Design of Transport Layer Assisted Routing for Thermal-Aware 3D Network-on-Chip

Tzu-Chu Yin, Chih-Hao Chao, Shu-Yen Lin, and An-Yeu (Andy) Wu
Graduate Institute of Electronics Engineering, National Taiwan University, Taipei
Email: {agmatthewoo, chihhao, linyan, andywu}@access.ee.ntu.edu.tw

***Abstract*—The thermal challenge of 3D Network-on-Chip (NoC) is more severe than 2D NoC. To ensure thermal safety and avoid huge temperature-limited performance back-off, run-time thermal management (RTM) is required. In RTM, the regulation of temperature requires throttling of the near-overheated router, which makes the topology become Non-Stationary Irregular Mesh (NSI-mesh). To successfully deliver packet in NSI-mesh, we propose the Transport Layer Assisted Routing (TLAR) framework, the Downward-Lateral Deterministic Routing (DLDR) algorithm, and the corresponding architecture. Based on the results of experiment and implementation, the proposed routing scheme can improve throughput by 70%, and the hardware overhead is only 11.1%.**

I. Introduction

With the advances of the semiconductor technology, the complexity and delay of interconnection increasingly dominate the performance of System-on-Chip (SoC). To provide more efficient interconnections and accommodate data transfer requirements, Network-on-Chip (NoC) has been viewed as a novel and practical solution. With the emerging three-dimensional (3D) IC technology, the interconnect delay can be reduced by providing shorter vertical connections [1][6]. However, thermal issues are the main challenges of 3D ICs and need to be considered for 3D NoC designs. Both the average length of heat conduction path and the power density projected on the heat sink increase. Besides, it has been shown that stacked routers generate thermal hotspots due to their higher switching activity and non-ignorable chip area [4][5].

To keep temperature below thermal limit in online operation, run-time thermal management (RTM) is required. Shang *et al*. [2] proposed *ThermalHerd*, a distributive and collaborative RTM scheme, to solve the thermal problem of 2D NoC. In our previous work [5], we proposed *Thermal-Aware Vertical Throttling-based RTM* (TAVT-RTM) for 3D NoC, improving the cooling speed and availability. In [2][5], the near-overheated routers are throttled in online operation, which change network topology to the Non-Stationary Irregular mesh (NSI-mesh), as shown in Fig. 1. The main problem of NSI-mesh is that the performance of downward routing, the reactive routing algorithm when there are throttled routers in the network, is insufficient. Therefore, the network throughput is very small.

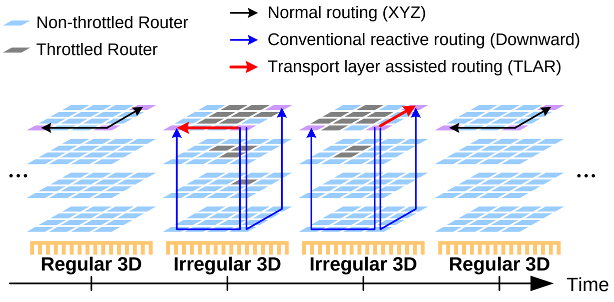

Fig. 1: Routing in the NSI-mesh of the thermal-aware 3D NoC.

For successful delivery and performance consideration, we proposed the framework of Transport Layer Assisted Routing (TLAR). The path diversity is increased, and the lateral-first path is selected with priority, as shown by the red line in Fig. 1. For TLAR, we propose a low latency and low cost architecture. Keeping low latency and reducing the memory overhead of TLAR is our major consideration in implementation. The contributions of this paper are:

- **Proposing Transport Layer Assisted Routing (TLAR)**: a routing framework guarantees successful data delivery in NSI-mesh where TAVT-RTM is adopted.
- **Designing the architecture of TLAR**: the architecture including the baseline network interface (NI) and the router for NSI-mesh, and the required modules for supporting TLAR.
- **Providing the cost reduction techniques for NSI-mesh network interface:** the techniques reduces the topology table from XYZ bits to $XY\log_2 Z$ bits; the routing mode memory from XYZ bits to XY bits.

