OCRA: An Oblivious Congested Region Avoiding Routing Algorithm for 3D NoCs

Maede Safari
Department of Computer Engineering, Sharif University of Technology
Tehran, Iran
masafari@ce.sharif.edu

Nezam Rohbani
School of Computer Science, Institute for Research in Fundamental Science (IPM)
Tehran, Iran
rohbani@ipm.ir

Mohammad Arman Soleimani
Department of Computer Engineering, Sharif University of Technology
Tehran, Iran

Hamid Sarbazi-Azad
School of Computer Science, Institute for Research in Fundamental Science (IPM) and Department of Computer Engineering, Sharif University of Technology
Tehran, Iran

ABSTRACT

The Three-Dimensional Network on Chip (3D NoC) is an interconnection architecture designed to address the increasing communication demands between processing cores. However, as traffic and power density continues to rise, efficient traffic management and thermal regulation within these chips have become crucial issues. One common problem encountered in mesh-based NoC structures, regardless of the routing approach employed, is traffic congestion in the central region. This paper introduces a novel routing algorithm named an Oblivious Congested Region Avoiding Routing Algorithm (OCRA) to evenly distribute packet flow across the entire network.

OCRA addresses the traffic imbalance by statically configuring specific routers located in the east and west of each layer in a 3D NoC using YXZ and YZX and configuring other routers using XYZ. This configuration aims to minimize traffic congestion in the network, reduce total queuing delay and improve packet latency. The paper explores various configurations for OCRA, and simulation results on a cycle-accurate simulator indicate that a specific configuration known as Pyramidical achieves a 51.67% improvement in traffic load distribution across the network. Additionally, this configuration reduces average queuing delay by 48.7% and improves the performance of the saturated 3D NoC by 35.24% with no impact on chip area.

CCS CONCEPTS

• **Networks** → **Network on chip**; **Traffic engineering algorithms**; **Routing protocols**; • **Computer systems organization** → *Interconnection architectures.*

ACM Reference Format:
Maede Safari, Nezam Rohbani, Mohammad Arman Soleimani, and Hamid Sarbazi-Azad. 2023. OCRA: An Oblivious Congested Region Avoiding Routing Algorithm for 3D NoCs. In *16th edition of International Workshop on Network on Chip Architectures (NoCArc '23), October 28, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3610396.3618092

NoCArc '23, October 28, 2023, Toronto, ON, Canada

ACM ISBN 979-8-4007-0307-2/23/10…$15.00
https://doi.org/10.1145/3610396.3618092

1 INTRODUCTION

By increasing the number of processing cores, accelerators, and shared/dedicated memory size in Systems on Chip (SoCs) and processors, utilizing Three-Dimensional Network on Chip (3D NoC) has become a mandatory option for communication [7, 14]. In 3D NoC, different layers of routers and links are stacked together with processing cores or other components of the chip, and they are connected using Through-Silicon Vias (TSVs). This stacked structure minimizes the physical distance between different nodes. However, as the density of generated packets per area/volume increases, the congestion rate and the rate of blocked region generation also increase considerably compared to two-dimensional NoCs.

Mishra et al. [12] have demonstrated that the central nodes of a Network-on-Chip (NoC) employing an oblivious routing algorithm experience approximately twice as much congestion compared to the border nodes in a two-dimensional (2D) NoC. Our simulations, utilizing AccessNoxim [9], reveal that this phenomenon is further exacerbated in three-dimensional (3D) NoCs. The imbalanced traffic load results in some congested and blocked routers which lead to a significant increase in total waiting time and average packet latency across the network, primarily caused by the congested nodes.

To reduce congestion in the network, various adaptive routing algorithms have been proposed in the literature. Two major approaches for achieving this goal are Oblivious and traffic- and temperature-aware techniques. Oblivious adaptive techniques, such as Odd-Even [6], Repetitive Turn Model (RTM) [17], and Minimum Pressure Turn Model (MPTM) [2] provide a wide path diversity but do not take congestion and other run-time information into consideration when routing packets. These techniques offer several advantages, including larger path diversity, low complexity, and latency of routers, as well as a reduced need for additional routing data flowing in the network. However, as shown later, they cannot alleviate the traffic imbalance problem between central and border routers. On the other hand, traffic- and temperature-aware adaptive techniques utilize more sophisticated information from the network to make packet routing decisions and forward packets through less congested paths. However, these techniques introduce additional latency and complexity to the router structure, as well as require network status information to pass through the network.

