國立臺灣大學電機資訊學院電子工程學研究所

碩士論文

Graduate Institute of Electronics Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

適用於溫度感知三維晶片內網路之傳輸層協助路由演算法和架構

Transport-layer Assisted Routing Algorithm and Architectures for Thermal-Aware 3D NoC

銀子駒

Yin Tzu-Chu

指導教授：吳安宇博士

Advisor: AN-YEU WU, Ph.D.

中華民國一百年六月

June, 2011

國立臺灣大學碩士學位論文
口試委員會審定書

適用於溫度感知三維晶片內網路之傳輸層協助路由演算法和架構

Transport-Layer Assisted Routing Algorithm and Architectures for Thermal-Aware 3D NoC

本論文係銀子駒君 (R98943140）在國立臺灣大學電子工程學研究所完成之碩士學位論文，於民國 100 年 6 月 29 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

吳安宇
（指導教授）

李進福　　呂學坤

李建模

系主任、所長　張耀文

Transport-Layer Assisted Routing Algorithm and Architectures for Thermal-Aware 3D NoC

By

Yin Tzu-Chu

THESIS

Submitted in partial fulfillment of the requirement
for the degree of Master of Science in Electronics Engineering
at National Taiwan University
Taipei, Taiwan, R.O.C.

June. 2011

Approved by：

An-Yeu Wu. J-Fu Li Shyue-Kung Lu

Chien-Mo Li

Advised by： An-Yeu Wu.

Approved by Director： Maowen Chang

誌謝

兩年的研究生活終告結束，此時我即將步出校門，走完最後一段校園路程。回想這段時光，滿是歡笑與汗水，以及成長的喜悅。從進入 Access 實驗室到今日的點點滴滴，都是我這一路上最珍貴的回憶。

感謝我的指導教授吳安宇老師，您教導我身為一個研究生的自覺，開啟我對研究題目的興趣，也在研究過程中不斷給予建議與激勵，讓我瞭解做研究的態度與方法。感謝我的口試委員：呂學坤老師、李進福老師和李建模老師，謝謝您們能蒞臨指導以及給予我建議和修正方向。

感謝實驗室每一位學長同學學弟妹。謝謝 Foster 在我剛進入實驗室時帶領我。謝謝小昊在我論文陷入瓶頸時給予指導。林彥、thinking、yagaru、阿ben、詹公、坤志跟恩瑞，你們總能在我報告陷入盲點時適時的糾正以及親切指導。謝謝賢楷、郁豪、紹維、朝陽，你們陪我一起撐過各個艱苦時光，以及一起享樂。謝謝輝舜、明翰、姿伶、冠宇、怡萱和寬育，能當你們學長是我的榮幸。謝謝玉霜等助理幫忙了許多事務。我永遠記得這兩年中，大家一起打牌、打電動、唱歌、看電影的時光。我永遠懷念大家一起出遊、聚餐、打球的歡樂。謝謝你們的陪伴，以及教導我的一切。希望你們都能順利畢業，都能有個璀璨的未來。

最後，謝謝爸爸，雖然你無法看到我順利完成碩士學歷，但你能參加到我的碩士畢業典禮，對我是莫大的安慰跟鼓勵。爸媽和弟弟，謝謝你們給予我最溫暖的親情和信任，有你們的陪伴是我人生中最大的福氣。祝媽媽跟弟弟能身體健康，快樂生活以及萬事如意。

謹以這篇論文，獻給每位家人朋友。

銀子駒　謹誌

於台大電子所 Access IC 實驗室

中華民國一百年七月二十五日

中文摘要

本文針對熱感知三維晶片內網路設計的效能降低問題做出演算法和架構設計。傳統上為了確保溫度上安全以及避免效能被溫度限制大幅縮減，我們需要及時溫度控制機制。若我們因為及時溫度控制機制，而針對快超過臨界溫度的路由器來啟動壓制以便降溫，卻會造成系統拓撲變成非穩態不規則狀網狀拓撲。為了在非穩態不規則狀網狀拓撲下讓封包能成功地傳送，我們為熱感知三維晶片網路下，提出一個傳輸層協助路由演算法。根據實驗結果，我們提出的演算法能有效地增加效能，以及使交通負擔量更加平衡。我們基於低成本實現技術，提出記憶體降低技術，只需多 11.1%得實現成本，可獲得 1.7x 的吞吐量提升。

Abstract

In this thesis, we proposed algorithm and architecture design for performance reduction in thermal-aware 3D network-on-chip (NoC). To ensure thermal safety and avoid huge performance back-off from the temperature constraint, run time thermal management is required. However 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 a Transport Layer Assisted Routing (TLAR) secheme for thermal-aware 3D NoC. Based on the experimental results, the proposed routing scheme can significantly improve the performance and balance traffic load. For low cost implementation, we also propose memory reduction techniques, and we gain 1.7x throughput improvement for only 11.1% area overhead.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Motivation and Goal

1.1.1 3D IC and 3D Network-on-Chip

With the advances of the semiconductor technology, it is possible to integrate a large number of Intellectual Properties (IPs) on a single chip. However, the complexity of on-chip communication increases rapidly in the advanced VLSI technology, which impacts the performance and design complexity of SoC dramatically [1]-[4].

Fig. 1-1 shows the trend of the on-chip interconnections [3]. As the increases of IPs, the complexity of on-chip interconnections increases rapidly. Traditional methods, such as point-to-point interconnection and shared bus, are not enough to handle this problem. In order to provide more efficient interconnections and accommodate communication requirements, Network-on-Chip (NoC) has been proposed to as a novel and practical solution [5].

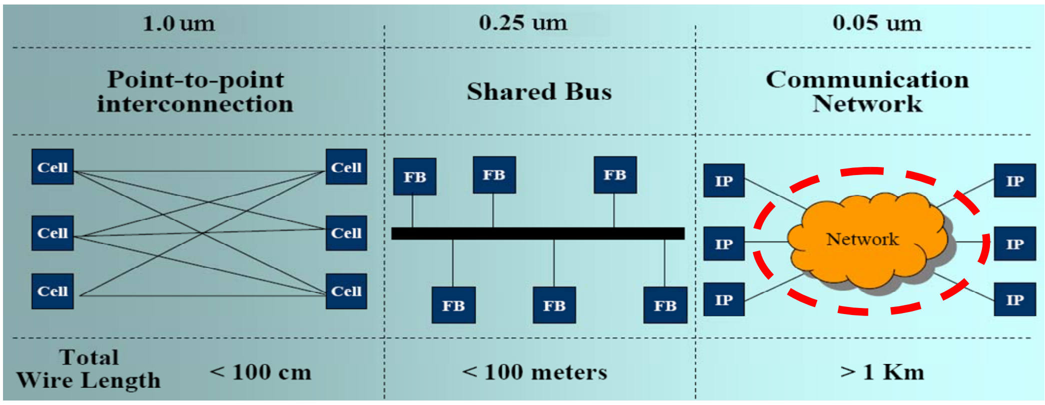
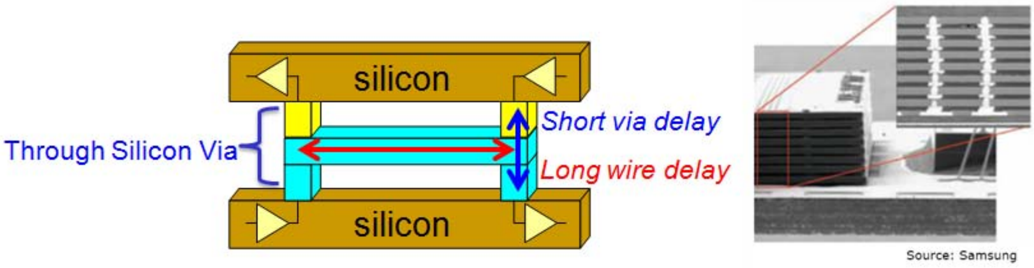


Fig. 1-1 On-chip communication trend [3].

Reduce system form factor by technology scale down in the planar IC becomes more difficult. Technology scaling has enabled an increase in integration density and a decrease in gate delay. However, higher integration densities require both a greater number of interconnects and longer interconnects. As the device delay reducing, performance is now dominated by the interconnect delay. In addition, power consumption and signal integrity have become more pronounced with technology scaling.

Three-dimensional (3D) integration is an effective design paradigm to manage the limitation of conventional two-dimensional IC [6]-[12]. 3D chip technology reduces interconnect delays by stacking multiple layer on top of each other, and by providing shorter vertical connections [13]. One of the most attractive technologies for 3D integration is based on Through Silicon Vias (TSVs), which cut across thinned silicon substrates to establish inter-die connectivity after die bonding [14], which is shown in Fig. 1-2. TSV based 3D IC has matured and motivated the combining of vertical interconnects with a NoC fabric, i.e., 3D NoC system.



Source: Samsung

Fig. 1-2 Structure of Through Silicon Via (TSV) [14].

Mesh-based NoC provides simple interconnection model for inter-IPs communication with the scalable and regular network architecture [2]. However, the intrinsic delay of IP and router cannot be reduced by network interconnection. Besides, the wire length between routers is dominated by the area of IP. The long wires lead to a considerable propagation delay. To solve these problems, 3D NoC is capable to provide several features:

- **Higher IPs mapping density in the network**: With vertical stacking, we can place more IPs in same area. The system form factor can be reduced. As shown in Fig. 1-3(a), for 2D NoC we only have one plant placing IP. However, we can stack more layers for 3D NOC, and the vertical distance is much smaller than horizontal distance, which we can ignore.
- **Lower connection cost and lower network power consumption:** The global net-length distribution scales in length as the square root of the number of strata [16]. For example as Fig. 1-3 (b), assume the number of strata is four, the net-length will be reduced by half. Besides, implementing the router hardware with 3D architecture can reduce the router area and inter-router wire length [17]. As the reduction of interconnection cost, the power consumption can also be reduced.
- **Higher network bandwidth**: The router of 3D NoC provides vertical dimension comparing to 2 D router, as shown in Fig. 1-3 (c). This increases the routing path diversity of routing and prevents congestion of network, and increases the network bandwidth. Besides, the vertical interconnection

has a negligible propagation delay since the short distance of TSV. This feature can speed up the vertical data transmission of 3D NoC.