The experimental result shows the proposed TLAR scheme can improve the throughput by 70%, and the implementation overhead is around 8.4%.

The rest of paper is organized as the following. In Section II, we present the proposed TLAR framework. In Section III, we propose the architecture of TLAR and the memory reduction techniques. In Section IV, we show the experiments to support our claim and discuss area overhead of TLAR. Finally, we conclude this paper in Section V.

II. Transport Layer Assisted Routing (TLAR)

A. Analysis of Fail Delivery in NSI-Mesh

To ensure the success of data delivery in a NSI-mesh network, all the following four cases should be avoided. The first one is shown in Fig. 2(a). The source router is fully throttled. The second one is shown in Fig. 2(b) the destination router is fully throttled. The third case is shown in Fig. 2(c). Any one of the router on the routing path is fully throttled. The last one is shown in Fig. 2(d). The channels on the routing path are blocked by other blocked packets from Fig. 2(a)-(c). If the source router is fully throttled, the packetized message will be blocked in the network interface. If any case of Fig. 2(b)-(d) occurs, the injected packets will be blocked and form a congestion-tree, stalling the network.

To eliminate the two cases of Fig. 2(a) and Fig. 2(b), the throttling information of the entire network is required in the transport layer. The case of Fig. 2(d) is a typical flow control problem, and the probability of occurrence can be reduced by adopting virtual channel flow control or output buffering router architectures [10]. The difficulty is that the case of Fig. 2(c) may occur on any router of the entire routing path and cannot be avoided in traditional routing algorithm. The reason is that usually the router only has the information of its neighbor routers. A typical example is shown in Fig. 2(c). The packet moves to the neighbor router of the throttled router and then is blocked. Briefly, the case of Fig. 2(c) cannot be avoided only with the information in network layer.

B. Transport Layer Assisted Routing (*TLAR*) *Framework*

To completely remove the case of Fig. 2, the information of the network layer and transport layer have to be jointly considered. We must guarantee that there is at least one non-fully throttled path toward destination router before we transfer the packet, and the packet is routed on the guaranteed path. According to [5], the bottom layer of the 3D NoC is never throttled and used as the guaranteed path for reactive routing, as shown in Fig. 1. Therefore, the channels in the bottom layer are very congested, and the throughput is very limited.

To relieve the heavy congestion in the bottom layer, we propose the Transport Layer Assisted Routing (TLAR) scheme. The main idea is to increase path diversity between the source and the destination. Besides, the extra path has to be guaranteed routable if it is selected as the routing path in the network layer. In TLAR framework, we apply downward-lateral deterministic routing (DLDR) in network layer, which is described in Section II.C in detail. There are two routing paths in TLAR-DLDR: lateral-first path and downward-first path. Due to the characteristic of the NSI-mesh, downward-first path is guaranteed routable.

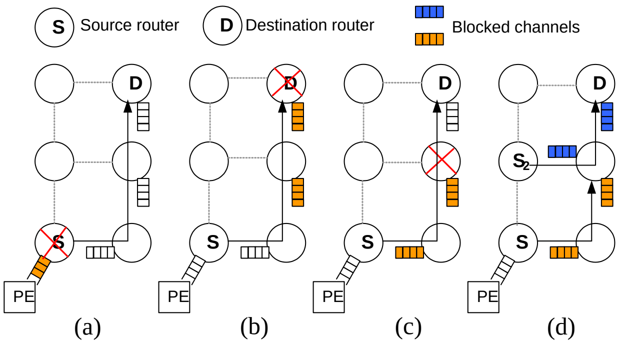

Fig. 2: Conditions of fail delivery.