Some techniques in this category leverage neural networks or table-based reinforcement learning techniques to determine the next hop at each node [10, 15]. However, these techniques can impose significant hardware overhead on each router by storing neural networks or large Q-tables and making router architecture more complex.

To reduce the traffic imbalance problem between central and border routers while keeping the router complexity and hardware overhead negligible, this work proposes an Oblivious Congested Region Avoidance (OCRA) routing algorithm that statically configures the network into different regions and routes packets through the network using fixed Dimension Order Routing (DOR) algorithms for each region. In OCRA, the nodes located at the borders of each layer of the network, are configured with either the YZX or YXZ routing algorithms (for east and west border routers) and either XYZ or XZY routing algorithms (for north and south border routers). By utilizing border routers more effectively, the congestion at the center of each layer in the 3D NoCs decreases and traffic is distributed more evenly through different layers.

Simulation results on the cycle-accurate AccessNoxim [9] simulator show that OCRA improves traffic load distribution of the 3D NoC by 51.67% and reduces the total waiting time by 48.7% compared to the state-of-the-art 3D NoC routing techniques found in the literature. Furthermore, by routing packets through the less congested area, the average packet latency of the network is reduced by 48.7%. Notably, these improvements are achieved without introducing any extra packet transmission over the network and with limited modifications to the routers, thereby imposing negligible area overhead.

The rest of this paper is organized as follows. The related routing algorithms are reviewed in section 2. Section 3 explains the motivation behind the proposed method followed by the detail of the OCRA routing algorithm. Simulation setup and results are given in Section 5 and the paper conclusion is presented in Section 6.

2 RELATED WORK

Thermal and traffic management in NoC architectures has been the subject of extensive research, leading to the development of various techniques. These techniques provide adaptive routing algorithms and can be broadly categorized into two groups: oblivious and traffic- and temperature-aware routing algorithms. The former methods distribute traffic congestion through the network by providing larger path diversity oblivious to the network condition. While the latter methods collect the congestion information and select a less-congested path between a source and destination pair.

Some oblivious adaptive routing algorithms such as Odd-Even, RTM, and MPTM provide wide path diversity (while restricting some turn models for guaranteeing deadlock freedom) and try to deliver packets through different available paths to distribute traffic congestion more evenly. In the Odd-Even turn model, different turns in odd and even columns are prohibited to improve performance. In the RTM and MPTM methods, the repetitive distance has been increased to 3 to further improve performance. However, because of the lack of traffic congestion information in these methods, they would select some congested paths while there exist other free paths.

Traffic- and temperature-aware adaptive routing algorithms dynamically adjust system behavior based on real-time feedback and run-time conditions, offering more balanced traffic congestion. For

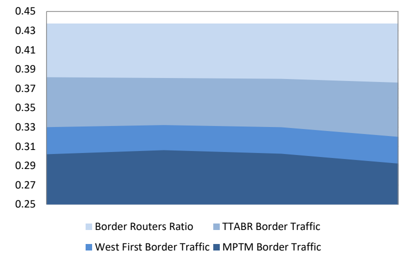


Figure 1: The ratio of border routers vs. the ratio of border traffic for XYZ, West-first, and Negative-first routing algorithms in a 3D NoC.

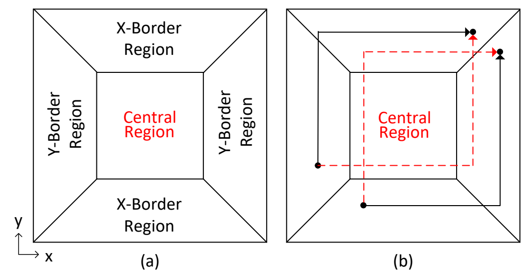


Figure 2: (a) Top view of the three distinct border regions in a single layer of a 3D NoC, (b) Routing paths of border regions to minimize entrance to the central region. Top-down view.