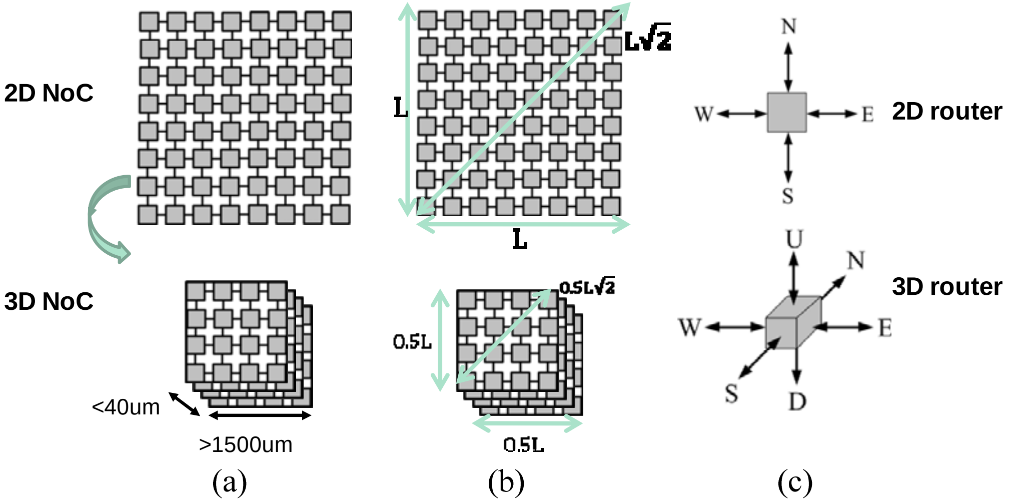


Fig. 1-3 (a) Topologies of 2-D and 3D NoC (b) Comparison of longest distance between 2D and 3D topologies (c) Comparison of path diversity between 2D and 3D topologies.

1.1.2 Thermal Issue for 3D Network-on-Chip

Thermal issues are the main challenges of 3D-IC. 3D integration results in power density increasing linearly in the number of vertically-stacked active layers. More-over, systems in the layers closer to the heat sink have higher cooling efficiencies than those farther from the heat sink [18], which is shown in Fig. 1-4. For NoC-based 2D-IC, power consumption of routers will increase system temperature and make the thermal issues be a large and growing concern [18]. Therefore, the temperature generated from NoC cannot be neglected. The networks consume a significant portion of the chip power budget [20][21], and hence having a substantial thermal impact. Therefore, understanding the joint thermal behavior of both processors and networks is the key to achieving efficient thermal design. There is an experimental result from [19], by studying the thermal impact of the on-chip networks in the MIT Raw CMP [22].

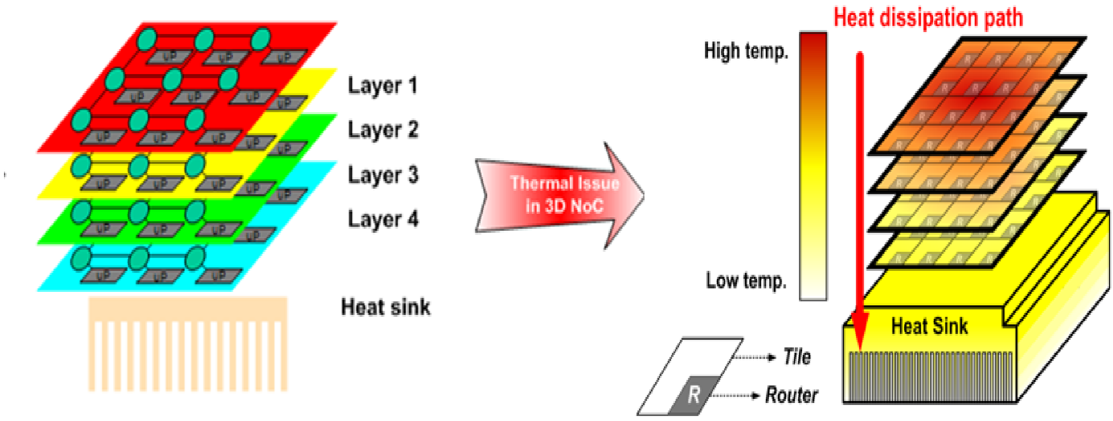


Fig. 1-4 Heat dissipation of 3D IC.

For 3D NoC, thermal problem becomes more serious [23][24], as shown in Fig. 1-4. Because the 3D NoC heat dissipation path is longer than 2D NoC, it needs long cooling time to cool the tile. Since many dies stacking vertically in 3D IC, there are more systems and communication networks over same area. This will make the power density increase dramatically. Besides, 3D IC has heterogeneous thermal characteristics. Systems in the layers closer to the heat sink have higher cooling efficiencies than those farther from the heat sink [25].

The straightforward way to deal with thermal issue is using better heat sink. However the cost of the heat sink grows exponentially with the power dissipation [20], as shown in Fig. 1-5. The more appropriate and cost effective method is using cheaper cooling device and applying thermal-aware design techniques [21]-[22].

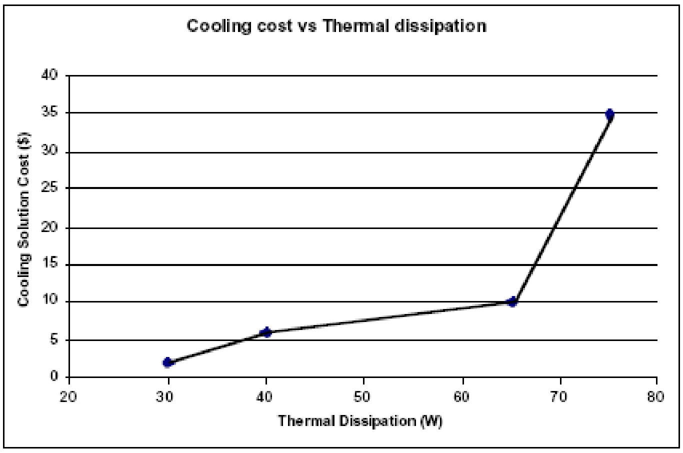


Fig. 1-5 The growth of cooling cost. [20]

1.1.3 Goal and Contribution

In this thesis, we focus on Transient-State Temperature impact, which will trigger run time thermal management (RTM) to block input packets to router for cooling down temperature. However, because RTM will throttled routers and let them not service, the performance will degrade. We want to reduce the impact to optimize performance, and don't let temperature increase above the thermal limit, as shown in Fig. 1-6.

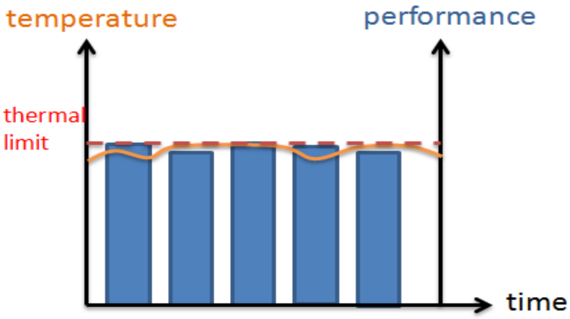


Fig. 1-6 Optimize performance under thermal limit.

Base on above design issues, we have some features to achieve:

- **Provide algorithm to maximum throughput for NSI-mesh:** Because the performance will degrade by using run-time thermal management (throttling), we should maximum performance under this constraint. 3D NoC suffers from thermal impact and we can’t increase thermal limit. We should provide algorithm to maximum performance under run-time thermal management.
- **Reduce area overhead:** Because we will provide algorithm to maximum performance under RTM, the cost will increase due to proposed algorithm. We should not increase cost too much, and maximum the performance effective.

Based on above design goal, we have to mitigate the performance degradation of thermal-aware designs for 3D NoC, and minimize the cost increase. A Transport Layer Assisted Routing (TLAR) scheme is proposed, and the architecture design is also proposed for estimating cost. Our contribution is listed as follow:

- **Transport-Later Assisted Routing Schemes for Vertical Throttling-Based Thermal-Aware 3D NoC**: Because of traffic congestion by throttling, we propose Transport Later Assisted Routing Schemes to improve network performance. For this algorithm, we need transport-layer to solve the problem. Before transmitting packet to network layer, we should collect some simple information about network traffic situation and throttling information. After receive the information, we can arrange some routing paths to destination router, and choose the suitable paths to detour throttled routers and make traffic load in network balance in order to make temperature more balance.

- **Architecture Design for Dual-Mode Router and Network Interference**

Because the laboratory focused on algorithm before, we need consider implement for the future research. So we can implement on three dimensional routers to consider cost and save area for future 3D router for simulation before three dimensional processes emerging.

1.2 Thesis Organization

The rest of this thesis is organized as follows: Chapter 2 introduces related works and introduces problems and solution of NSI-mesh. Chapter 3 presents proposed Transport-Layer Assisted Routing algorithm. Chapter 4 shows the experimental results of proposed techniques. Chapter 5 introduces proposed network interface and router architecture design, and compares area of synthesis results with proposed memory reduction techniques. Chapter 6, we conclude the work of this thesis and points out other potential future research direction.

Chapter 2
Related Works and Problem Description

The related and prior works about 3D thermal management and reactive routing for thermal-aware 3D NoC are introduced in this chapter. In order to utilize the interconnection advantage of 3D NoC, the related architecture designs are referred as our basic design. We evaluate the prior works of the thermal management and reactive routing to analyze the pros and cons of them. Finally, we discover problem and provide solution to solve these problem.

2.1 Run-Time Thermal Management for 3D NoC System

Since the thermal issue becomes more serious in 3D chips, *Chao et al.* proposed a thermal management for 3D NoC in [22], which is based on *ThermalHerd* [19]. The following techniques are proposed to control temperature:

Considering the heat flow in the 3D ICs, the author proposed a throttling scheme that shut down the routers along a vertical pillar, called the Vertical Throttling. Because we want to cool efficiently, we should throttle a vertical path to let heat conduct to heat sink, which is like a cooling channel. When throttling the routers, system performance greatly degrades. To prevent the performance degradation, the Vertical Throttling determinates different throttling ratio based on the temperature of routers, as shown in Fig. 2-1. The experimental results show that the Vertical

Throttling provides an efficient way to cool the NoC, comparing to the conventional distributed throttling [19].