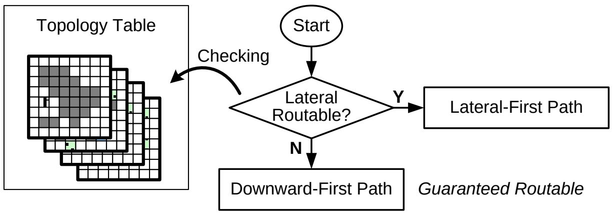

Fig. 3: Path selection in TLAR. The determination of lateral routability requires the topology table in transport layer.

The added lateral path in TLAR is lying in the source layer, the XY plane that the source router locates. The key idea of TLAR is that the throttling information in transport layer is used to assist the decision of layer for lateral routing and the selection of the corresponding routing algorithm. The result of path decision and the result of routing selection are combined as the routing mode, saved in packet header. When the packet is injected to the network layer, and the routers follow the mode to route.

Fig. 3 shows the flow chart of path selection in TLAR. The topology table stores the throttling information of all routers, which is necessary for the transport layer of the NSI-mesh network. The checking of lateral routability is done once for each destination as topology changing. For the lateral-first routable destination, the routing path is lateral-first in the source layer and then vertically routed. Otherwise, the path is downward-first, laterally routed in the bottom layer, and then vertically routed to the destination. The results of lateral routability for all destinations are saved in the routing mode memory for fast querying

C. Downward-Lateral-Deterministic Routing(*DLDR*) *in TLAR Framework*

The proposed 3D routing algorithm in TLAR is the combination of downward routing and a deterministic routing (DLDR). The downward routing is used for moving packets up and down in the vertical direction. The lateral deterministic routing is used for routing packets in the lateral direction. The path diversity is two because we can select to route in the source layer or the bottom layer. For reducing the computational complexity of checking rout ability, we adapt XY routing [10], a dimension-ordered routing (DOR), as the deterministic routing.

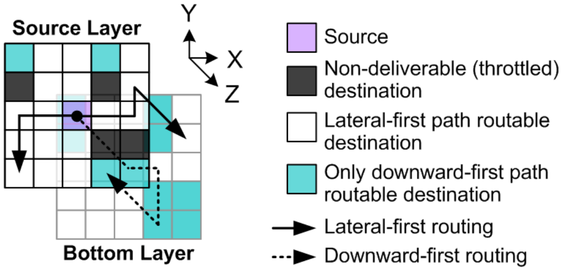
Source Layer
Y
X
Z
Bottom Layer
Source
Non-deliverable (throttled) destination
Lateral-first path routable destination
Only downward-first path routable destination
Lateral-first routing
Downward-first routing

Fig. 4: TLAR-DLDR routing examples

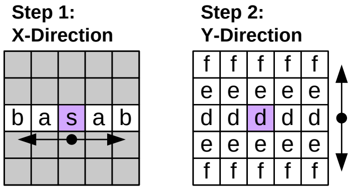

Fig. 5: Checking orders for determining whether the destination is lateral-first routable.

The examples of routability is shown in Fig. 4. There are three kinds of destination routers. First, the gray blocks are throttled destinations. The messages toward these destinations are kept in message queue until destinations are routable. Otherwise, the packets will be blocked in the network because the destination router is not active. Second, the white blocks are routable destinations with lateral-first XY routing; an example path is shown by the black line. Third, the aquamarine blocks are destinations those are only routable with downward-first downward routing. An example of the path of downward routing is shown by the black dotted line. Conclusively, if the path is lateral-first routable, the packet first traverses through the lateral path in the source layer. Then, the packet goes up or down to the destination router. Otherwise, the downward path is the only path, so the packet first traverses to the bottom layer and is routed laterally in the bottom layer. Then, the packet goes up to the destination router.

