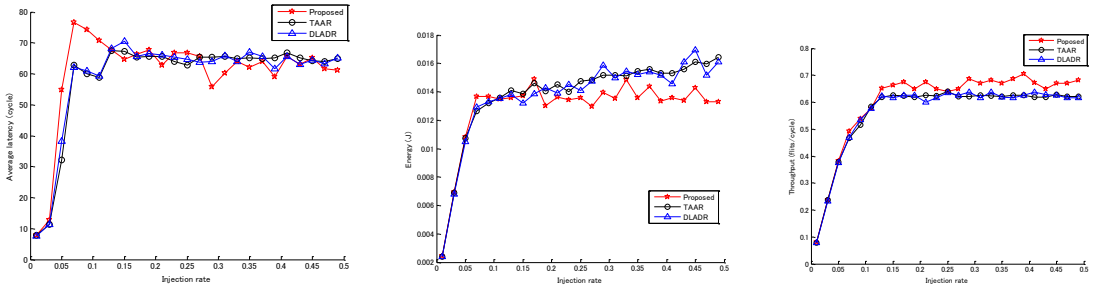
From these results we can find, our proposed routing algorithm can save the energy by comparing with TAAR and TLAR-DLADR algorithms no matter on any traffic, and the improvement is remarkable especially on random traffic. That's because our proposed algorithm introduces little additional hardware cost and the process for routing computation is very simple, which results in lower energy consumption than other algorithms. In addition, the network latency of the proposed one is not worse than the others, which shows the effectiveness of the path selection and the ability of traffic load balance. However, the throughput of the proposed one is worse than the others at random and shuffle traffic patterns. TLAR-DLADR and TAAR are based the awareness of the varying topology of the whole network, and the topology table is used in the path computation, which makes an contribution to the improvement of the network throughput. TLAR-DLADR applies a combination of deterministic routing and adaptive routing in the horizontal layer, which can increase the throughput in some traffic patterns. TAAR uses two separate physical channels for different routing modes and a multiple downward layer selection scheme based on the queuing analysis, which improves the throughput of the network.

ISBN: 978-988-14047-6-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)



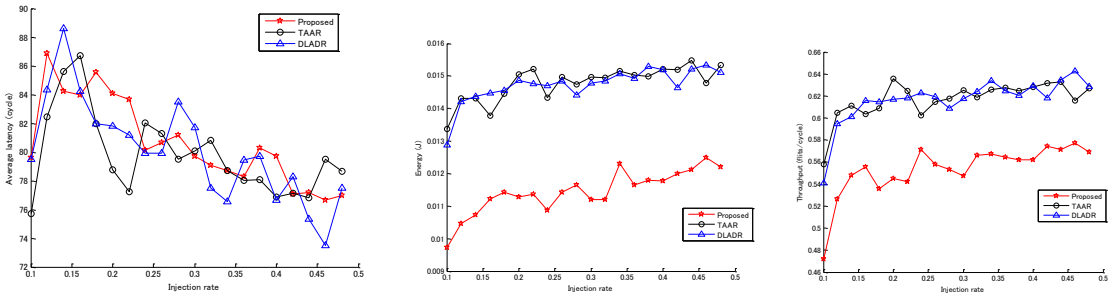
(a) Average latency (b) Energy (c) Throughput

Fig. 6 Results on 4*4*4 mesh random traffic



(a) Average latency (b) Energy (c) Throughput

Fig. 7 Results on 4*4*4 mesh transpose traffic



(a) Average latency (b) Energy (c) Throughput

Fig. 8 Results on 4*4*4 mesh shuffle traffic

*B. Results on 8*8*4 Mesh NoC in Normal Mode*

Then the simulations are executed on a larger network 8*8*4 mesh. We also implement the simulations with different traffic patterns. The comparison results including average latency, energy consumption and throughput at random and shuffle traffic patterns are shown in Fig. 9 and Fig. 10 respectively.

The simulation results illustrates that the energy consumption and latency are improved compared with the other routing algorithms, and the throughput is better than the other two algorithms on shuffle traffic. The results indicate that the effectiveness of the proposed routing algorithm is related with traffic patterns and network scales.

C. Results on Fixed Throttling Mode

Next we test our routing algorithm on the network with one fixed 1*1*2 throttling region. The simulation results on 8*8*4 mesh network at shuffle traffic pattern are shown in Fig. 11.

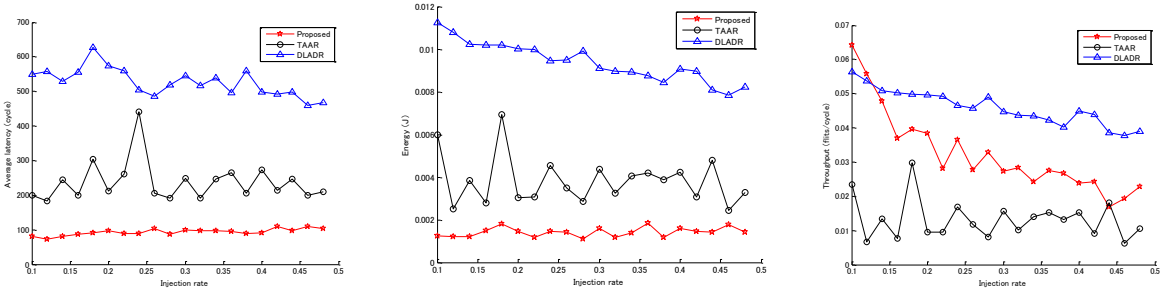
In this experiment, our proposed routing algorithm achieves better results on latency, energy and throughput by comparing with TAAR and TLAR-DLADR thermal-awarerouting algorithms, which also demonstrates the effectiveness of the proposed algorithm.