example, a Traffic and Throttling-Aware Routing (TTAR) is proposed in [11], which avoids paths along with the neighbors of the throttled nodes. In [3], Chao, et al. proposed a Transport-Layer Assisted Routing (TLAR) that utilizes the topology information in the transport layer to guarantee successful packet delivery in Non-Stationary Irregular Mesh (NSI-mesh). In [5], Chen et al. proposed a Traffic-balanced Topology-aware Multiple Routing Adjustment (TTMRA) that selects a cascaded node to extend the lateral routable area of each layer. Traffic- and Temperature-aware Adaptive Beltway Routing (TTABR) algorithm [4] employs both minimal and non-minimal beltway paths to detour congested regions. TTABR adaptively selects the minimal or beltway paths based on traffic information. Dynamic-XYZ (DyXYZ) [8] selects the less-congested neighbor in each hop and provides a fully-adaptive routing algorithm. Some other techniques, based on machine learning, learn the best routing path for a packet based on the network traffic information during normal operation [10, 15]. Although all the abovementioned congestion-aware routing algorithms improve traffic distribution, they imply significant performance overhead for collecting traffic information during run-time, running neural networks, or saving large Q-tables for each node during packet routing.

3 PROPOSED ROUTING ALGORITHM

In this section, we provide an overview of the motivation behind the proposed technique, followed by the proposed OCRA routing algorithm.

3.1 Motivation

Although different adaptive routing algorithms (including oblivious and traffic- and temperature-aware methods) alleviate network congestion, they still suffer from non-uniform traffic load distribution, especially congesting central routers while under-utilizing border routers. To illustrate this issue, some motivational experiments

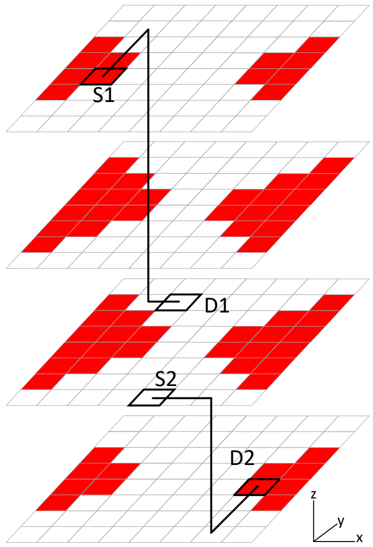


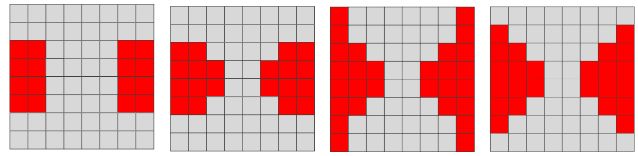
Figure 3: Example of OCRA routing algorithm in a 3D NoC.

have been done in this paper for analyzing border regions traffic using West-first and MPTM routing algorithms as oblivious adaptive methods, and the TTABR routing algorithm as traffic-aware adaptive methods, under different traffic patterns. Results show that while border routers shown in Fig. 1 are about 43% of the total routers in the network, the traffic load of this area using different adaptive routing algorithms under different traffic patterns is less than 33% of the total traffic for oblivious turn-restricted routing algorithms (i.e. West-first and MPTM) and less than 38% of the total traffic for traffic-aware routing algorithms (i.e. TTABR). The results prove that the traffic in the border regions of the network is less than in the central regions and could be more balanced. To address these limitations, the proposed routing algorithm utilizes a location-based analysis to determine inherently less congested regions of the 3D NoC offline and then route the packets through these regions. This routing algorithm balances the traffic load over the 3D NoC passively and imposes negligible additional complexity on the routers. To the best of our knowledge, this routing is the first location-based passive load-balancing 3D NoC routing algorithm.

3.2 An Oblivious Congested Region Avoiding Routing Algorithm (OCRA)

Using different adaptive routing algorithms, packets flowing from east and west to the north and south and top and bottom pass through the central region of a 3D NoC. Therefore, routing packets from north and south to top and bottom regions impose additional traffic load at the central region of the network while there are many less congested routers at the borders and corners of the network. The objective of OCRA is to efficiently route network traffic by utilizing less congested areas of the network. As a result, certain packets that can be statically routed through inherently less congested paths are directed accordingly.

The proposed routing algorithm classifies a 3D NoC into three distinct regions: 1) the X-border region: borders in the X dimension, 2) the Y-border region: borders in the Y dimension, and 3) the central region. Fig. 2a shows these three distinct regions. Based on the motivational results presented in the previous section, OCRA tends to utilize the XZY routing algorithm for X-border regions, the YZX



(a) (b) (c) (d)

Figure 4: Top-down view of different shapes of Y-border regions in a layer of NoC. YZX/YXZ routers are shown in red. Only the Y-Border regions are shown for simplicity.