When topology changes, the routing modes have to be checked once again for each destination, and the decisions are saved in the network interface. The controller in transport layer checks if there is any fully-throttled router on the paths based on the topology table. The checking of the routability of all destinations in the source layer can be done by using the 2-step depth-first search (DFS) style, as shown in Fig. 5. The dependency is based on XY-routing, and the prerequisites that a node routable is its previous node also routable. The first step is checking routers along x-direction from source router s; the second step is checking routers along y-direction from the routers check in step 1: d. For a source sending packets to the entire X-by-Y-by-Z 3D mesh network , the checking can be done in XY cycles.

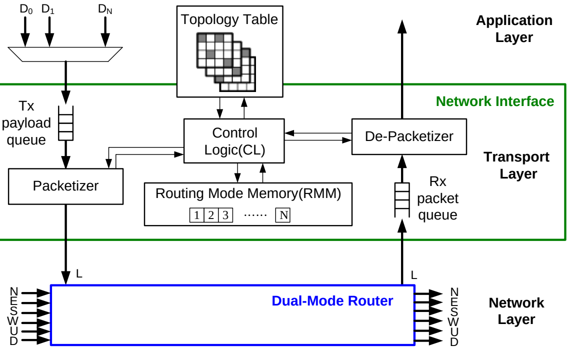

Fig. 6: Proposed architecture of TLAR

III. Architecture Design for TLAR

A. Transport Layer and Network Layer Architectures

In architecture design, we focus on transport layer and network layer, including network interfaces (NI) and router. NI connects the router and the IP module; router connects other routers and transfers packets hop by hop. As shown in Fig. 6, the NI of TLAR is composed of four parts:

- **Baseline Datapath and Tx/Rx Queues (Tx/Rx)**: Tx deals with the message from application layer and packetize the payloads in to packets to network layer. In contrast, Rx receives packet from network layer, de-packetizes, and combines to message to application layer. Tx and Rx require data queue for storing payloads and packets respectively.
- **Topology Table (TT)**: This table stores the 1-bit throttling information of each destination and is updated as topology change. TT is required for all NSI-mesh networks to solve the delivery problems in Fig. 2(a) and Fig. 2(b). Application layer and transport layer share the information in TT. Direct implementation of TT requires XYZ bits for an X-by-Y-by-Z 3D NoC.
- **Routing Mode Memory (RMM)**: RMM is required to reduce the timing overhead of checking routing mode for each packet. The mode for each destination is checked once as topology changing and stored in RMM. Before injecting a packet to network layer, the correspond routing mode is queried from RMM. Direct implementation of RMM also requires XYZ bits for an X-by-Y-by-Z 3D NoC.
- **Control Logic (CL)**: In baseline NI, CL controls the functionality of Tx/Rx. For TLAR network interface, CL also includes the TLAR routing mode checking, and controllers for reconfiguring the topology table. Finite-State Machines (FSMs) are to implement in CL for timing and signal controls.

The router in TLAR has to support both downward and deterministic routing. Therefore, a dual mode router is required, as shown in Fig. 9.

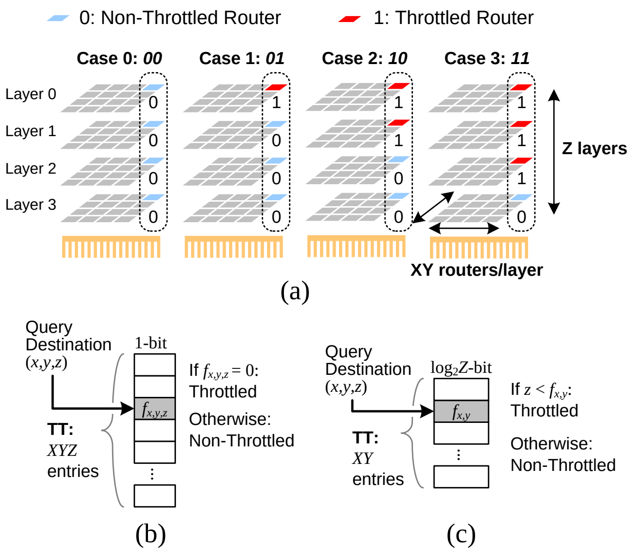

Fig. 7: (a) Reduce the size of topology table by storing the first non-throttled layer for each XY location. (b) Direct implementation. (c) Implementation with proposed TT reduction technique.