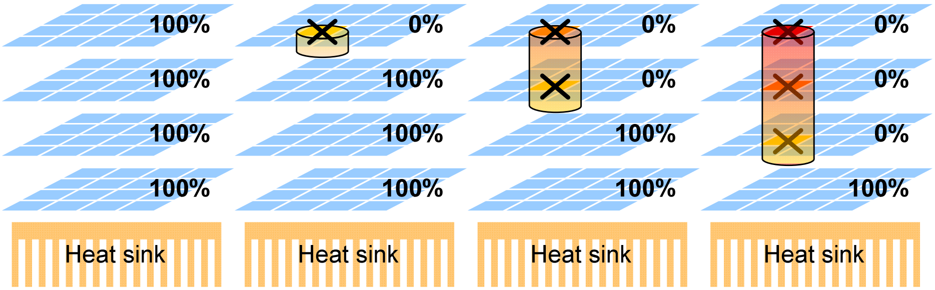


Fig. 2-1 Vertical throttling at different emergency level.[22]

2.2 Definition of Non-Stationary Irregular Mesh 3D NoC

Before describing the detail of problem of related work, we define a new term: Non-stationary irregular mesh (NSI-mesh). First, irregular mesh means that there are routers or links removed in the network, which means the topology is not regular. We define the time-varying irregular-mesh topology as a Non-Stationary Irregular Mesh (NSI-mesh). A mesh-based thermal-aware 3D NoC adopting vertical throttling can be categorized in NSI-mesh topology. In NSI-mesh, the distributed thermal sensors detect temperature and trigger throttling. Throttling will cause topology changing during the online operation. The topology changing in thermal-aware 3D NoC results in the problem of packet delivery. For faster cooling, the traffic quota of the near-overheat router has to be very small and even zero. When the router reaches the limit of quota, it has to fully block the input packets. This behavior results in that the topology becomes a Non-Stationary Irregular Mesh (NSI-mesh), as shown in Fig. 2-2.

Although we do not know which router will throttled or not, but we know the present topology change by 10^7 cycles, and is different form fault tolerant NoC, which change topology by cycle.

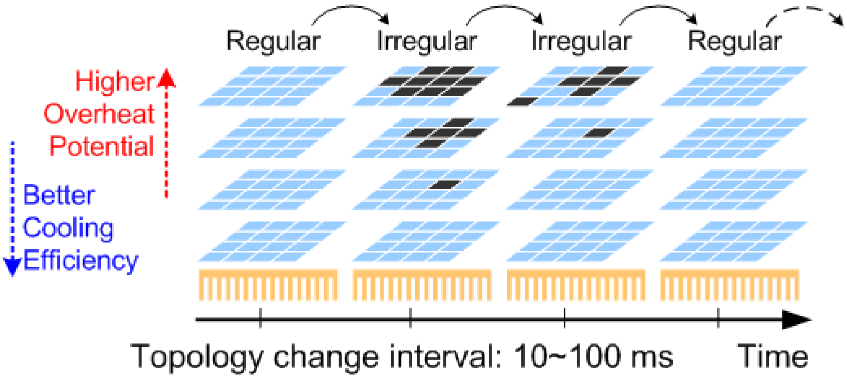


Fig. 2-2 An example of Non-Stationary Irregular Mesh (NSI-Mesh) network. The topology changes in online operation because of the throttling in RTM.

The key characteristics of the NSI-mesh of Thermal-Aware Vertical Throttling (TAVT) based 3D NoC are: (i) if a router is throttled, all the routers above it are throttled, and (ii) if a router is not throttled, all the routers below it are not throttled. As shown in Fig. 2-2. , the topology in offline stage and starting time of online stage is a traditional mesh. As temperature arises, near-overheat routers will be throttled, and the topology turns into irregular-mesh. Owing to the accuracy of temperature sensing, we can view that the network changes its topology at ms interval, which means 10 or 100 of millions cycles (10^7~10^8 cycles) if the NoC is operating at 1GHz.

2.3 Related Works of Reactive Routing for Thermal-Aware 3D NoC

To prevent the packet congested because of throttled tiles, we need reactive routing to detour throttled routers or prevent throttled paths. A downward routing algorithm [22] is proposed to migrate the horizontal routing to bottom layer. Besides, a traffic-aware downward level selection scheme is proposed to prevent network saturation. It compares the features of different downward levels. It also specifies the spatial thermal distribution in the 3D NoC system is non-uniform, while the traffic load is balanced. In order to balance the spatial thermal distribution, downward routing provides different downward levels for balancing thermal distribution, and the maximum network throughput improvement is shown under normal thermal limit. The reactive routing is simply the extension of Downward Routing. While the Vertical Throttling guarantees the layer close to the heat sink always available, the reactive routing detours packets from throttled routers by different downward levels. In the worse case, all the packets are transported at the layer close to the heat sink, as shown in Fig. 2-3.

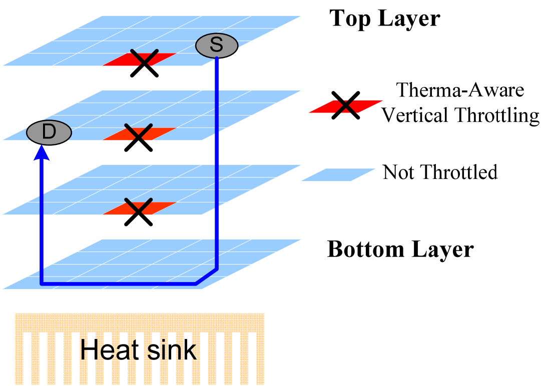


Fig. 2-3 Downward routing is applied to detour throttled routers. [22]

2.4 Problem of Data Delivery in Non-Stationary Irregular Mesh 3D NoC

Downward routing can detour throttled routers successfully, and guarantee the routing path to destination router. However, if we use downward routing vertically to detour throttled router, all packet will congest around throttled router and the bottom layer, and the traffic become unbalance and performance will degrade soon. This is not algorithm we think as high performance based on throttling. We should consider reasons for packets being blocked in 3D NoC, and know how to solve them by our proposed algorithm.

Before describing reasons for fail delivery, we know NoC can be divided to five layers [33]: Application layer, Transport layer, Network layer, Data link layer and Physic layer, shown in Fig. 2-4

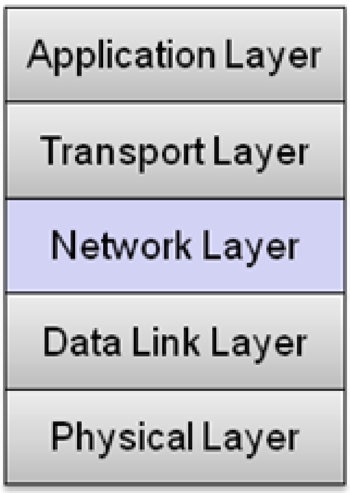


Fig. 2-4 NoC composed by five layers.

And we can define five layers as:

- **Application layer:** Network architectures and control algorithms constitute the infrastructure and provide communication services to the end nodes, which are programmable in most cases.
- **Transport layer:** Atop the network layer, the transport layer decomposes messages into packets at the source. It also resequences and reassembles the messages at the destination. Packetization granularity presents a critical design decision because most network-control algorithms are highly sensitive to packet size.
- **Network layer:** This layer implements end-to-end delivery control in network architectures with many communication channels.
- **Data link layer:** Data-link protocols increase the reliability of the link, up to a minimum required level, under the assumption that the physical layer by itself is not sufficiently reliable.
- **Physical layer:** The physical layer is an unreliable digital link in which the probability of bit upsets is non-null. And it composed the basic physical connection between any nodes.

We know the routing and throttling occur in network layer. In [22], when the temperature of router surpasses the thermal limit, we should trigger throttling to prevent overheat, which makes system unstable or break. To efficiently cool down overheated tiles, the Vertical Throttling in [22] shuts down the overheat tile and the tiles below it, except at the bottom layer. However, when triggering throttling (no matter single or pillar routers), the performance degrade. Since the throttled tiles are unavailable, network packets cannot directly go through but take a turn to detour them. Consequently, routers neighboring to the throttled tiles become more congested, and more seriously, the throttled routers make the packet in network layer with no routing paths to destination.

To ensure the success of packet delivery in a NSI-mesh network, we should prevent the occurrence of all the following four cases:

(i). Source router is not serving

(ii). Destination router is not serving

(iii).Any router on selected path is not serving

(iv).Any required channel on the selected path is occupied (Head-of-Line, HoL blocking)

The first one, as shown in Fig. 2-5(a), the source router is fully throttled. The second one, as shown in Fig. 2-5(b), the destination router is fully throttled. The third case, as shown in Fig. 2-5(c), some of the router on the routing path is fully throttled. The last one is shown in Fig. 2-5(d), where the channels on the routing path are blocked by other blocked packets, and we take vertical dimension as example. (iv) is also emerges in the horizontal dimension, and we call (iv) as Head of Line Blocking (HoL).

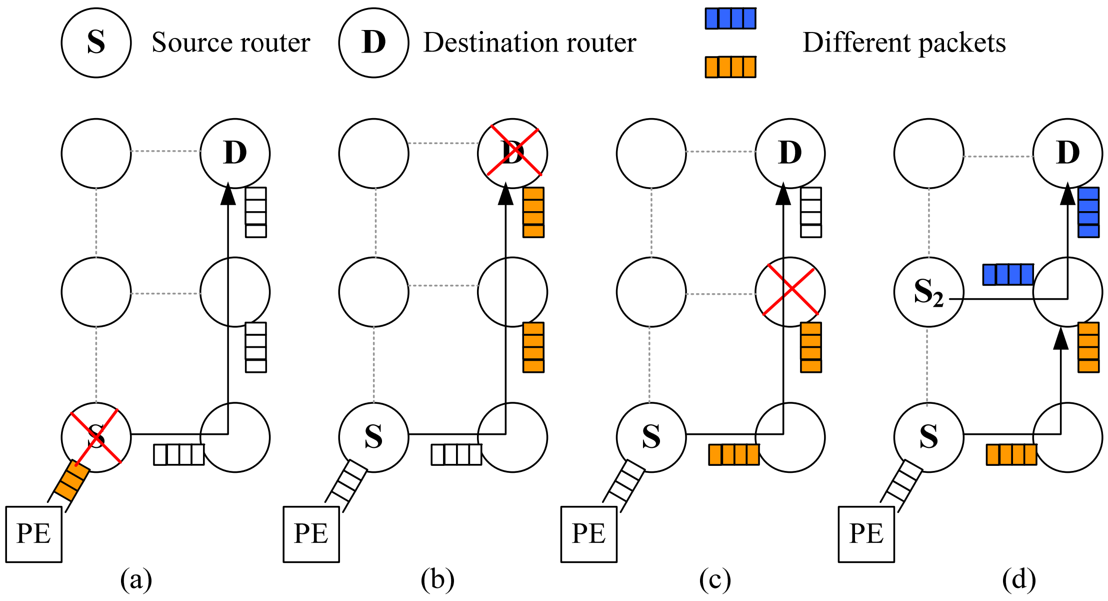


Fig. 2-5 Problem of usual reactive routing: (a) Source router is not serving. (b) Destination router is not serving. (c) Any router on selected path is not serving. (d) Head of line Blocking.

