Thermal-aware Dynamic Weighted Adaptive Routing Algorithm for 3D Network-on-Chip

Muhammad Kaleem[1]*, Ismail Fauzi Bin Isnin[2]
School of Computing, Faculty of Engineering
Universiti Teknologi Malaysia, Johor Bahru, Malaysia[1, 2]
Department of CS & IT, University of Sargodha, Sargodha, Pakistan[1]

***Abstract*—3D Network-on-Chip NoC based systems have severe thermal problems due to the stacking of dies and disproportionate cooling efficiency of different layers. While adaptive routing can help with thermal issues, current routing algorithms are either thermally imbalanced or suffer from traffic congestion. In this work a novel thermal aware dynamic weighted adaptive routing algorithm has been proposed that takes traffic and temperature information into account and prevents packets being routed across congested and thermally aggravated areas. Dynamic weighted model will consider parameters related to congestion and thermal issues and provide a balanced suitable approach according to the current scenario at each node. The efficiency of the proposed algorithm is analyzed and evaluated with state-of-the-art thermal-aware routing algorithms using a simulation environment. Results obtained from the simulation shows that the proposed algorithm has performed better in terms of global average delay with 17-33 percent improvement and better thermal profiling under various synthetic traffic conditions.**

Keywords*—Routing algorithms; thermal-aware; dynamic weighted model; 3D Network-on-Chip

I. INTRODUCTION

3D-IC (Three Dimensional Integrated Circuit) is capable of providing small interconnections resulting in reducing delays due to die stacking. 3D NoC based chip multiprocessors (CMP) are estimated to have higher performance with less data transmission connection cost and power consumption [1]. Conventional NoC architectures consist of processing element (PE), network interface and router in every tile. Routers provide communication mechanisms for communication among tiles along communication paths. During high traffic situations, packets have to reside in the router's buffers waiting for its turn, causing congestion in the network. Routers are a source of thermal hotspot due to high switching activity and congestion resulting in higher power density [2]. Power density of the router area is higher than power density in intellectual property IP [3]. It is also the fact, higher power consumption of the chip elements results in deteriorating heat. Routers are responsible for providing communication between IPs at the cost of high heat dissipation.

Cooling mechanisms also known as heat sinks usually exist on only one side of the chip in multi-layer 3D NoC. Hence, layers further away from the heat sink have higher possibility of the thermal hotspots [4]. Due to these thermal hotspot difficulties and aggravation of failure mechanism puts an extra strain on cooling cost of the chip and reliability reduction in 3D NoC. It is essential for designers to maintain performance of the system while reducing thermal hotspots [5]. To balance temperature distribution, various methods of thermal-aware application mapping [6] [7], floor-plan optimization [8] and thermal-aware routing [9] [10] [11] were proposed. Thermal-aware routing algorithms are classified in temporal and spatial routing algorithms. Temporal DTM (Dynamic Thermal Management) can dynamically adjust frequencies, voltages or clock cycles to reduce on-chip temperatures[12]. Usually a fully throttling scheme such as clock gating is applied to control the temperature of the thermal aggressive nodes. Hence, the temporal DTM results in reducing overall system performance but it can regulate system temperatures within short cooling time. Spatial routing algorithms are reducing thermal hotspots by diffusing traffic away from the heated regions [13]. Spatial DTM can manage thermal situations without reducing the speed of the node in terms of frequencies, voltages and clock cycles, hence, a tiny impact on the performance of the system.

Due to the lack of heat sink among the layers and poor traffic distribution, tackling the heat dissipation problem in 3D NoC is difficult. Since the center of the top layer in 3D NoC is more susceptible to thermal problems. One of the effective solutions is to divert traffic away from the center of the network or to the layers closer to the heat sink. Detoured traffic results in cooler routes yet increasing the path length. Heavy intermediate traffic results in more delays, causing congestion and induces thermal issues. Due to diverse thermal conductance between the intra layer and inter layer of the 3D NoC, the relationship between temperature and traffic behavior among NoC nodes is divergent. It is difficult to solve thermal issues without considering traffic conditions in the network. Heavy traffic can cause congestion in the network with low outflows, leads to packets stuck at router buffers waiting for its turn, dissipating heat and causing thermal difficulties. Hence, design goal of this work is to blend temperature and traffic information along with other routing parameters during the thermal control period in order to make better routing decisions. Highlights of the contribution in this work are listed below.

1) Proposed a novel thermal-aware dynamic weighted routing (TADWR) technique for dynamic distribution of traffic and heat in the 3D NoC. This technique allows packets to adaptively select their next neighbour by dynamically adjusting weights of the cost model.

*Corresponding Author.

2) TADWR works on a dynamic weight management mechanism designed to regulate new weights among the parameters to work according to network situation and need of time.