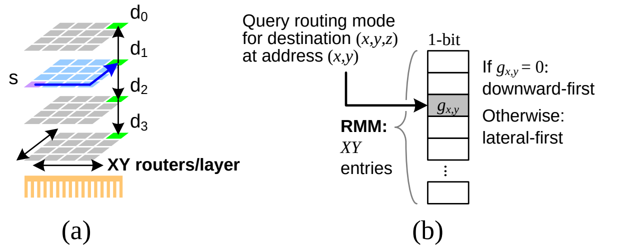
d0
d1
s
d2
d3
XY routers/layer
(a)
Query routing mode for destination (x,y,z) at address (x,y)
1-bit
g_{x,y}
RMM: XY entries
If g_{x,y} = 0: downward-first
Otherwise: lateral-first
(b)

Fig. 8: (a) The source destination pairs (s,d_0), (s,d_1), (s,d_2), (s,d_3) has the same source layer for lateral-first path, so their routing modes are identical. (b) For an X-Y-Z network, the size of RMM is XY bits.

B. Table and Memory Reduction Techniques for Network Interface (NI)

To prevent the cases in Fig. 2(a) and Fig. 2(b), CL queries TT for the payload of each message. If the source router or the destination is throttled, the payload will not be packetized. Otherwise, the payload is packetized and use the routing mode queried from RMM. The proposed memory reduction technique is based on the three characteristics of NSI-mesh of TAVT-RTM: 1) TAVT never throttles the router in the bottom layer; 2) if a router is throttled, all the routers above it are throttled; 3) if a router is not throttled, all the routers below it are not throttled.

For topology table (TT), if the throttling can be applied to all routers, 1-bit information is required for each destination in TT. Because of the throttling characteristics (2) and (3), we only need to store which layer is the top of the non-throttled routers, as the green nodes shown in Fig. 7(a). Therefore, the number of bits can be reduced from XYZ to $XY \log_2 Z$, as shown in Fig. 7(b) and Fig. 7(c).

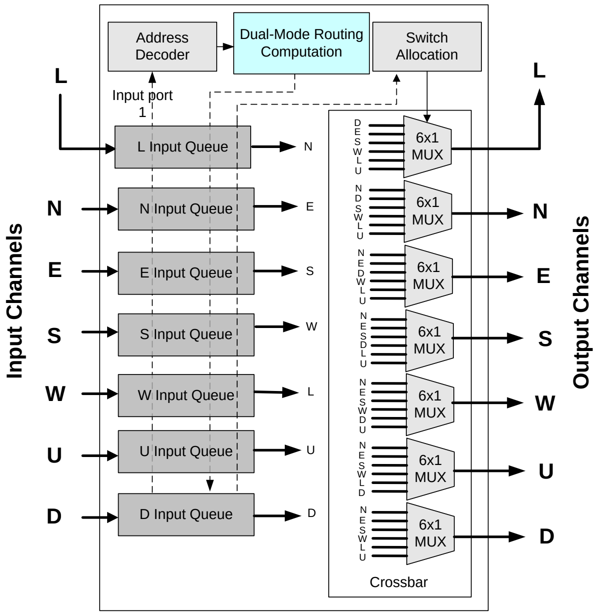

Fig. 9: Proposed architecture of 3D dual-mode router.

For routing mode memory (RMM), TLAR only requires XY bits to store the routing modes for all XYZ destinations. We use the example in Fig. 8(a) to illustrate the reason. Because all the source destination pairs (s,d_0), (s,d_1), (s,d_2), and (s,d_3) have the same source layer for the lateral-first path, their routing modes are identical. Therefore, CL can obtain the routing mode of the destination (x,y,z) by querying RMM for the entry at (x,y), as shown in Fig. 8(b).