We know routing emerges in network layer, and the congestion also emerges in network layer. We should try to solve the problems (i)-(iv) by considering applying layers of NoC. We know about throttling influences performance and should minimize the influence of throttling, but we cannot solve the problem caused by run-time thermal management only by network layer. We should consider other layers to help solve our problem. For problem (iv), we can solve in flow control layer (data link layer), like virtual channel or increasing buffer or link, to prevent packet block by other packet. And we know if we solve other three problems, we can eliminate this HoL blocking problem. If we eliminate other three problems, HoL blocking will sometimes emerge, and it block for up to hundreds of cycles, which is different to 10^7 cycles caused by run-time thermal management. If we block for 10^7 cycles, it will decrease our performance a lot. (i) to (iii) is our consideration, because it influences our performance more than (iv).

For application layer, it consider all the system and algorithm what it should execute. It does not need to consider these detail problems, so we exclude it. For physical layer, it is too detailed for use to consider this problem, so we still exclude it. For transport layer, we can solve tow problems (i) and (ii). For transport layer, if we have source router and destination router status, we can understand source or destination router is serving or not. However, we can determine transmit packet to network layer or not in transport layer. We can eliminate these two problems in transport layer.

For the rest problem (iii), we know throttling problem emerge in network layer, so we consider solving this problem in network layer previously. Nevertheless, we cannot solve it only in network layer. Because we cannot predict next router routing path with no throttled router, we cannot guarantee that the routing path has no throttled routers. Additionally, we cannot see all NoC buffer and router status to determine which routing path is routable, because it is source routing, and it differs from our constraints and goal. We can take Fig. 2-6 as an example.

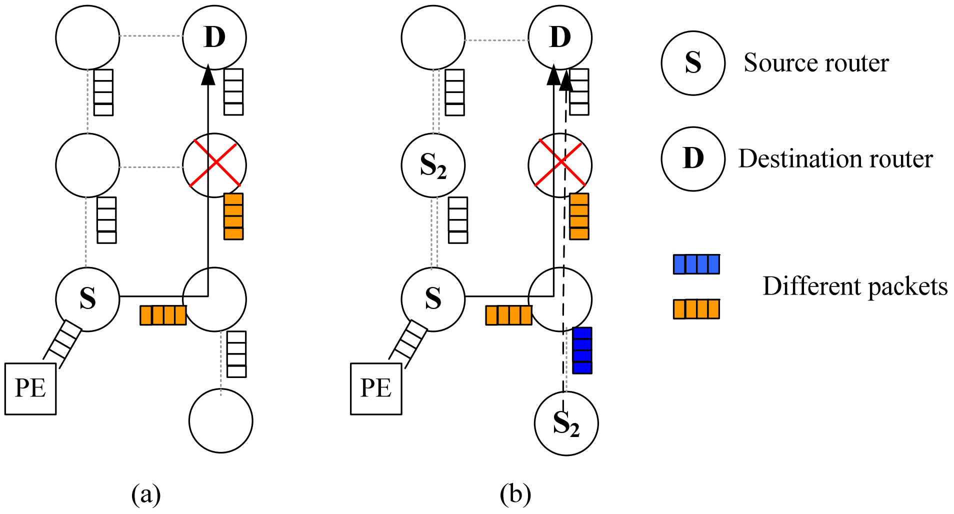


Fig. 2-6 (a) (iii) situation cannot guarantee routing path which router choose is routable. (b) Block other packet, which is same as (iv).

In Fig. 2-6(a), from source router, we know the router in east is not throttled, so we may go eastern router. But after we arrive at eastern router, we only have northern router to route, and it is throttled. So packet is blocked by throttled router, and it will be block for 10^7 cycles, and the other packet is blocked by this packet. As shown in Fig. 2-6(b), the packet blocked by throttled routers will block other packets, and the congestion tree will grow soon to whole network.

To completely remove the case of (iii), we have to jointly consider the available information of the network layer and transport layer. Here we choose the style of distributed routing instead of source routing for performance consideration. Although traditional source routing can be applied in this scheme, the computation overhead of source routing for optimizing performance of NSI-mesh is too high. Besides source routing cannot balance the loading of the network by adapting the network information as adaptive routing. If the topology of the network is far from regular mesh, it would be difficult to use source routing. The small changing interval and large range of inactive number characteristics of throttled NSI-mesh make conventional routing algorithms infeasible. The routing algorithms for irregular-mesh [34] are not feasible owing to the non-stationary characteristics. Besides, the regulations of the location of oversized-IP make the conventional algorithms infeasible because throttling may be required for all the upper layer routers. Moreover, the offline optimization effort for routing in irregular-mesh is not affordable for the online computation of the throttled NSI-mesh. The fault-tolerant routing algorithms those detour packets from faulty routers could be candidates. However, the characteristics of faulty NSI-mesh and throttled NSI-mesh are very different. The number of faulty router is non-decreasing but usually small. Besides, the interval of topology transformation in faulty NSI-mesh is much longer and unpredictable. The topology changing of faulty NSI-mesh is occurred after detection, testing, and reconfiguration of the system. Usually the latter two operations are even done in the reboot sequence, which makes the problem going back to traditional offline irregular-mesh. Similarly the regulations of the location of faulty routers make the conventional fault-tolerant routing algorithms infeasible because throttling may be required for all the upper layer routers.

If we use transport layer to solve (iii), it still not work. If we know status of source and destination router, we still cannot guarantee the packet can route to destination router successfully. When a packet inject form transport layer to network layer, we think it can transmit to destination router in previous knowledge, excluding congestion or head–of–line blocking. It can work in normal NoC, but it fail in throttled NoC.

We can conclude throttling problem emerge in network layer, but we cannot solve (iii) only in network layer or transport layer. If we cannot solve (iii) situation, the whole network will stop for a period of time. Therefore, we should combine these two layers to solve the (iii). See in Fig. 2-7.

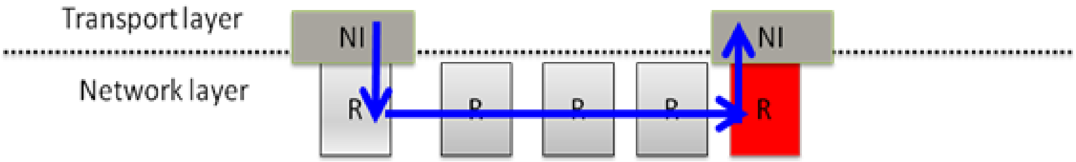


Fig. 2-7 Transport layer and network layer operation.

2.5 Summary

We reference related works for thermal-aware 3D NoC, and discover a new problem: Non-Stationary Irregular Mesh (NSI-Mesh). We find four problems of delivery packets, and also show how to solve them by using transport layer and network layer. Finally, we conclude that we need to joint transport layer and network layer to solve the problems caused by NSI-Mesh.

Chapter 3
Transport-Layer Assisted Routing

Transport-layer assisted routing is composed of the transport layer assisted routing schemes and the dual-mode routing algorithms. Transport layer shares topology information with network layer for high performance in NSI-mesh. Network layer follows the initial routing decision provided by transport layer, and tries to balance the lateral traffic loading. In this chapter, we introduce the proposed Transport-Layer Assisted Routing (TLAR) schemes and algorithms.

3.1 Operation Flow of Transport-Layer

The proposed operation flow of Transport-layer assisted routing is shown as following.

The system of 3D NoC is switching between the normal stage and the reconfiguration stage. In normal stage the 3D NoC works as usual irregular or regular mesh network. In this stage, we assume distributed thermal sensing mechanism is embedded in the network for each router to obtain its own temperature, and each router has a timer for synchronizing their operation stages. After N-cycle normal stage, the network enters the R-cycle reconfiguration stage. The reconfiguration stage means that we should prepare some management and controller, which let 3D NoC remains execution in normal work. In comparison with the cycle number in normal operation stage, the cycle number required for reconfiguration is very small. Here we assume the network is operated at 1GHz. In each 10ms interval, 10^4 cycles is absolutely sufficient for each tile to reconfigure, and N is around 10^7. The reconfiguration stage only occupies 0.1% of the total available time, so the overhead of reconfiguration is negligible. If the interval is 100ms, the overhead is 0.01%, which is more negligible. The reconfiguration stage, shown in Fig. 3-1, consists of three sub-stage: (i) cleaning up and policy determination; (ii) synchronization of topology information; (iii) routing mode checking and throttling. The detail is described as following:

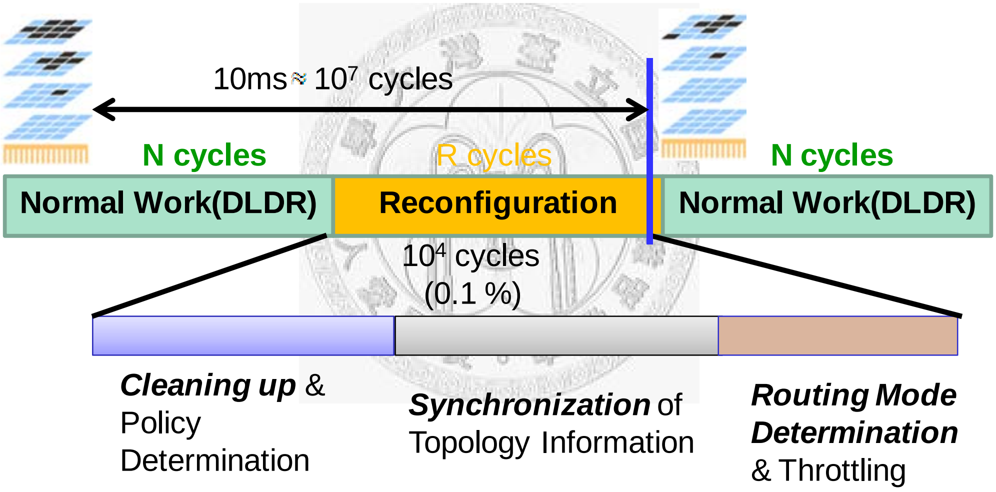

10ms ≈ 10⁷ cycles
N cycles
R cycles
N cycles
Normal Work(DLDR)
Reconfiguration
Normal Work(DLDR)
10⁴ cycles
(0.1 %)
Cleaning up & Policy Determination
Synchronization of Topology Information
Routing Mode Determination & Throttling

Fig. 3-1 Network states and operation stages in transforming topology for run-time thermal management.