3) Extensive simulation is performed and compared the results with state-of-the-art techniques under various synthetic traffic scenarios.

The rest of the paper is organized as follows. In Section II related work to thermal-aware routing algorithms has been discussed along with its limitations. In Section III detailed methodology of the proposed routing algorithm has been presented. In Section IV results of simulations have been presented and compared with other existing routing techniques. Finally, this work has been concluded in Section V.

II. Literature Review

Thermal-aware routing algorithms have gained attention among researchers all over the world due to high switching activity in routers. Traffic congestion, hotspot formations, and packet delays can be reduced by using an efficient and effective routing algorithm. Hence, thermal-aware routing has gained considerable attention among researchers in recent years. In order to reduce on-chip temperature, the routing strategies outlined in [14] aim to route packets through a layer closer to the heat sink. Thermal-aware Selective Detour (TSD) and Thermal-aware MILP-based Detour (TMD) are the two techniques presented. This routing technique is non-adaptive detour-based application-specific routing for 3D mesh NoC. Authors examine the impact of various detour decisions on the chip's steady-state and transient-state temperature profiles, as well as the network's average packet latency. However, being non-adaptive it is hard to maintain performance at higher packet injection rates with critical applications.

Fast multi objective thermal-aware adaptive routing algorithm (FMoTAR) is introduced in [15] to optimize the thermal profile of 3D NoC. FMoTAR uses a bidirectional search to easily locate the shortest path. With a higher packet injection rate in FMoTAR, maintaining efficiency with sensitive applications is a challenge. The Q-learning mechanism [16] is used to create a proactive thermal management strategy for 3-D NoCs using a feedback-based technique. An agent learns its own behavior in an immersive environment during system activity. The reward values of the agent behavior are stored and updated in a table called Q-table in Q-learning. The average temperature of routers is assigned to packet traversing in the header of each ordinary packet (a packet that passes data between nodes). The routers' Q-table stores and updates these values. As a result, no learning packets are needed to spread thermal information across the chip. Incoming packets are routed to cooler routes based on their Q-table values, and they are detoured from high-temperature areas. However, it can be observed that Q-learning based routing is focusing on average temperatures which will be always less than the actual temperature of the router hence deprived decisions will be made during routing.

QTTAR [17] is a Q-learning-based adaptive 3D routing algorithm that improves overall node utilization by balancing inter-layer traffic distribution and offering a more precise congestion analysis to prevent RTM-related performance degradations. By balancing the distribution of overheated regions in a layer, QTTAR reduces differences in inter-layer cooling performance. By learning dynamically evolving networks, QTTAR detects regional congestion and thermal hotspots. QTTAR produces an effective route and a routing decision based on the Q-table. It uses a Q function-based routable direction selection technique to achieve routable path diversity. According to 3D symmetrical buffered clock tree for thermal variation synthesis [18] initially, sinks with similar power consumption for selecting closest to median cost of the neighbor in a 3D abstract tree topology. Second, the layer assignment of the internal node is calculated for uniform TSV distribution. Finally, after completing the thermal profile centered on the grid, the buffer, wire and exact position of TSV insertion are completed. However, at low-power 3D abstract trees and clock tree synthesis needs to be further investigated.

To balance the thermal distribution and meet the efficiency requirements, an energy- and buffer-aware fully adaptive routing algorithm is proposed [19]. To balance the flow of thermal energy and reduce network congestion, a network state feature model is built that takes into account both historical and current network states. Fully adaptive routing indicates improvement in thermal distribution without performance degradation to lower the temperature and meet the performance specifications of high priority packets. A collaborative thermal-aware adaptive routing (CTTAR) scheme [4] to synchronize network traffic and thermal information is presented. Since unnecessary packet switching causes hotspots, the CTTAR first employs dynamic buffer change, which can restrict the routing resource around overheated regions to slow the rate of temperature increase based on expected thermal details. Since the routers in the overheat area switch less packets, the dynamic buffer change will reduce heat production and diffusion. CTTAR converts overheated areas into congested areas. However, it is not suitable for complicated hotspot distribution. GTDAR [20] is a game theory-based thermal delay-aware adaptive routing system that transfers long-term thermal information into short-term traffic information, allowing it to orchestrate traffic and thermal information more effectively and reduce the temperature problem into a traffic problem. The traffic load distribution in GTDAR's network is unbalanced.

In ATAR [21] packet is traversed in the network on the bases of its weighted cost model calculation. Highest weightage is given to temperature and decreasing weightage to the subsequent factors i.e. path length, neighbor queue length and its workload. Cost is calculated to find the potential best neighbor for forwarding the packet. At the source end, all best neighbors until destinations are computed to make a path list. Path list contains all the neighbors that will be used to deliver the packet. ATAR has taken all decisions at an individual node level. ATAR is lacking its view of congested and thermally active regions. If a particular neighbor has the least cost but if it is part of a congested or thermally hostile region. ATAR has little or no ability to identify congested regions due to an inflexible cost model. If such nodes are selected for

traversing packets, this leads to enhance congested areas hence higher heat dissipation and higher thermal issues. Centre of the network is naturally prone to become congested. Sending more packets to the congesting or throttling region can be fatal.