Algorithm 1 OCRA routing algorithm

```
function OCRARouting (sourceNode, currentNode, destinationNode)
if sourceNode is in the X-border region then
    if destinationNode.z > sourceNode.z then
        XZY(currentNode, destinationNode)
    else
        XYZ(currentNode, destinationNode)
    end if
else if sourceNode is in the Y-border region then
    if destinationNode.z > sourceNode.z then
        YZX(currentNode, destinationNode)
    else
        YXZ(currentNode, destinationNode)
    end if
else
    XYZ(currentNode, destinationNode)
end if
```

routing algorithm for Y-border regions, and XYZ for central routers as shown in Fig. 2b. By doing so, the routers in the border of the source layer are utilized and traffic load generated at the border regions will not enter the central congested regions. OCRA performs the move in the Z dimension before the Y dimension in the X-border regions and before the X dimension in the Y-border regions when routing a packet downward to utilize border routers in different layers. As discussed later, for the sake of deadlock freedom, the move in the Z direction is the last move when going upward. Fig. 3 shows an example of OCRA routing algorithms. Router S1 located in the Y-border region of layer 1 tends to send a packet to router D1 in layer 3. Instead of using XYZ or other oblivious algorithms which pass through the central congested regions, OCRA first performs the movement in the Y dimension which is in the less congested area. Second, it performs the movement in the Z dimension which is also in the less congested regions of layer 2 and layer 3. Finally, it moves in the X dimension to reach the destination by the minimum entrance to the congested region of layer 3. Similarly, when router S2, which is located in the X-border region of layer 3, tends to send a packet to router D2 in layer 4, OCRA first routes the packet through the X dimension to avoid entering the central region of layer 3. Then, it routes the packet through the Z dimension to utilize the border routers of layer 4. At last, it moves in the Y dimension to deliver the packet to the destination. It is worth stating that unless there are some adaptive and congestion-aware routing algorithms that can detour congested areas, they all impose different overheads on the network.

The shortest number of nodes that a packet should traverse from the source to the destination is achieved through XYZ. However, opting for less congested paths results in reduced packet delay. OCRA utilizes the less congested regions of 3D NoC by employing a location-based static configuration of routers during the design phase. This approach eliminates the need for run-time decisions and extra links to transmit traffic and congestion information to

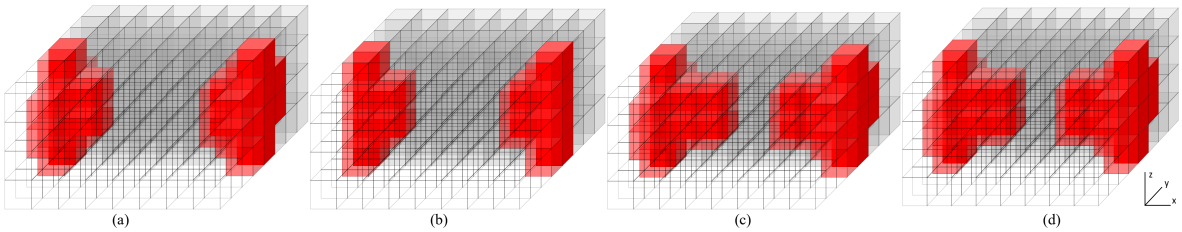


Figure 5: Different configurations of OCRA. YZX/YXZ routers are shown in red. Only the east and west regions are shown for simplicity.

adjacent nodes, which are typically required in adaptive routing algorithms.

The pseudo-code of the proposed OCRA routing algorithm is illustrated in Alg. 1, which has constant time complexity. At the beginning of the algorithm, the X-border, Y-border, and central regions are defined. In the following lines, the packets are routed based on their source router location. Traffic originating from the Y-border regions is initially routed in the Y dimension. Consequently, packets generated in these nodes with destinations at the top and bottom are first directed to the border nodes, which experience lower congestion. Subsequently, they are routed in the Z direction and then in the X direction. This approach ensures that congested nodes at the center of all layers along the route to the destination are avoided, resulting in a less congested possible path.