C. DLDR Router Architecture

The proposed dual-mode router for DLDR is shown in Fig. 9, adopting wormhole flow control [10], which is broadly adopted in NoC routers for its low memory requirement. The router is consisted of five major functional modules; 1) routing computation logic (RC); 2) switch allocation logic (SA); 3) crossbar switch (CS); 4) input queues (IQs); and 5) inter-router physical channels (ICs and OCs). The router is two-stage pipelined, and further pipeline is achievable for higher performance. The traditional 2D NoC router has channels to connect north (N), east (E), south (S), west (W), and local (L) directions. The 3D NoC router requires extra two physical channels: up (U) and down (D) for vertical connections. Consequently, the size of CS increases from 5×5 to 7×7, and the number of IQs is increased from 5 to 7. In addition, the cost of routing logic and arbitration logics also increase due to the extra two channels. The main overhead of the proposed TLAR is RMM and the extra control logics in RC. To support the functionality of downward routing and XY routing, RC has to be dual-mode, as shown in Fig. 9. When the packet is processed, the 1-bit routing mode, the source address, and the destination address in the packet header are used as inputs of RC. If current XY address is equal to the source XY address, then RC determines the direction is lateral-first or downward-first based on the routing mode. If current address is equal to the destination address, RC indicates SA to transfer the packet to the local output channel to the transport layer of the destination router. Otherwise the routing is based on the description of Section II.C.

IV. Experiments and Implementations

A. Performance Evaluation

In this section we show the performance evaluation of the proposed TLAR algorithms. We developed a traffic-thermal mutual-coupling co-simulation platform [4]. The modeled 3D NoC system is composed of 8×8×4 tiles, and each tile consists of a 7-port router, a local memory, and a processor. The network model, power model, and floorplan are based on [9]. The traffic distribution are uniform random, and the length of a packet is randomly from 2 to 10 flits. Because we model the network architecture and of [9], the depth of each input queue is 16 flits. The link level flow control protocol is full handshake request-ack.

For detailed performance analysis, we use the case of two 2x2x3 throttled regions, which are the white regions in Fig. 10(a) and Fig. 10(b). The Statistical Traffic Load Distribution (STLD) [9] is used to show the loading of the network. The number on the side of the colorbar represents the number of passed flits during a period of time in the steady state. As shown in Fig. 10(a) and Fig. 10(b), though there is only two 2x2x3 pillars throttled in the upper three layers, many packets have to be routed downward through the bottom layer. The congestion degree of the conventional downward routing in the bottom layer is larger because there are more packets traversed in the bottom layer. In Fig. 10(b), TLAR routes more packets laterally in the source layer, so the network is more balanced vertically. The STLD shows the number of flits in the bottom layer is reduced around 35%. Because of the more balanced loading of the network, the TLAR-DLDR has better performance than the baseline reactive downward routing algorithm. Fig. 11 shows the average latency versus network injection rate. The throughput of the entire NSI-mesh network is improved from 13.5 flits/cycle to 23flits/cycle, which is improved around 70%.

B. Implementation Results

In this section, we show the implementation results of the proposed TLAR, including the NI and the router. Table 1 shows the design parameters for implementation and synthesis. The NI and the router are designed for a 8x8x4 3D NoC. The number of ports per router is increased from 5 to 7. The input queue depth is set to 16 according to [9]. The TLAR NI contains a 16-flit queue for Tx payloads from application layer and a 16-flit queue for Rx packets from network layer. With TSMC 130nm technology, the post synthesis simulation shows the NI and the router is able to operate at 360MHz.