(i) Cleaning up and policy determination: In order to make sure packet transport t0 destination router successfully in next normal work, the network has to be cleaned up before topology changing. In this stage, the packetization of the payloads from transport layer to network layer is paused. As shown in Fig. 3-2, the payloads stay in the transmitter payload queue. In this stage, we should not only stop transmitting packet form transport layer to network layer, but also deal with the rest packet still in network layer. It means transmitter packet queue will become empty after a small period of time. In the meanwhile, the distributed thermal-aware controller in each tile should determine the throttling of the router within the tile for the next normal stage. The implementation of thermal-aware management can be in the transport layer controller or in the application layer as a software routine. No matter which layer the policy is determined, the application layer and transport layer share the information of control policy of this tile. The important thing for us is that the new throttling emerges for the next normal work stage, and we should guarantee no packet still in network layer is blocked in next normal work stage.

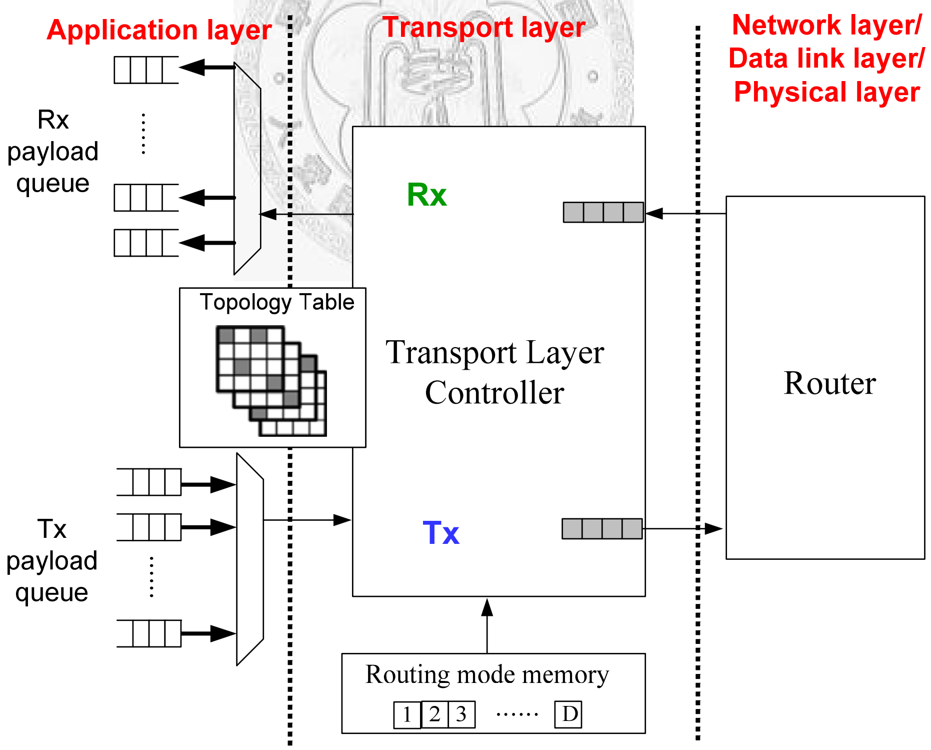


Fig. 3-2 Block diagram of transport layer in the tile of thermal-aware 3D NoC.

(ii) **Synchronization of topology information:** If we trigger throttling, we should let every router in 3D NoC know which router is throttled and how the topology change in next normal work. In this stage, all routers have to transmit packets containing their throttling information to all their upstream and downstream routers. No matter in current normal stage the router is fully throttled or not, it is not throttled in this sub-stage. Because all the routers are not throttled in this sub-stage, the network is regular mesh in each layer. We can see topology table in Fig. 3-2 which is shared by application layer and transport layer. In this topology table, each router requires one bit for representing the state of each router in the next normal stage. If a router is fully throttled in the next normal stage, the corresponding bit will be toggled to inactive. Otherwise the bit will be active. Then the information of topology is synchronized to each tile. The technology of transmitting throttling information is not our consideration. Because the throttling is triggered by 10ms, the transmission of throttling information is just up to hundreds of cycles, and it is just 0.1% of 10 ms. We have 99.99% time of normal work, and we collect correct throttling information and make correct routing selection. We can see in Fig. 3-3, so the transmission of throttling information is not our problem in NSI-mesh.

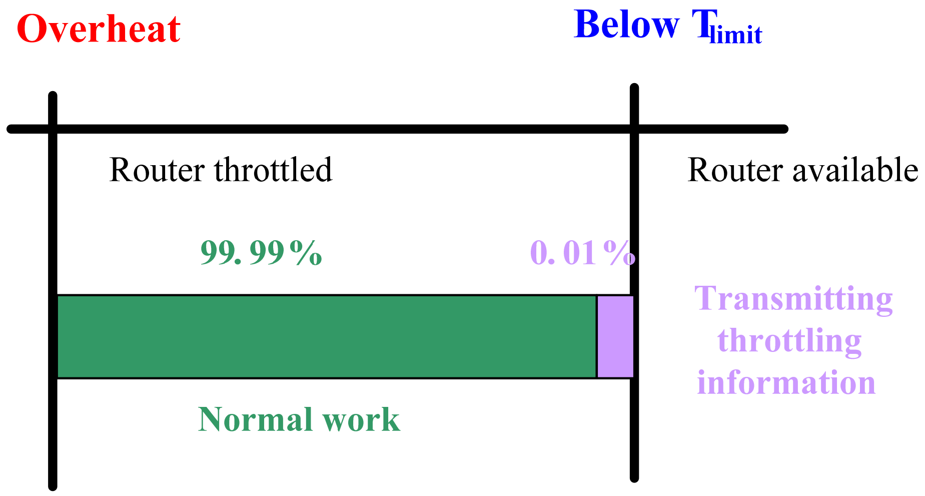


Fig. 3-3 Required time of transmitting throttling information.

(iii) Decisions of routing mode and throttling: In this stage, the throttling of router is applied now. If all routers in 3D NoC are not throttled, the routing is just like in regular mesh. But when throttling is trigger, we need determine routing mode for transmission toward each destination router in the transport layer. If the source router is throttled, the payload will stay in the transmitter payload queue. If the source router is not throttled, we should execute transport-layer assisted routing (TLAR) to check all routing mode of destination router for ensuring no packet is blocked by run-time thermal management. After executing TLAR, network goes back to the normal stage, and the packet injection continues for the tile where the router is not throttled.

3.2 Proposed Framework of Transport Layer Assisted Routing

To correctly select a path that makes packet delivery success, we propose the Transport Layer Assisted Routing Scheme (TLAR) for packets with non-fully throttled source and destination routers. The routing in TLAR is based on our previously proposed downward routing, which is a combination of vertical routing and lateral routing. The key idea of TLAR is that the throttling information in transport layer is used to assist the selection of layer for lateral routing and the decision of routing algorithm in network layer. The selection and decision results, what we defined as the routing mode, are saved in packet header when the packet is injected to the network layer, and then the routers follow the mode to route.

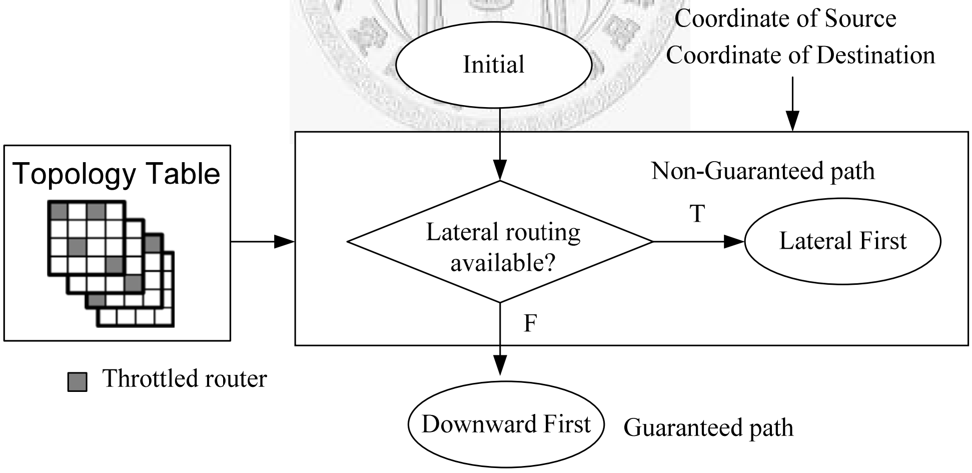


Fig. 3-4 Framework of proposed transport layer assisted routing scheme. The determination of lateral routable relies on the throttling information in transport layer.

Fig. 3-4 shows the flow chart of path selection in TLAR. The checking of lateral path is done during the reconfiguration stage for each destination in the transport layer above the source router. For the packet which is going to a lateral routable destination, it is routed first laterally. Otherwise downward path is selected because it is guaranteed routable. As shown in Fig. 3-2, the overhead of TLAR is the small memory for storing the checking results as the routing modes. In normal operation, the transport layer controller reads the routing mode from the memory and set the packetizer. Then the payload is packetized with the routing mode specified in header..