There are multiple paths between source and destination. If a routing algorithm dynamically adjusts its cost model according to intermediate nodes current conditions then we can adaptively adjust and choose better paths to reach its destination. It is observed from literature; apart from thermal-aware selection of algorithms, it is also necessary to include various other factors such as neighboring node temperatures, shortest path length, detection of congestion situation, and next router queue length.

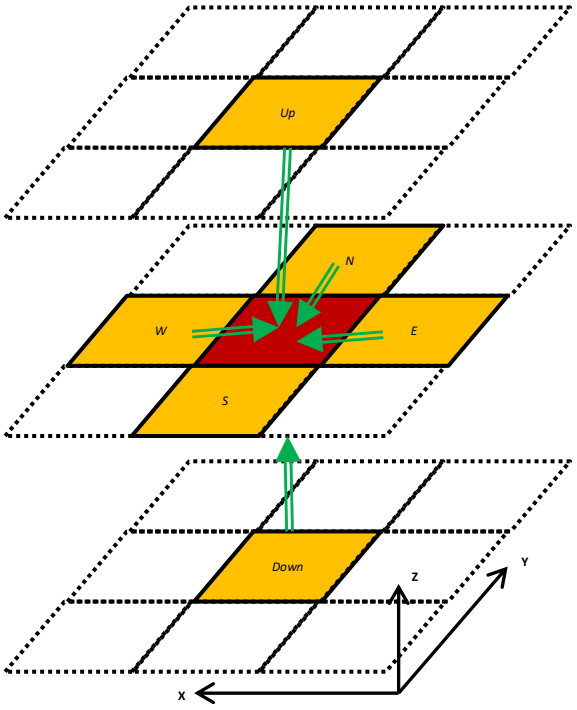


Fig. 1. Central node Neighbors of 3D NoC.

III. Methodology

Nodes in NoC get heated up due to high power usage. Even a smallest of unwanted operations can play its role in increasing the temperature and that could lead to a disaster. Therefore, uncontrolled and excessive movement in the network should be avoided. Consider if the router is already busy in sending upstream packets to the desired output ports. New arriving packet has to wait for its turn in the packet queue at the input buffers. These excessive packets stall in the input buffers accelerate the heating process. In Fig. 1, it can be observed that if the central node of the network is experiencing heavy congestion then very quickly output buffers of neighboring nodes also start to observe congestion. Which means the congestion region will extend itself if it is not handled in time. As congestion is the primary cause of thermal aggravation and throttling. Congestion cannot only occur in the center of the network but it can occur anywhere within the chip. Therefore, packets cannot be allowed to just detour away from the center of the network to get thermal stability. Dynamic mechanism is required to handle and adjust according to the need and requirement of the situation within a chip.

In this work a technique that can consider temperature, congestion and most importantly allow packets to adaptively select their next neighbor by dynamically adjusting weights of the cost model is proposed. Dynamically adjusting the cost model means, when NoC is thermally stable with no congestion proposed algorithm should consider shortest path on priority and allow packets to choose best possible paths with least workload to reach the destination. As soon as congestion starts occurring, the dynamic model should take actions adjusting it to choose paths with less congestion in reaching destination. Similarly, when thermal issue arises then priority will be given to the thermal issue to bypass the thermally active regions. In case of multiple issues, arising at the same time proposed dynamic weighted model will consider all parameters and provide a balanced approach suitable for the situation and need.

A. Dynamic Weight Adjustment Model

In this work, following parameters are considered as given below:

1) Path Length is denoted by L, it is the number of hops a packet takes to reach its destination from source. Consider (x1, y1, z1) is the source and (x2, y2, z2) is its destination so path length will be calculated from the equation 1 below.

$$L = m|x_1 - x_2| + n|y_1 - y_2| + o|z_1 - z_2| \quad (1)$$

According to equation (1) m is the number of rows in x dimension n is the number of columns in y dimension and o is the number of layers in z dimensions.

2) Candidate node temperature is denoted by T and it is the current temperature of the immediate candidate node next is succession.

3) Candidate node throughput is denoted by W and it is the current throughput of the immediate candidate node next is sequence.

4) Candidate node Q length denoted by Q and it is the current Q length of the intermediate node next under consideration.