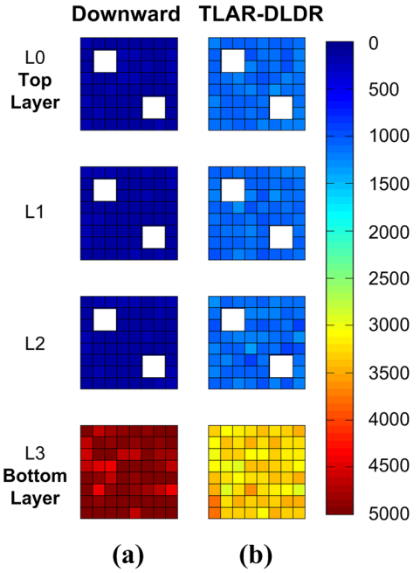

Fig. 10: (a) STLD of conventional design; (b) STLD of proposed TLAR framework with DLDR algorithm.

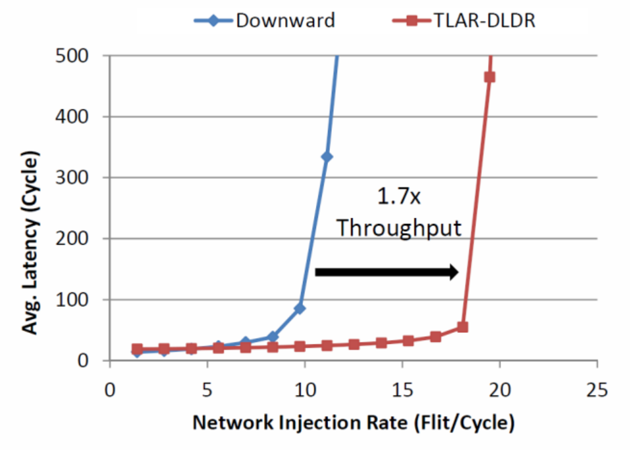

Fig. 11: latency vs. network injection rate.

First, we show effectiveness of the proposed table and memory reduction techniques by Fig. 12, which is the synthesis result of the TLAR NI. The area of direct implementation is 117,173μm^2. Applying the proposed TT reduction technique, the area of topology table is decreased from 31,988μm^2 to 15,487μm^2, and the total area of the NI is decreased to 101,698μm^2, which is reduced by 13.6% . Applying the proposed RMM reduction technique, the area of routing mode memory is decreased from 29,877μm^2 to 7,019μm^2. The combination of the proposed TT reduction and RMM reduction can reduce the total area of NI to 80,664μm^2, which is 68.5% of the direct implementation.

Table 1: Implementation parameters

Network Topology	8x8x4 3D Mesh
Number of ports per router	7
Input queue depth per channel	16 flits
Flit size	32 data bits + 2 control bits
Size of payload queue in NI:Tx	32 bits*16 flits = 512 bits
Size of packet queue in NI:Rx	34 bits *16 flits = 544 bits
Technology of Synthesis	TSMC 130nm
Clock period/frequency	2.8 ns/360 MHz

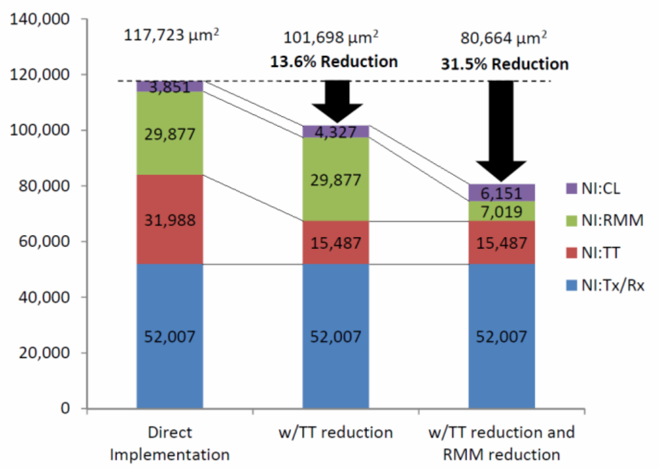

Fig. 12: Synthesis area (μm^2) of TLAR NI.