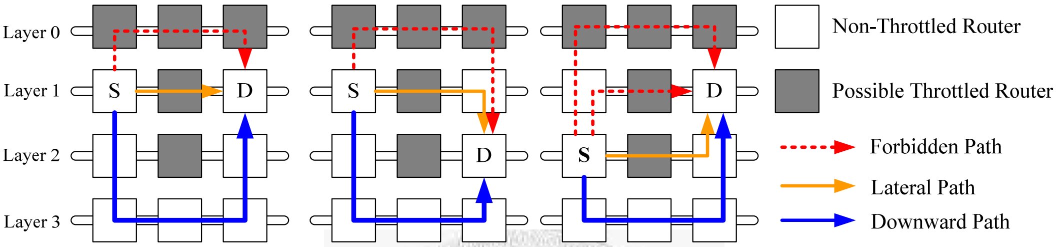
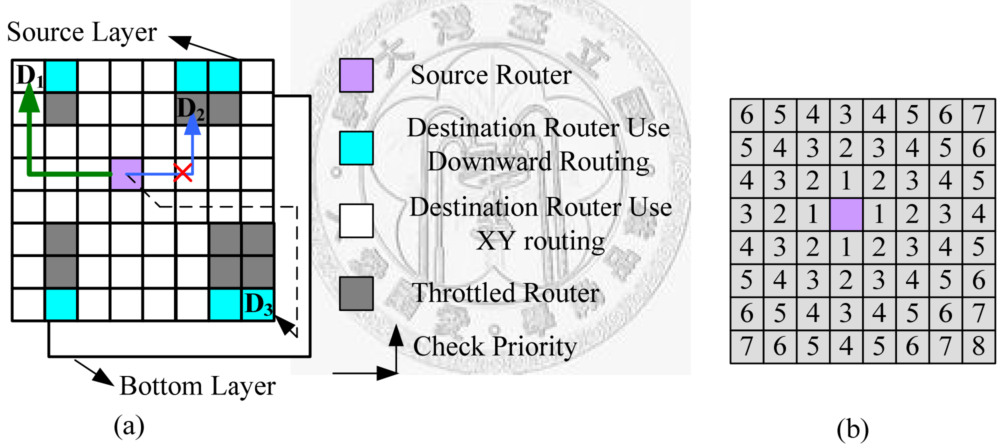


Fig. 3-5 Path selection of proposed transport layer assisted routing scheme.

In TLAR, packets change their z-location only when it is at the source or the destination xy-location, and the selection of routing path is dependent on the relative vertical location of source and destination, as shown in Fig. 3-5. As mentioned before, the routers in the bottom layer are never throttled. Therefore if source and destination routers are not fully throttled, the vertical path and lateral path through bottom layer will be guaranteed routable. If there is no fully throttled router on the non-guaranteed path, TLAR chooses this path for lateral routing. Owing to the bandwidth required for downward routing, TLAR prevents to choose layers below source router and above the bottom layer for lateral routing. Checking if the lateral path is routable for these layers also multiples of the computation overhead for path selection. Any lateral path locates above source router is forbidden owning to the limitation of turn model. As the proof in our previous works [33], the combined routing is deadlock-free if the lateral routing is deadlock-free and we remove the {UN, UE, US, UW} turns.

3.3 Proposed Algorithm: Downward-Lateral Deterministic Routing (DLDR)

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, a dimension-ordered routing (DOR), as the deterministic routing.



6	5	4	3	4	5	6	7
5	4	3	2	3	4	5	6
4	3	2	1	2	3	4	5
3	2	1	🟪	1	2	3	4
4	3	2	1	2	3	4	5
5	4	3	2	3	4	5	6
6	5	4	3	4	5	6	7
7	6	5	4	5	6	7	8

Fig. 3-6 (a) TLAR examples, and (b) checking dependency.

An example of routability is shown in Fig. 3-6(a). 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 XY routing; an example path is shown by the green line. Third, the pale blue blocks are destinations those are only downward-routable. An example of the path of downward routing is shown by the 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 is changed, the routing mode must be decided 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 incremental breadth-first search (BFS) style, as shown in Fig. 3-6(b). The dependency is based on XY-routing, and the prerequisites that a node routable is its previous node also routable. For an 8x8x4 network, the operation is completed in 63 cycles.

3.4 Summary

In this chapter, we propose the Transport-Layer Assisted Routing (TLAR) scheme. TLAR can make sure the delivery packets can reach destination router successfully, and the operation flow and checking of TLAR is also described here. Based on TLAR, we propose DLDR for sending packets with XYZ routing or downward routing.

Chapter 4
Performance Evaluation

4.1 Setting of Simulation Environmentst

Currently there is no real chip implementation of 3D NoC systems, so we start from modeling the 2D NoC system implemented in [30] and stack it to multiple layers. For network simulation, we start from Noxim [28], and we extend it to the third dimension. For temperature simulation, we use Hotspot [29], and We adopt the tile geometry and power model of Intel's 80-core processor [30]. We first add the model of basic 3D router and the Dimensionally-Decomposed (DimDe) router [23], and modified Noxim to generate the 3D-NoC architecture and the floorplan based on user-defined parameters of dimension. During network traffic simulation, a power trace is generated based on the power model of the NoC. The power trace and physical floorplan are input to the thermal model. In the proposed simulator, the tile geometry and power model are based on Intel's 80-core chip. Fig. 4-1 shows the construction of the co-simulation model and Fig. 4-2 shows the floor plan as we used. We adopt the basic wormhole flow control and use random arbitration for switch allocation

We construct 3D 8×8×4 NoC, and the packet length is randomly from 2 to 10 flits. The queue depth of each input channel is 16 flits, and the link level flow control protocol is full hand-shake request and ack. Because TSVs generally have high bandwidth, a crossbar-based vertical connection is assumed in our 3D router [10]. For each tile in the NoC, the tile area is 2.0mm × 1.5mm and the router area is 0.65mm ×

0.53mm.

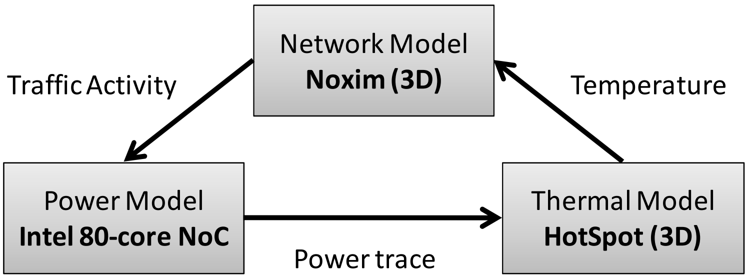


Fig. 4-1 Framework of co-simulation platform.

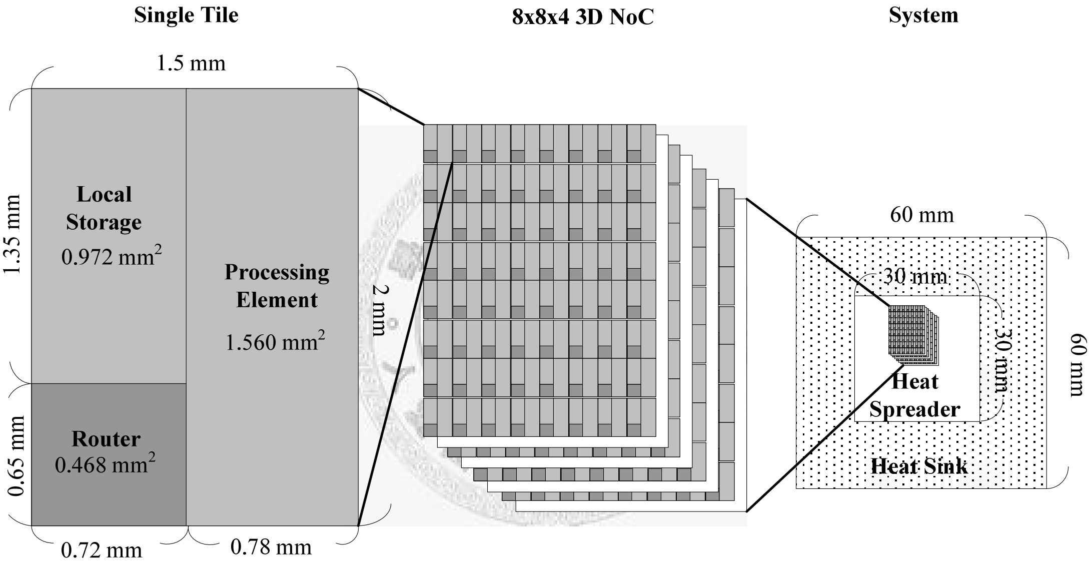


Fig. 4-2 Construction of the model of a 4x4x4 3D NoC with simplified tile model from. [30]

To keep the performance indices representative and comparison as fair as possible, several modifications of the simulator are required for modeling the TLAR. In Noxim[28], the statistics of received packet number, packet latency, and network throughput are based on the received packets during the simulation period while the network is assumed stable. The payloads toward the fully throttled destinations are hold in the transport layer and not packetized, and only the deliverable payloads are packetized and injected to the network. Originally the injection rate is simulated by generating Poisson arrived packets of given traffic distribution of destinations over the network. The network injection rate of the active routers follows the index by escaping the packets to the throttled destinations and regenerating more packets toward non-throttled destinations. Because we assume application layer and transport layer share the topology information, the packet injection process of the throttled router is paused until it is not throttled. In this setting, the total injection rate of the network can be obtained by multiplying injection rate and the number of the active routers, and the statistics of performance indices are not affected by throttling.

In this chapter we show the performance of the proposed TLAR algorithms. We use two throttling cases of vertical throttling: (a) 1 throttled router, and (b) 2 2x2 throttled region. In case (a), the one throttled routers is located in the center of the most top layer of the 8x8x4 network. In case (b), 8 1x1x3 pillars are throttled on the diagonal line of the upper three layers of the 8x8x4 network. And they are shown in Fig. 4-3.

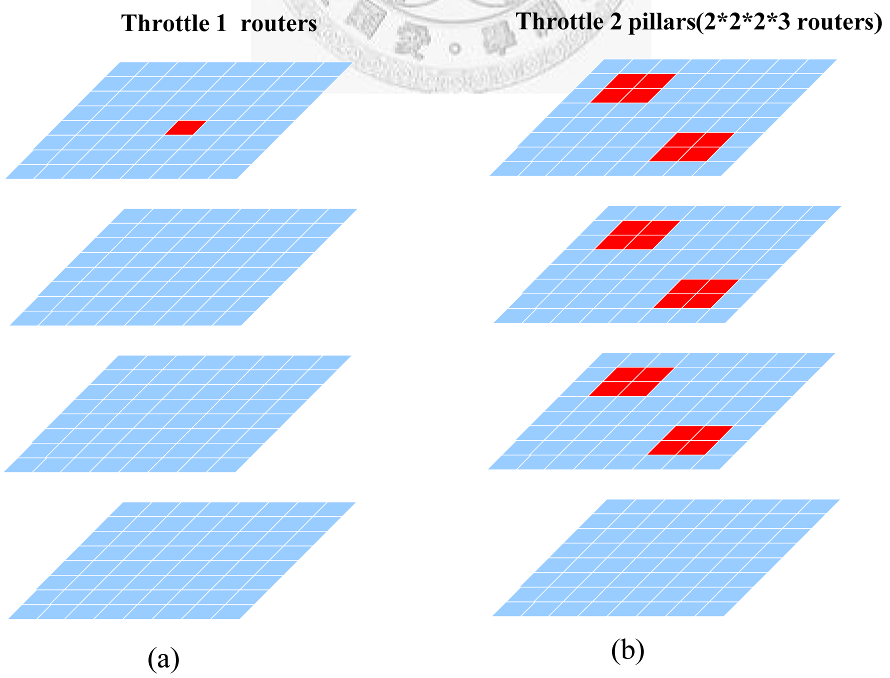


Fig. 4-3 Throttling cases: (a) 1 router (b) 2 2*2*3 routers.