This work is using the cost model presented in ATAR [21] equation (2). Where T is temperature, L is path length, Q is next router queue length and W is node throughput. T is responsible for temperature influence, L is responsible to cover path length till destination, Q is responsible for determining congestion in next buffer and W is responsible to reflect load on the link.

$$\text{cost} = a_1.T + a_2.L + a_3.Q + a_4.W \quad (2)$$

Initially equal weights are assigned to i.e. $a_1 = a_2 = a_3 = a_4$ where a_1, a_2, a_3, a_4 are respective weights of [T, L, Q, W] whereas $\sum_1^j a_j = 1$ and $a_j \neq 0$, such that $j \in \{1, 2, 3, 4\}$. As traffic load increases and more and more packets start traversing to reach their destination, packets start occupying

buffers hence Q length will start to vary. Routers with the smallest Q length will be given priority as a next hop for the packet. Therefore, it will keep assigning new weights to the highest values of Q length. Similarly, with the passage of time throughput of the node will also reduce as traffic and congestion is increased. So, highest new weights will be assigned to nodes with highest throughput in order to choose the next hop node with highest throughput. As the traffic conditions are worsening especially under higher packet injection rates the congestion occurs and node temperature will begin to rise. If a node temperature exceeds its previous temperature then it will be known as new peak value and new weight will be calculated and assigned to the new peak value in a node. If a node experiences a temperature less than the new peak value a weight will be assigned accordingly.

As all the parameters have their own intended weights. Technique cannot simply grant all parameters to use their intended weights and violate $\sum_1^j a_j = 1$ condition. Hence, weight management mechanism is required to regulate new intended weights among the parameters. For new weight formation first calculate the weight difference β_i given in equation (3). β_i is a difference between previous weight and new intended weight for any parameter.

$$\beta_i = \text{Previous\_weight}_i - \text{Intended\_weight}_i \quad (3)$$

Where i= {1, 2, …C} and C is the number of all parameters under consideration. After calculating the difference between previous weights and new weights of all parameters, a weight bucket (WB) is formed. WB is a sum of all weight differences in all parameters equation (4).

$$WB = \sum_{i=1}^{c} \beta_i \quad (4)$$

Where i= {1, 2, …C} and C is the number of all parameters under consideration. After calculating the weight bucket, weight fraction is calculated. Weight faction WF is the equal share for all the parameters weight in the consideration equation (5).

$$WF = \frac{\sum_{i=1}^{c} \beta_i}{C} \quad (5)$$

Where i= {1, 2, …C} and C is the number of all parameters under consideration. In the next step, weight fraction is added to the intended weight of the particular parameter and a new weight α_i is formed in equation (6) for calculation of the cost model given in equation (1).

$$\alpha_i = \text{Intended\_weight}_i + WF \quad (6)$$

It can be observed that each node in the network will have its own weights and preferences according to the current state and condition of a particular node. Now if the routing algorithm has more than one possible neighbor to traverse, a better decision could be made according to the scenario a node is currently experiencing.

Algorithm 1 TADWR

Input: Source node, destination node, node temperature, workload, path length and queue length
Output: path
1: **function TADWR**(s_node , d_node, route_data)
2: **if** s_node= d_node then
3: directions ← direction_local
4: **else**
5: **set** i_node ← s_node
6: **while** i_node ≠ d_node do
7: **set** dir ← getAvailableDirections(i_node)
8: **for** k Є dir do
9: **set** e ← i_node
10: $Intended\_weight_1$ ← getTempIntendedWeight(i_node,dir)
11: $Intended\_weight_2$ ← getPathlength(i_node, d_node)
12: $Intended\_weight_3$ ← getqueuelength(i_node, dir)
13: $Intended\_weight_4$ ← getWorkload(i_node,dir)
14: $\beta_i = Previous\_weight_i - Intended\_weight_i$
15: $WB = \sum_{i=1}^{c} \beta_i$
16: $WF = \frac{\sum_{i=1}^{c} \beta_i}{C}$
17: $\alpha_i = Intended\_weight_i + WF$
18: **set** cost ← α_1.e.T + α_2.e. L + α_3.e. Q + α_4.e. W
19: **if** V[s_node][i_node] +cost < V[s_node][e.next] then
20: **set** V[s_node][e.next]← cost
21: **end for**
22: **set** i_node ← mincostNode(V, s_node, directions)
23: **set** directions ← direction_mincostnode(s_node, i_node)
24: **end while**
25: **end else**
26: **return** directions