The size and shape of the three distinct regions defined in OCRA have a significant impact on traffic balancing within the network. When considering network size and traffic, various shapes and sizes should be taken into account for these regions in design time. The shape of the X-border and the Y-border regions of the layers can be specified differently as shown in Fig. 4. Experimental results showed that considering border regions as Fig. 4a may result in congestion on the inner side of the border regions. Among different border region configurations, the pyramidical configuration, which is shown in Fig. 4d, had better traffic load distribution through our experiments. Fig. 5 illustrates four different pyramidical configurations of OCRA in three-dimensional NoC. Fig. 5a and Fig. 5b have smaller east and west regions which prevent increasing the traffic load inside these regions. Fig. 5c and Fig. 5d show larger east and west regions, which prevent the generation of traffic at the center of the NoC. The traffic inside these regions may increase as well.

3.3 Deadlock Freedom

Deadlock-freeness is a critical requirement for routing algorithms in 3D NoCs. Extensive research has shown that certain oblivious routing algorithms, such as XYZ or YXZ, guarantee deadlock freedom. However, caution must be exercised when combining different oblivious routings, as it can potentially introduce the risk of deadlock in certain scenarios.

Deadlock in OCRA halts the injection of packets into the network. Deadlock occurs when there is a cycle dependency within the network, leading to a situation where all packets within the cycle are waiting for an event that never occurs because each packet has preempted a resource that one or more packets are waiting for.

There are three different approaches for handling deadlock in NoCs. The first approach, which is widely used in partially adaptive routing algorithms, restricts some specific turn models to prevent deadlock. Odd-Even, RTM, and MPTM routing algorithms use this approach to guarantee deadlock-freedom. The second approach is using a number of virtual channels to break the cyclic dependency and remove deadlock. This approach definitely imposes hardware and power overheads depending on the number of virtual channels (VCs) employed. The last approach is to detect and recover deadlock when it happens. Some recent works [1, 13, 18] propose a number of deadlock detection and recovery methods that accurately detect deadlocks and remove them with a cost-efficient deadlock recovery technique. Although this approach imposes hardware and power overheads on the routers, the experiments we have done showed that the overhead of this approach is smaller than adding VCs to the physical channels in the OCRA routing algorithm. Therefore, we use the deadlock detection and recovery method described in [18] to handle the deadlock issue in the OCRA routing algorithm. As it is stated in [18], the overhead of this method is 7% additional area and 3.5% extra power consumption. To further decrease the deadlock recovery overhead, we use a turn-restricted model to avoid deadlock in the vertical direction. To this end, all turns after the upward movement are forbidden to avoid cyclic dependency in the vertical direction.

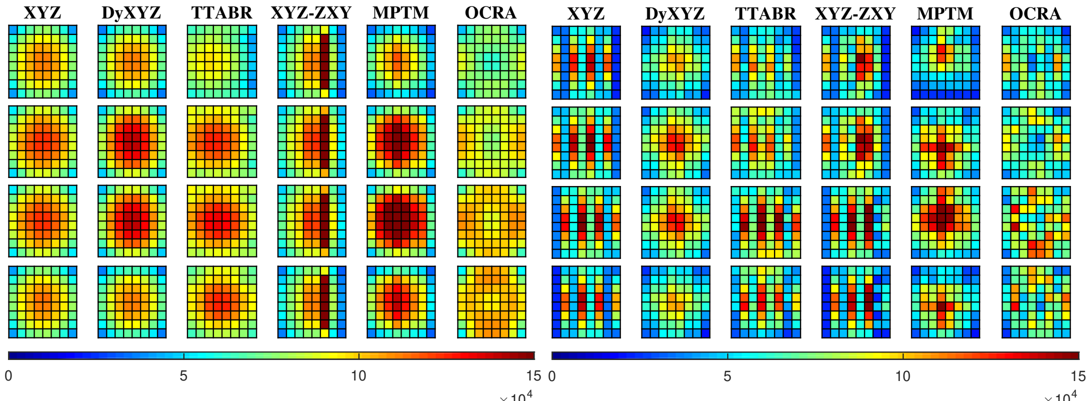
4 EXPERIMENTAL SETUP

To evaluate the OCRA routing algorithm and compare its traffic and performance behavior with the state-of-the-art schemes, we utilize the Access Noxim simulator, which is a cycle-accurate traffic thermal co-simulation platform for 3D NoC. The experiments are conducted on a 3D NoC configured based on the Intel 80-core chip. The network follows an 8×8×4 3D mesh topology. Each buffer's channel depth is set to 8 packets, and the packet length is randomly selected from 2 to 10 flits. The wormhole technique is employed for flow control, and random arbitration is used as the switching algorithm. The results are collected after a warm-up period of 10,000 cycles, with an injection rate of 0.02 packet-per-cycle during 100,000 cycle simulations.