4.2 Traffic balance and rate of transmitting packet under different routings

First we use statistical traffic load distribution (STLD) [34] and decision distribution to show the network loading. All the experiment in Fig. 4-4 and Fig. 4-5 use the same injection rate that makes average latency of TLAR-DLDR twice the zero load latency.

Fig. 4-4(a) shows the STLD of the baseline downward routing and Fig. 4-4(b) shows the TLAR-DLDR algorithm, which is the combination of downward routing and TLAR. Though there is only one router throttled in the network, some packets have to be routed downward through the bottom layer. The congestion degree of DLDR in the bottom layer is reduced and the loading of the work is more balanced by using proposed DLDR. The packets in the upper layers are more balanced, because the congestion in the bottom layer is relaxed. We use Fig. 4-4(c) to show distribution of the routing mode decision. In DLDR scheme, 80% packets are determined to route on the deterministic paths in the source layers and 20% packets are routed in the downward mode. We can prove we decrease the rate of downward routing by proposed DLDR.

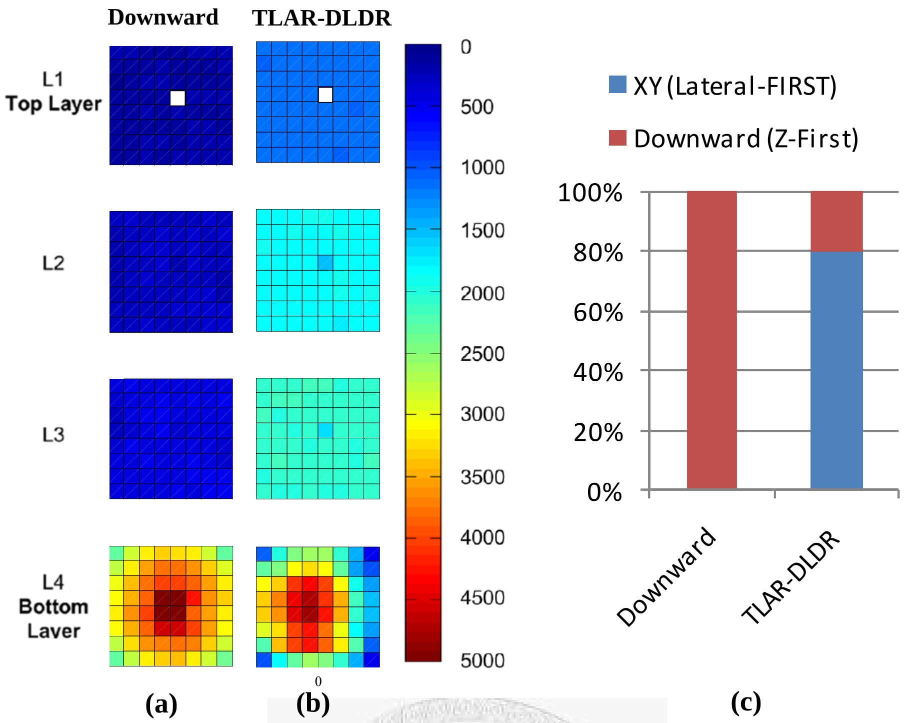


Fig. 4-4 (a) Statistical traffic load distribution (STLD) of conventional design; (b) STLD of proposed TLAR framework with DLDR algorithm; (c) latency vs. network injection rate.

Fig. 4-5 shows similar results when we increase the number of disconnected throttled routers and the size of the region. We increase the number of throttled routers from 1 router to 24 routers. The congestion degree of downward in the bottom layer is still larger than DLDR, because downward cannot balance the loading of the bottom layer. The packets the upper layers are more because the congestion in the bottom layer is relaxed by using of our proposed DLDR algorithm. We use Fig. 4-5(c) to show distribution of the routing mode decision. In DLDR scheme, 40% packets are determined to route on the deterministic paths in the source layers, and 60% packets are routed in the downward mode. The DLDR algorithm has fewer packets choosing the downward mode, so DLDR is more vertical balanced than downward routing.

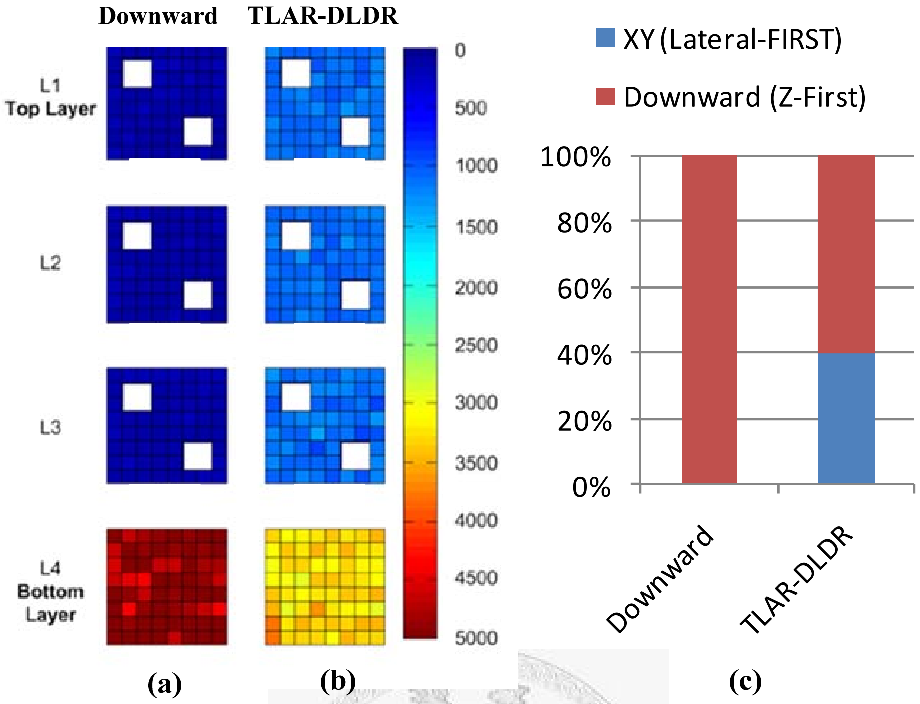


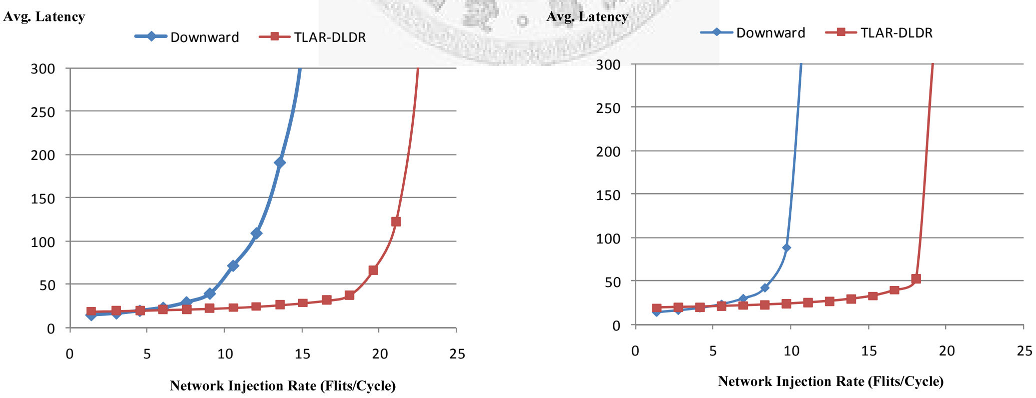
Fig. 4-5 (a) Statistical traffic load distribution (STLD) of conventional design; (b) STLD of proposed TLAR framework with DLDR algorithm; (c) latency vs. network injection rate.

We use Table 4-1 to show the statistics of the statistical traffic load distribution (STLD). As we can see, the mean packet number is increased by adopting the TLAR scheme and both total and inter-layer standard deviations are reduced by applying DLDR algorithm. The statistics is corresponded to the performance simulations, which are shown in Fig. 4-6. In the case of 1 throttled router, there are 255 active routers injecting packets to the network. For the case in Fig .4-3(b), only 232 routers are active. These active routers can transmit (receive) the packets to(from) the network. Because of the more balanced loading of the network, the DLDR has better performance than the baseline algorithm downward routing in both 1 and 2 2x2x3 throttling cases. The throughput in Fig. 4-6(a) is improved by 95% by adopting the

DLDR algorithms. In Fig. 4-6(b), the throughput is improved by 70%. We can see our proposed algorithm DLDR is outperforming than downward routing.

Table 4-1 Statistics of statistical traffic load distribution.

		One Throttled router		Two 2x2x3 Throttled Pillars	
		Downward	TLAR-DLDR	Downward	TLAR-DLDR
Mean	L1	330.75	798.6	149.7	1113.5
	L2	403.05	1598.6	147.4	1100.7
	L3	546	2004.2	148.2	1110.4
	L4	6495.24	5208.84	4960.8	3280.9
	Total	2372.55	2633.76	1476.0	1707.6
Stdv.	L1	27.9	11.85	12.8	74.9
	L2	31.2	15.3	12.0	70.8
	L3	44.85	98.1	12.2	97.3
	L4	2067.6	1446	280.2	185.3
	Total	3097.44	2190.84	2160.5	980.4
	Inter-Layer	2821.28	2002.88	2406.2	1086.4



(a)

(b)

Fig. 4-6 Average latency vs injection rate with (a) one router throttled and (b) two 2x2x3 pillars throttled.

4.2.1 Network Sustainability and Degree of Graceful Degradation

Here we show the network sustainability and the degree of graceful degradation. Network sustainability describes the total throughput provided by the network while some parts of the network are not working. If a router is throttled, it cannot provide the bandwidth for packet delivery. With higher network sustainability, the 3D NoC can provide larger throughput when there are routers fully throttled. Because we cannot simulate all throttling cases, we simulate extreme cases from on throttled routers to 7x7x3 throttled routers, as shown in Fig. 4-7. As shown in Fig. 4-8, all algorithms degrade as the size of throttled region increases. Here we want to observe the throughput degradation in the different case of vertical load balancing of different irregular topologies occurred in NSI-mesh. We start from the 1x1x1 throttling case and then extend to the 7x7x3 throttling case. All the throttled regions are located in the center of the xy-plane. In all cases, the DLDR algorithms have better performance the conventional reactive downward routing. In comparison with conventional reactive downward routing, the proposed TLAR-DLDR can averagely improve the sustainable throughput from 85.5% to 48%.