B. Thermal-Aware Dynamic Weighted Routing (TADWR)

Pseudo code for TADWR is presented in Algorithm 1. Algorithm takes arguments such as source node, destination node and route data (node parameters i.e. L, T, Q, W). Source node is denoted by s_node represents the node from which the packet has been initiated. Destination node is denoted by d_node, representing the node at which the packet will be terminated. In the beginning, the algorithm will check location of source node and destination node. If it is addressing itself, it will be terminated by returning direction Local in line 2-3. Checking for all possible directions available for the node in line 7-8. Get parameter T, L, Q, W values from i_node. Find the intended weight for the intermediate candidate node all parameters line 10-13. Calculate change in weight (either weight loss or gain) in all parameters with respect to previous weights line 14. Calculate sum of all the weight changes and assigned to weight bucket WB line 15. Calculate weight fraction WF according to number of criteria and to be added in respective intended weights line 16-17. The weighted cost sum is then calculated. In lines 19-20 the cost matrix is updated, if the calculated cost is less than the current cost. Now the minimum cost node will become the new i_node (intermediate node) and the minimum cost nodes direction is pushed in directions. In order to reduce the delay caused by the loop in the algorithm, parallel architecture is used to reduce the computation delay.

IV. Simulation Results and Discussion

TADWR routing algorithm has been simulated in Access Noxim [22]. Access Noxim simulator is a cycle accurate simulator integrated with HotSpot [23] and Noxim [24]. Noxim is designed by using the System C liberary. SystemC is an open-source hardware description language designed in C++. Noxim is portable and can run on any SystemC based infrastructure. NoC parameters can be defined and set using a command based interface, e.g. user can set number of nodes in the network, traffic distribution system, buffer capacity, network size and dimensions, packet sizes, routing algorithm, traffic distribution time, packet injection rate, etc. Noxim can evaluate capability of NoC in terms of delay, throughput and power consumption. Noxim also possesses a transaction level mode where detailed analysis of even a single transaction can be made. HotSpot is responsible for providing architectural level thermal models. Overall Access Noxim is capable of providing thermal model, power model and network model of 3D network-on-chip.

A. Simulation Setup

To evaluate the performance of the proposed routing algorithm, 8 x 8 x 4 3D NoC is considered. Parameters used in the simulation are listed in Table I. Different synthetic traffics are simulated to test the ability of TADWR as compared to its counterparts. TADWR is compared with state-of-the-art ATAR and fully adaptive routing algorithms. Each simulation is carried out for 200,000 cycles with different PIR (packet injection rate). PIR is varying from 0.02 to 0.22 with the interval of 0.02 (flits/cycle/node). A 0.02 PIR means each node sends 0.02 flits every clock cycle. The instance at which a packet is injected depends on the distribution of the time interval.

TABLE I. Simulation Parameters

Parameters	Value
Network Dimension	8 x 8 x 4
Simulation Time (Cycles)	200,000
Warm-up Time (Cycles)	10,000
Buffer Size (Flits)	16
Packet size (Flits)	2~10
Packet injection rate (flits/cycle/node)	0.02-0.22
Packet injection interval	0.02
Traffic Pattern	Random, Shuffle, Hotspot, Bit-Reversal

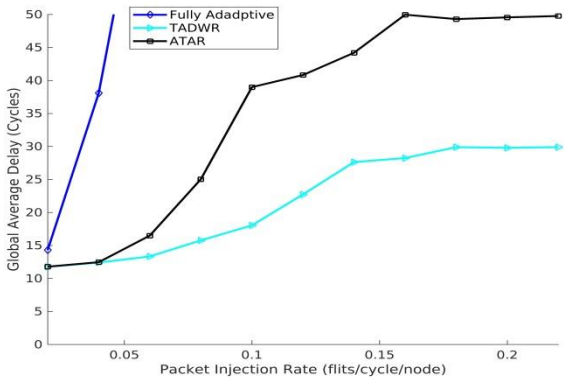
B. Performance Evaluation

The traffic patterns applied in this section are bit-reversal, random, shuffle and hotspot traffic. Global average delay (cycle) chart is presented for each simulation. ATAR uses fixed weighted cost model and TADWR uses dynamic weighted cost model whereas, fully adaptive does not use any cost model criteria for routing. The global average delay diagram of fully adaptive, ATAR and TADWR under bit-reversal traffic is presented in Fig. 2(a). It can be observed that for the injection rate between 0.02 to 0.10 with the interval of 0.02 both ATAR and TADWR have similar results but as injection rate increases it has begun to show diverse behaviors. Examining the graph reveals that the after 0.10 global average delay of TADWR with dynamic weight assignment has sustained as compared to ATAR, which has a fixed weighted cost model. As far as fully adaptive routing is concerned, it can be observed that it has rocketed up at PIR 0.10 due to unawareness of traffic congestion situation.