The OCRA routing algorithm is compared with the state-of-the-art routing algorithms mentioned in Table 1. The state-of-the-art methods have been selected from different categories so that the comparison can be made with a wide range of methods. These algorithms are evaluated under Random and Shuffle synthetic traffic patterns. In addition to synthetic traffics a set of realistic traffics from Parboil [16] benchmark suit is considered. The communication for Parboil benchmarks is obtained by simulating on a 64-core manycore system.

4.1 Statistical Traffic Load Distribution (STLD) Analysis

Traffic distribution over the network is presented in Fig. 6 using XYZ, DyXYZ, TTABR, XYZ-ZXY, MPTM, and OCRA by applying Random and Shuffle synthetic traffic patterns.



(a) Random

(b) Shuffle Traffic

Figure 6: Traffic load distribution in different layers of a 3D NoC using different routing algorithms.

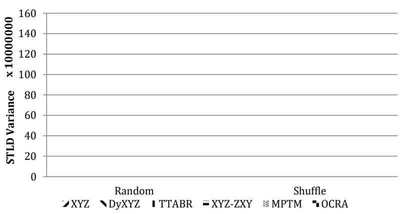


Figure 7: Statistical Traffic Load Distribution (STLD) variance of different routing algorithms.

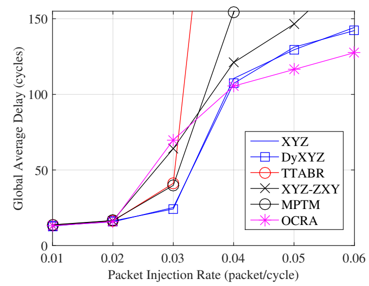


Figure 8: Average delays across various packet injection rates for different routing algorithms.

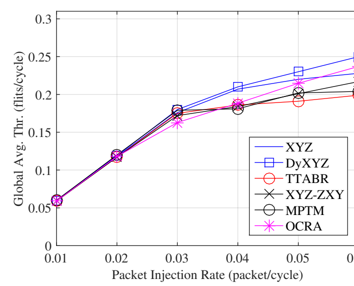


Figure 9: Average throughput of different routing algorithms by increasing packet injection rate.

Table 1: Evaluated previously proposed routing algorithms.

	Deterministic	Adaptive	
Algorithm		Oblivious	Cong. Aware
XYZ	✓		
DyXYZ			✓
TTABR			✓
XYZ-ZXY		✓	
MPTM		✓	

In Uniform Random traffic (Fig. 6a), XYZ-ZXY shows the worst traffic distribution over the network. The traffic of the network using routing algorithms of XYZ and TTABR are congested at the center. XYZ routing algorithm congests two central layers (Layer 2 and Layer 3), while TTABR congests Layer 4 more than XYZ because this routing algorithm tries to route the packets from the lower layers of the network for cooling issues. OCRA shows the better distribution of the packets around the network in Random traffic compared with the XYZ, TTABR, and XYZ-ZXY, respectively. The same trend can be observed in the Shuffle traffic pattern shown in Fig. 6b.

To better compare the traffic load distribution of different routing algorithms, we calculate the STLD variance of the different routing algorithms under different traffic patterns. Fig. 7 shows the comparison of the STLD Variance of the XYZ, DyXYZ, TTABR, XYZ-ZXY, MPTM, and the proposed OCRA under Uniform Randomand Shuffle traffic patterns. As shown in the figure, the OCRA has at least 42.08%, 39.58%, and 41.78% less traffic load variance among the other routing algorithms under Random and Shuffle, respectively. Because of using border and corner routers in OCRA more effectively for routing packets, OCRA can distribute traffic load through lateral routers more evenly. It also distributes traffic between layers because of routing packets in the Z dimension through border routers.

4.2 Performance and Throughput Analysis

One of the key advantages of NoC routing algorithms is their ability to efficiently route network traffic, even during periods of high load. Fig. 8 illustrates the average delay experienced by packets in the network as the number of injected packets by network nodes increases. The figure reveals that, at a packet injection rate of 0.03 packets-per-cycle, all of the routing algorithms experience a significant increase in global average delay. However, OCRA stands out by maintaining a lower global average latency compared to previously proposed routing techniques.