Table 2: Synthesis Area of NI and Router (μm^2)

	Traditional NI + Router, XYZ routing	**Direct Implementation of TLAR NI + Router**	**TLAR NI+ Router w/TT and RMM reduction**
NI:Tx/Rx	52,007	52,007	52,007
NI:TT	N/A	31,988	15,487
NI:RMM	N/A	29,877	7,019
NI:CL	1,937	3,851	6,151
NI	53,944	117,723	80,664
Router	191,059	191,577	191,577
Total	245,003	309,300	272,241
Overhead	N/A	26.2%	11.1%

Second, we compare the synthesis result of the traditional NI + router design and the synthesis result of the proposed TLAR NI + router design in Table 2. The traditional NI only contains Tx/Rx and CL, so the area is only 53,944μm^2. TT and RMM are not required because the traditional NI is designed for regular mesh topology. For the 3D NoC router, the difference is in routing computation. The TLAR router requires the dual-mode routing computation (downward routing and XY routing); the traditional router only computes for XY routing. As shown in Table 2, the computation overhead for downward routing is only 518μm^2 more than traditional router. The area of direct implementation of TLAR NI and router is 309,300μm^2. The area overhead of direct implementation is 26.2%. With proposed memory and table reduction techniques, the area is reduced to 272,241μm^2. The area overhead is reduced to 11.1 %.

V. Conclusion

In this paper, we introduce the delivery problem in NSI-mesh of thermal-aware 3D NoC. To avoid the cases of fail delivery, we propose the TLAR framework, DLDR algorithm, and the corresponding architecture. From our experiments, the proposed TLAR can effectively balance the vertical load distribution and improves the throughput to 1.7x. The area overhead of direct implementation of TLAR is 26.2%. With the proposed table and memory reduction techniques, the area overhead of TLAR can be reduced to 11.1%, which is relative small in comparison with the throughput improvement.

References

[1] V. Pavlidis and E. Friedman, “3-D Topologies for Networks-on-Chip,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 15, no. 10, pp. 1081-1090, Oct. 2007.

[2] L. Shang, L. Peh, A. Kumar, and N. K. Jha, “Thermal modeling, characterization and management of on-chip networks,” in *Proc. IEEE/ACM Int. Symp. Microarchitecture* (Micro’04), Dec. 2004, pp. 67-78.

[3] Noxim: network-on-chip simulator [Online]. Available: http://sourceforge.net/projects/noxim/

[4] 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 *Proc. IEEE Intl. Symp. VLSI Design, Automation, and Test* (VLSI-DAT’10), Apr. 2010, pp. 135-138.

[5] C. H. Chao, K. Y. Jheng, H. Y. Wang, J. C. Wu, and A. Y. Wu, “Traffic and thermal-aware run-time thermal management scheme for 3D NoC systems,” in *Proc. ACM/IEEE Int. Symp. Networks-on-Chip* (NOCS’10), May 2010, pp.223-230.

[6] B.S. Feero and P.P. Pande, “Networks-On-Chip in a Three Dimensional Environment: A Performance Evaluation,” *IEEE Trans. Comput.*, vol.58, no. 1, pp. 32-45, Jan. 2009.

[7] S.-Y. Lin, C.-H. Huang, C.-H. Chao, K.-H. Huang, and A.-Y. Wu,"Traffic-Balanced Routing Algorithm for Irregular Mesh-Based On-Chip Networks," *IEEE Trans. Comput.*, vol. 52, pp. 1156–1168, Sept. 2008

[8] L. Benini and G. De Micheli, “Networks on chip: a new SOC paradigm”, *IEEE Comput.*, vol. 35, pp.70-78, Jan. 2002.

[9] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar, "A 5-GHz mesh interconnect for a teraflops processor," *IEEE Micro*, vol. 27, no. 5, pp. 51-61, Oct. 2007.

[10] W. J. Dally, B. Towles, “Principles and Practices of Interconnection Networks,” Morgan Kaufmann Publishers, 2004.