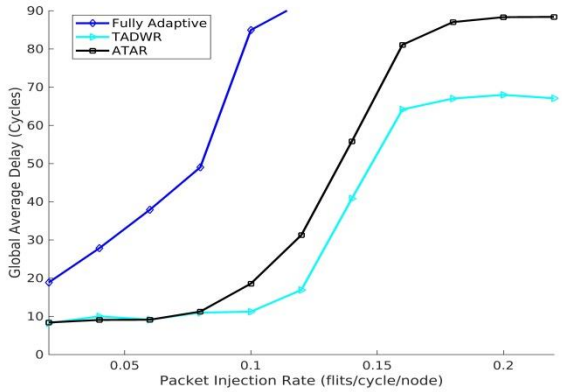
The global average delay diagram of fully adaptive, ATAR and TADWR under random traffic is presented in Fig. 2(b). It can be observed that for injection rate between 0.02 to 0.10 with the interval of 0.02 both ATAR and TADWR have similar results but as injection rate starts to grow the difference start to get more apparent. Analyzing the graph, it is obvious that after 0.10 global average delay of TADWR with dynamic weight assignment has sustained as compared to ATAR, which has a fixed weighted cost model. As far as fully adaptive routing is concerned, fully adaptive has sharply shot at PIR 0.10 due to unawareness of traffic congestion situation.

The global average delay diagram of fully adaptive, ATAR and TADWR under shuffle traffic is presented in Fig. 2(c). It can be observed that for injection rate between 0.02 to 0.10 with the interval of 0.02 both ATAR and TADWR have similar results but as injection rate starts to increase greater that PIR 0.10 the difference between ATAR and TADWR become clear. Analyzing the graph, it is obvious that after 0.10 global average delay of TADWR with dynamic weight assignment is better than ATAR, which has a fixed weighted cost model. As far as fully adaptive routing is concerned, it can be seen that fully adaptive has skied sharply around PIR 0.12 due to unawareness of traffic congestion situation.

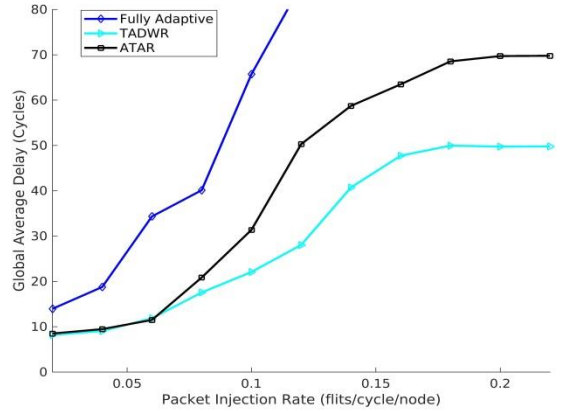
The global average delay diagram of fully adaptive ATAR and TADWR under hotspot traffic with 20 percent is presented in Fig. 2(d). It can be observed that for injection rate between 0.02 to 0.10 with the interval of 0.02 both ATAR and TADWR have identical results but as injection rate increases change in behavior is obvious. Examining the graph tells that the after 0.10 global average delay of TADWR with dynamic weight assignment has sustained as compared to ATAR which has a fixed weighted cost model even in highly demanding traffic conditions. As far as fully adaptive routing is concerned, it can be observed that it has started to rise fairly early due to unawareness of traffic congestion situation and unable to cope with stressed traffic such as hotspot.



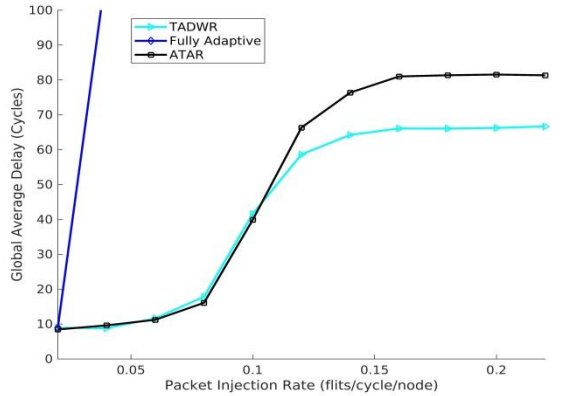
(a)



(b)



(c)



(d)

Fig. 2. Comparison of Global Average Delay under Various Traffic Patterns (a) Bit-reversal Traffic (b) Random Traffic (c) Shuffle (d) Hotspot Traffic.

C. Thermal Performance

Simulations are conducted for steady state temperatures. The thermal results are shown in the Fig. 3 for random traffic pattern with packet injection rate (PIR) of 0.02 (flits/cycle/node). The thermal image shows the drop in peak temperatures of the on-chip network of TADWR as compared to ATAR is around 6 K and massive 20K with respect to fully adaptive routing during extensive simulations and results comparison.