In particular, among these techniques, TTABR demonstrates a rapid growth in average packet delay at the packet injection rate of 0.03. Conversely, OCRA exhibits an average packet delay that is 2.44% and 30.49% lower than that of the XYZ and XYZ-ZXY routing algorithms, respectively. These findings indicate that OCRA achieves superior load balancing across the network, enabling more efficient utilization of network resources.

Fig. 9 illustrates the average throughput of various routing algorithms as the packet injection rate increases. When the traffic has a lower packet injection rate than 0.02 packets per cycle, all routing algorithms exhibit nearly identical average throughput. However, as the packet injection rate rises, the global average throughput of TTABR and XYZ-ZXY saturates at approximately 0.20 flits-per-cycle and 0.21 flits-per-cycle on average, respectively. On the other

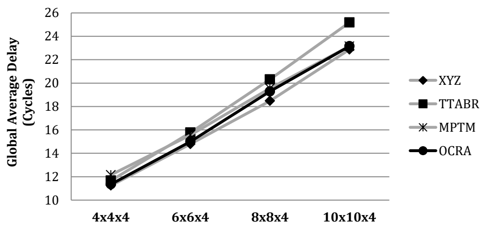


Figure 10: Global average delay of OCRA routing algorithm under different network sizes

Figure 11: Global average delay of OCRA routing algorithm compared with XYZ and TTABR under real-world Parboil traces.

hand, XYZ achieves a saturation value of 0.23. OCRA's global average throughput reaches 0.27 flits-per-cycle at a packet injection rate of 0.06 packets-per-cycle and continues to increase with further increases in the packet injection rate. This behavior is attributed to a more even distribution of packets throughout the network, resulting in improved performance.

To evaluate the performance and throughput of the network under different real traffic patterns, the Parboil benchmark suite has been run on gem5 simulator considering 64 cores. Results from these simulations can be seen in Fig 11. OCRA outperforms TTABR in all but one, and has less average delay compared with XYZ every trace except bfs and mri-q. OCRA exhibits less global average delay due to the reduction in congestion in the center of the network. If traffic characteristics are known during the network design, OCRA's regions can be configured to further accommodate the traffic profile and provide better traffic load balancing, which would result in reduced average delays.

To investigate the effect of network size on performance scalability, we measure the average delay of each routing algorithm under Uniform Random traffic pattern, when the network size increases. which leads to saturation throughput. The result of this experiment is shown in Fig. 8. As the results show, the global average delay of the OCRA routing algorithm scales with network size. It means that the OCRA routing is appropriate for both small and large-scale mesh-based 3D NoCs. As expected, the XYZ routing algorithm has the least average delay in different network sizes among the evaluated routing algorithms.

5 CONCLUSIONS

Three-dimensional Networks on Chips (3D NoCs) offer a promising solution for achieving low latency and high bandwidth communication among multi-core processor components. However, conventional 3D NoCs employing direction-ordered routing algorithms often encounter congestion issues, particularly at the center of the network. This congestion arises because routers located at the network's sides route their packets through the congested area, leading to sub-optimal performance. To address this concern, our work introduces a novel routing algorithm known as Oblivious Congested Region Avoiding (OCRA). The OCRA algorithm minimally modifies the existing network infrastructure while effectively distributing traffic by approximately 51.67% on average. By implementing this routing approach, we observe a significant reduction of around 48.7% in average queuing latency.

REFERENCES

[1] Akram Ben Ahmed, Achraf Ben Ahmed, and Abderazek Ben Abdallah. 2013. Deadlock-recovery support for fault-tolerant routing algorithms in 3d-noc architectures. In *2013 IEEE 7th International Symposium on Embedded Multicore Socs.* IEEE, 67–72.

[2] Yuan Cai, Dong Xiang, and Xiang Ji. 2020. Deadlock-free adaptive 3D network-on-chips routing algorithm with repetitive turn concept. *IET Communications* 14, 11 (2020), 1783–1792.

[3] Chih-Hao Chao, Kun-Chih Chen, Tsu-Chu Yin, Shu-Yen Lin, and An-Yeu Wu. 2013. Transport-Layer-Assisted Routing for Runtime Thermal Management of 3D NoC Systems. *ACM Transactions on Embedded Computing Systems (TECS)* 13, 1 (2013), 11–22.

[4] Kun-Chih Chen, Che-Chuan Kuo, Hui-Shun Hung, and An-Yeu Wu. 2013. Traffic- and Thermal-Aware Adaptive Beltway Routing for Three Dimensional Network-on-Chip Systems. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS).* 1660–1663.