V. Conclusions

In this paper, we proposed a runtime thermal-aware fully adaptive routing algorithm for 3D NoC. The algorithm used a downward routing to balance the traffic load on the vertical layer and a fully adaptive 2D routing algorithm in which transmission distance, traffic state, route diversity, and neighbor throttling information are considered simultaneously. By comparing with other thermal-aware routing algorithms, our proposed algorithm largely saves the energy and improves the network latency in most of the cases. Especially for large scale networks, our proposed algorithm is promising more effective on the network performance.

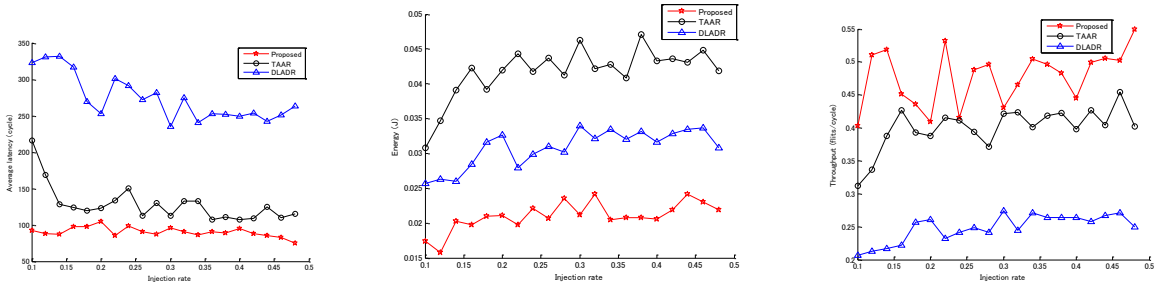
Acknowledgment

This paper is a part of the outcome of research performed under a Waseda University Grant for Special Research Projects (Project number: 2015S-112).



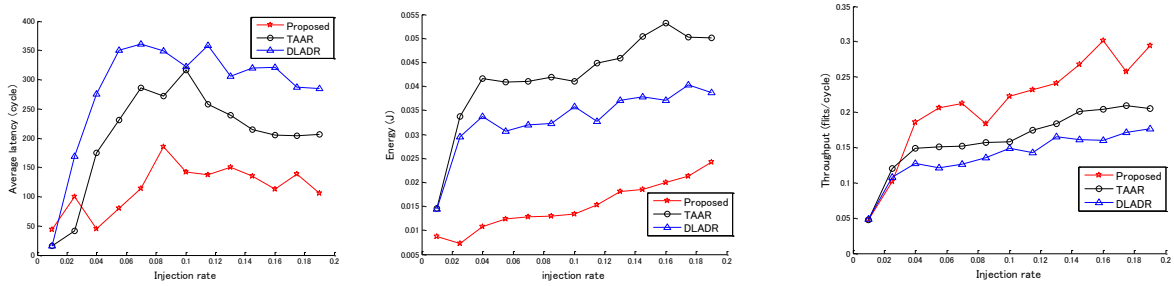
(a) Average latency (b) Energy (c) Throughput

Fig. 9 Results on 8*8*4 mesh random traffic



(a) Average latency (b) Energy (c) Throughput

Fig. 10 Results on 8*8*4 mesh shuffle traffic



a) Average latency (b) Energy (c) Throughput

Fig. 11 Results on 8*8*4 mesh shuffle traffic in fixed throttling mode

REFERENCES

[1] M. Palesi et al., “Routing Algorithms in Networks-on-Chip,” *Springer*, 2014, Chapter 12.

[2] L. Shang et al., "Thermal modeling, characterization and management of on-chip networks, *in Proc. IEEE/ACM International of Symposium on Microarchitecture (Micro)*, 2004, pp. 67-78.

[3] K.C. Chen, S.-Y. Lin, H.-S. Hung and A.-Y. Wu, “Topology-Aware Adaptive Routing for Non-Stationary Irregular Mesh in Throttled 3D NoC Systems,” *IEEE Transactions on Parallel and Distributed Systems*, 24(10), 2013, pp. 2109–2120.

[4] C.-H. Chao, "Traffic- and Thermal-Aware Run-Time Thermal Management Scheme for 3D NoC Systems", *in Proc. NOCS, 2010*, pp. 223 -230.

[5] S. Y. Lin et al., "Traffic-and Thermal-Aware Routing for Throttling Three-Dimensional Network-on-Chip System," *in Proc. IEEE Int'lSymp. VLSI Design, Automation, and Test (VLSI-DAT)*, 2011, pp. 135-138.

[6] C.H. Chao et al., “Transport Layer Assisted Routing for Non-Stationary Irregular Mesh of Thermal-Aware 3D Network-on-chip Systems,” *in Proc. IEEE Int’l SOC Conf. (SOCC)*, Sept. 2011.

[7] S. Pasricha, and Y. Zou, "A low overhead fault tolerant routing scheme for 3-D networks-on-chip," *Int. Symp. Quality Electron. Design*, 2011, pp. 598-602.

[8] W. Dally, and B. Towles, "Principles and Practices of Interconnection Networks," *Morgan Kaufmann Publishers Inc., San Francisco,* 2003.

[9] A. Patooghy, S. G. Miremadi, "Complement routing: A methodology to design reliable routing algorithm for Network on Chips," *Microprocessors and Microsystems 34*, 2010, pp.163-173.

[10] Access Noxim: http://access.ee.ntu.edu.tw/noxim/index.html.

ISBN: 978-988-14047-6-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)