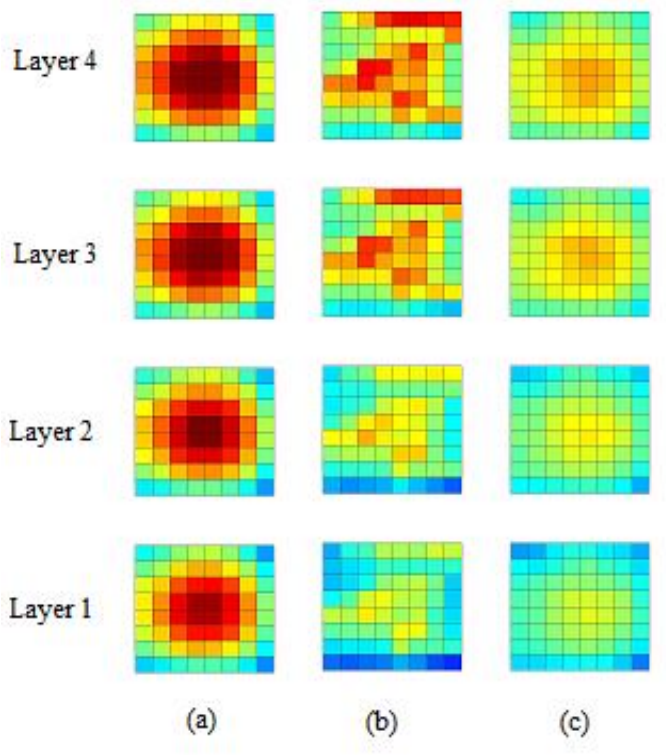


Fig. 3. Thermal Profile Comparison under Random traffic (a) Fully Adaptive Routing (b) ATAR (c) TADWR.

This indicates that TADWR achieves better thermal optimization with respect to other routing algorithms considered in this study. The simulations at each PIR have been repeated for a number of times to guarantee the accuracy of results. Fully adaptive shown in Fig. 3(a) has the highest thermal profile; this is due to the fact that fully adaptive routing does not consider thermal parameters during the routing process. In the case of ATAR, temperature imbalance occurs due to longer routes. Consequently, increases in congestion in the network leads to high thermal profile in the 3D NoC shown in Fig. 3(b). TADWR provides a better thermal profile than other routing algorithms. Thermal profile for TADWR at PIR 0.02 is illustrated in Fig. 3(c). On higher PIR, the peak temperature of the top layer is increased up to 8 K at PIR 0.10 and 10 K at PIR 0.22 with respect to thermal profile of TADWR at PIR 0.02.

V. Conclusion

This work proposes a thermal-aware dynamic weighted routing TADWR. TADWR can consider temperature, congestion and most importantly allow packets to adaptively select their next neighbor by dynamically adjusting weights of the cost model. Weight management mechanism is designed to regulate new intended weights among the parameters to work according to network situation and need of time. In contrast to previous work, proposed approach achieves better and balanced thermal distribution, improved network efficiency, and less hotspot in a chip. TADWR performs extremely better when in heavy packet injection rates in terms of global average delay having 17-33 percent improvement under different synthetic traffic scenarios. Thermal profiling

of TADWR is also clearly better than other routing algorithms. This work is limited to fully-connected 3D mesh topology. It can be further extended to become fault-aware in future.

ACKNOWLEDGMENT

The research is supported by Ministry of Higher Education Malaysia (MOHE) and conducted in collaboration with Research Management Center (RMC) at the Universiti Teknologi Malaysia (UTM) under Fundamental Research Grant Scheme with grant number: R.J130000.7851.5F029. The authors appreciate greatly for the support.

REFERENCES

[1] E. Fusella and A. Cilardo, "Lattice-Based Turn Model for Adaptive Routing," IEEE Trans. Parallel Distrib. Syst., no. 1, p. 1, 2018.

[2] C.-H. Chao, K.-C. Chen, T.-C. Yin, S.-Y. Lin, and A.-Y. A. Wu, "Transport-layer-assisted routing for runtime thermal management of 3D NoC systems," ACM Trans. Embed. Comput. Syst., vol. 13, no. 1, p. 11, 2013.

[3] E. Taheri, M. Isakov, A. Patooghy, and M. A. Kinsy, "Addressing a New Class of Reliability Threats in," IEEE Trans. Comput. Des. Integr. Circuits Syst., vol. PP, no. c, p. 1, 2019.

[4] L. Shen, N. Wu, G. Yan, and F. Ge, "Collaborative thermal-and traffic-aware adaptive routing scheme for 3D Network-on-Chip systems," IEICE Electron. Express, pp. 18–20200425, 2021.

[5] D. Lee, S. Das, and P. P. Pande, "Analyzing power-thermal-performance trade-offs in a high-performance 3D NoC architecture," Integration, vol. 65, pp. 282–292, 2019.

[6] M. Abdollahi, Y. Firouzabadi, F. Dehghani, and S. Mohammadi, "THAMON: Thermal-aware High-performance Application Mapping onto Opto-electrical network-on-chip," J. Syst. Archit., p. 102315, 2021.