[5] Kun-Chih Chen, Shu-Yen Lin, Hui-Shun Hung, and An-Yeu Wu. 2012. Traffic-Balanced Topology-Aware Multiple Routing Adjustment for Throttled 3D NoC Systems. In *Proceedings of the IEEE Workshop on Signal Processing Systems (SiPS).* 120–124.

[6] Ge-Ming Chiu. 2000. The odd-even turn model for adaptive routing. *IEEE Transactions on Parallel and Distributed Systems* 11, 7 (2000), 729–738. https://doi.org/10.1109/71.877831

[7] Ranjita Dash, Amartya Majumdar, Vinod Pangracious, Ashok Kumar Turuk, and José L. Risco-Martín. 2018. ATAR: An Adaptive Thermal-Aware Routing Algorithm for 3-D Network-on-Chip Systems. *IEEE Transactions on Components, Packaging and Manufacturing Technology (TCPMT)* 8, 12 (2018), 1–8.

[8] Masoumeh Ebrahimi, Xin Chang, Masoud Daneshtalab, Juha Plosila, Pasi Liljeberg, and Hannu Tenhunen. 2013. DyXYZ: Fully adaptive routing algorithm for 3D NoCs. In *2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing.* IEEE, 499–503.

[9] Kai-Yuan Jheng, Chih-Hao Chao, Hao-Yu Wang, and An-Yeui Wu. 2010. Traffic-Thermal Mutual-Coupling Co-Simulation Platform for Three-Dimensional Network-on-Chip. In *Proceedings of the IEEE International Symposium on VLSI Design, Automation, and Test (VLSI-DAT).* 135–13.

[10] Sheng-Chun Kao, Chao-Han Huck Yang, Pin-Yu Chen, Xiaoli Ma, and Tushar Krishna. 2019. Reinforcement learning based interconnection routing for adaptive traffic optimization. In *Proceedings of the 13th IEEE/ACM International Symposium on Networks-on-Chip.* 1–2.

[11] Shu-Yen Lin, Tzu-Chu Yin, Hao-Yu Wang, and An-Yeu Wu. 2011. Traffic-and Thermal-Aware Routing for Throttled Three-Dimensional Network-on-Chip Systems. In *Proceedings of the IEEE International Symposium on VLSI Design, Automation and Test (VLSI-DAT).* 1–4.

[12] Asit K Mishra, Narayanan Vijaykrishnan, and Chita R Das. 2011. A case for heterogeneous on-chip interconnects for CMPs. *ACM SIGARCH Computer Architecture News* 39, 3 (2011), 389–400.

[13] Nezam Rohbani, Zahra Shirmohammadi, Maryam Zare, and Seyyed-Ghassem Miremadi. 2017. LAXY: A Location-Based Aging-Resilient Xy-Yx Routing Algorithm for Network on Chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)* 36, 10 (2017), 1725–1738.

[14] Ronak Salamat, Misagh Khayambashi, Masoumeh Ebrahimi, and Nader Bagherzadeh. 2016. A Resilient Routing Algorithm with Formal Reliability Analysis for Partially Connected 3D-NoCs. *IEEE Transactions on Computers (TC)* 65, 11 (2016), 3265–3279.

[15] Narges Shahabinejad and Hakem Beitollahi. 2020. Q-thermal: A Q-learning-based thermal-aware routing algorithm for 3-D network on-chips. *IEEE Transactions on Components, Packaging and Manufacturing Technology* 10, 9 (2020), 1482–1490.

[16] John A Stratton, Christopher Rodrigues, I-Jui Sung, Nady Obeid, Li-Wen Chang, Nasser Anssari, Geng Daniel Liu, and Wen-mei W Hwu. 2012. Parboil: A revised benchmark suite for scientific and commercial throughput computing. *Center for Reliable and High-Performance Computing* 127 (2012), 27.

[17] Minghua Tang, Xiaola Lin, and Maurizio Palesi. 2016. The repetitive turn model for adaptive routing. *IEEE Trans. Comput.* 66, 1 (2016), 138–146.

[18] Yibo Wu, Liang Wang, Xiaohang Wang, Jie Han, Shouyi Yin, Shaojun Wei, and Leibo Liu. 2020. A deflection-based deadlock recovery framework to achieve high throughput for faulty nocs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 40, 10 (2020), 2170–2183.