[7] W. Liu et al., "Thermal-aware Task Mapping on Dynamically Reconfigurable Network-on-Chip based Multiprocessor System-on-Chip," IEEE Trans. Comput., 2018.

[8] S. Balakrishnan and R. Venkatesan, "Splay Tree Hybridized Multicriteria ant Colony and Bregman Divergencive Firefly Optimized Vlsi Floorplanning," 2021.

[9] K. N. Dang, A. Ben Ahmed, A. Ben Abdallah, and X.-T. Tran, "HotCluster: A thermal-aware defect recovery method for Through-Silicon-Vias Towards Reliable 3-D ICs systems," IEEE Trans. Comput. Des. Integr. Circuits Syst., 2021.

[10] Z. Shirmohammadi, M. Mahmoudi, and M. Rostamnezhad, "Int-TAR: An Intelligent Thermal-Aware Routing Algorithm for 3D NoC," J. Electr. Comput. Eng. Innov., 2021.

[11] S. S. Kumar, A. Zjajo, and R. van Leuken, "Immediate Neighborhood Temperature Adaptive Routing for Dynamically Throttled 3-D Networks-on-Chip," IEEE Trans. Circuits Syst. II Express Briefs, vol. 64, no. 7, pp. 782–786, 2017.

[12] D. Lee, S. Das, J. R. Doppa, P. P. Pande, and K. Chakrabarty, "Performance and Thermal Tradeoffs for Energy-Efficient Monolithic 3D Network-on-Chip," ACM Trans. Des. Autom. Electron. Syst., vol. 23, no. 5, p. 60, 2018.

[13] N. Rohbani, Z. Shirmohammadi, M. Zare, and S.-G. Miremadi, "LAXY: A Location-Based Aging-Resilient Xy-Yx Routing Algorithm for Network on Chip," IEEE Trans. Comput. Des. Integr. Circuits Syst., vol. 36, no. 10, pp. 1725–1738, 2017.

[14] P. Mukherjee, N. Chatterjee, and S. Chattopadhyay, "Thermal-aware detour routing in 3D NoCs," J. Parallel Distrib. Comput., 2020.

[15] A. Majumdar, R. K. Dash, J. L. Risco-Martín, and A. K. Turuk, "FMoTAR: a fast multi-objective thermal aware routing algorithm for three-dimensional network-on-chips," in Proceedings of the 50th Computer Simulation Conference, 2018, p. 12.

[16] N. Shahabinejad and H. Beitollahi, "Q-Thermal: A Q-Learning-Based Thermal-Aware Routing Algorithm for 3-D Network On-Chips," IEEE Trans. Components, Packag. Manuf. Technol., vol. 10, no. 9, pp. 1482–1490, 2020.

[17] S. C. Lee and T. H. Han, "Q-Function-Based Traffic-and Thermal-Aware Adaptive Routing for 3D Network-on-Chip," Electronics, vol. 9, no. 3, p. 392, 2020.

[18] D. K. Oh, M. J. Choi, and J. H. Kim, "Thermal-aware 3D Symmetrical Buffered Clock Tree Synthesis," ACM Trans. Des. Autom. Electron. Syst., vol. 24, no. 3, p. 28, 2019.

[19] J. Wang, H. Gu, Y. Yang, and K. Wang, "An energy-and buffer-aware fully adaptive routing algorithm for Network-on-Chip," Microelectronics J., vol. 44, no. 2, pp. 137–144, 2013.

[20] K.-C. Chen, "Game-based thermal-delay-aware adaptive routing (gtdar) for temperature-aware 3d network-on-chip systems," IEEE Trans. Parallel Distrib. Syst., vol. 29, no. 9, 2018.

[21] R. Dash, A. Majumdar, V. Pangracious, A. K. Turuk, and J. L. Risco-Martín, "ATAR: An Adaptive Thermal-Aware Routing Algorithm for 3-D Network-on-Chip Systems," IEEE Trans. Components, Packag. Manuf. Technol., no. 99, pp. 1–8, 2018.

[22] K.-Y. Jheng, C.-H. Chao, H.-Y. Wang, and A.-Y. Wu, "Traffic-thermal mutual-coupling co-simulation platform for three-dimensional network-on-chip," in Proceedings of 2010 International Symposium on VLSI Design, Automation and Test, 2010, pp. 135–138.

[23] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "HotSpot: A compact thermal modeling methodology for early-stage VLSI design," IEEE Trans. very large scale Integr. Syst., vol. 14, no. 5, pp. 501–513, 2006.

[24] V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti, "Noxim: An open, extensible and cycle-accurate network on chip simulator," in 2015 IEEE 26th international conference on application-specific systems, architectures and processors (ASAP), 2015, pp. 162–163.