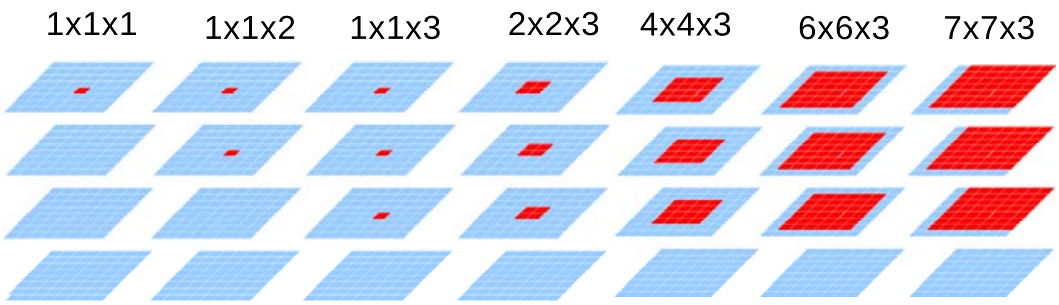


Fig. 4-7 Different throttling cases.

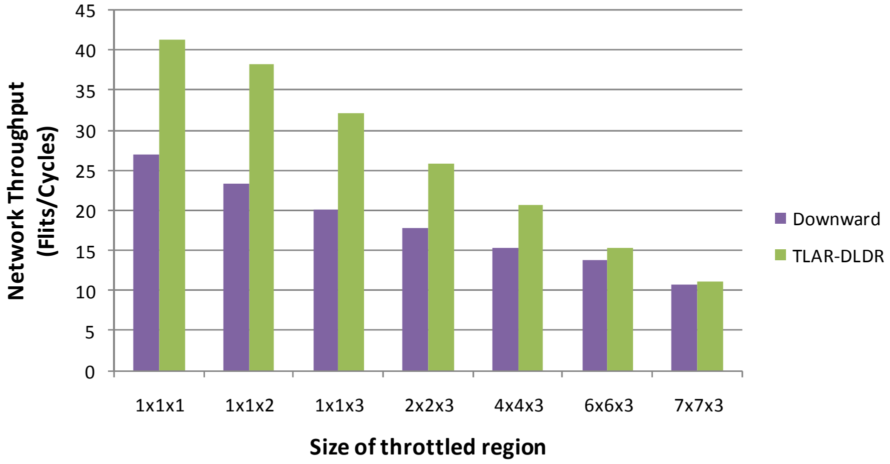


Fig. 4-8 Network sustainability of NSI-mesh 3D NoC, uniform traffic offered.

4.2.2 Run-Time Temperature and Throughput

Here we show the real case for simulating TLAR in run-time thermal management. The simulation setup is as same, and the only difference is that we do not fixe location of throttled routers in simulation. Thermal-Aware Vertical Throttling (TAVT) is adopted in RTM to throttle overheat routers, and TLAR will detect throttled routers and detour them. As shown in Fig. 4-9, the total simulation cycle in network simulator is M+1000K cycles. The total simulation time for temperature is 10 seconds, and is divided into 1000 10ms intervals for observing the transient-state temperature. For each 10ms interval, our network simulator uses K cycles to evaluate the transient power. The network simulator first runs for M cycles to warm up the network, and the thermal simulator setups the ambient temperature T_{amb} and initializes the temperature distribution T_0. For each 10ms interval, the network simulator uses K cycles to estimate the power distribution, which is denoted as $P_{(t\ ,t+10)}$. Then the thermal simulator is called to estimate the transient-state temperature T_{t+10} based on given short-term power distribution $P_{(t\ ,t+10)}$ and the beginning temperature distribution T_t. In this simulation, K=50000 cycles for reaching steady state of traffic between thermal checking, and we focus on transient temperature and throughput of each interval.

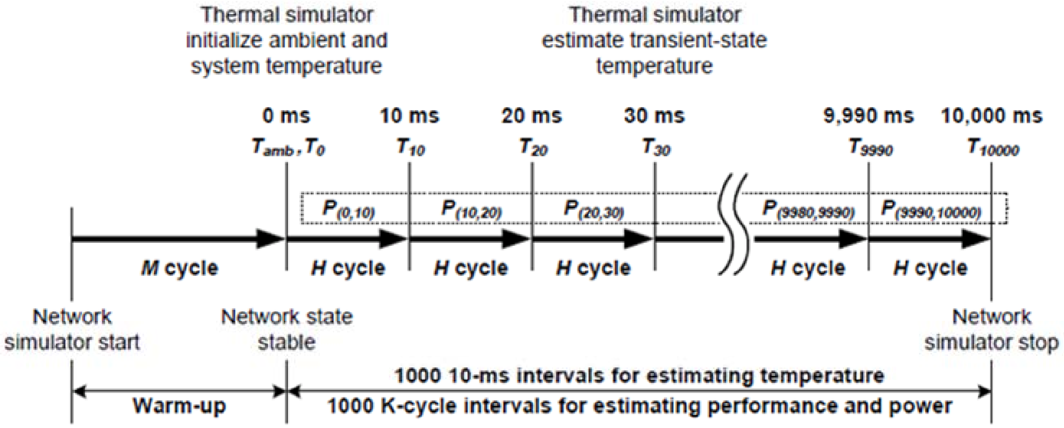
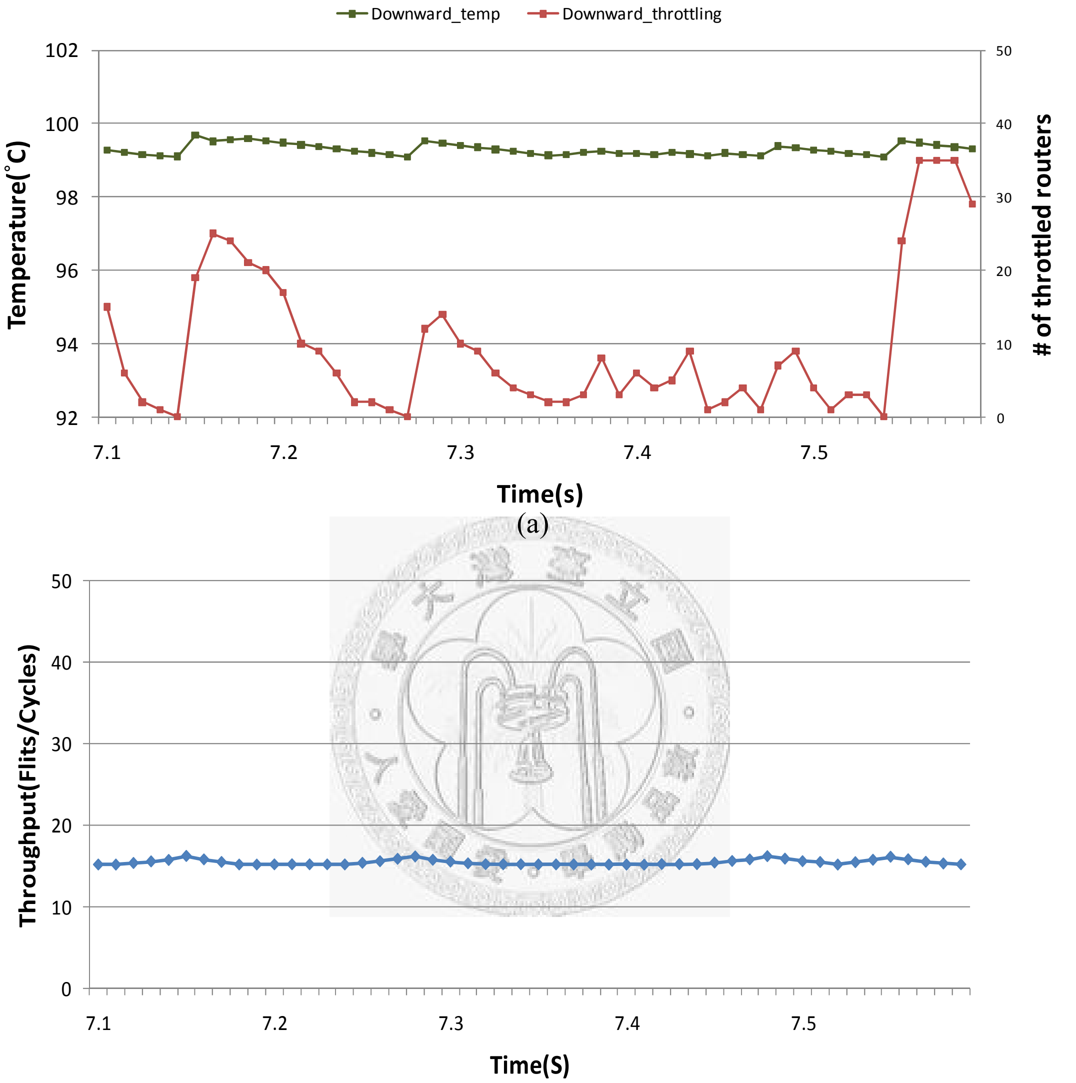


Fig. 4-9 Mutual-coupling co-simulation for throughput and temperature evaluation.

We show the temperature and numbers of throttled routers from 7.1s to 7.6s. As shown in Fig. 4-10(a), the transient temperature is below thermal limit, because TAVT effectively controls the temperature. The number of throttled routers means the changing of topology, from regular to irregular and back to regular. The throughput of downward routing is shown in Fig. 4-10(b), and the average throughput between 7.1s to 7.6s is 15.4 (Flits/Cycles).

Fig. 4-11(a) shows the temperature and throughput of TLAR with same simulation setup. The average temperature of TLAR is 0.15°C more than downward routing, which is relatively small to changing of temperature. However, as shown in Fig. 4-11(b), average throughput of TLAR is 25.4 (Flits/Cycles), which is improved 66% in comparison with downward routing. The temperature of TLAR is a little higher than traditional downward results from the larger throughput. TLAR delivery more packets, so the power is higher. Therefore the transient temperature is high, too.



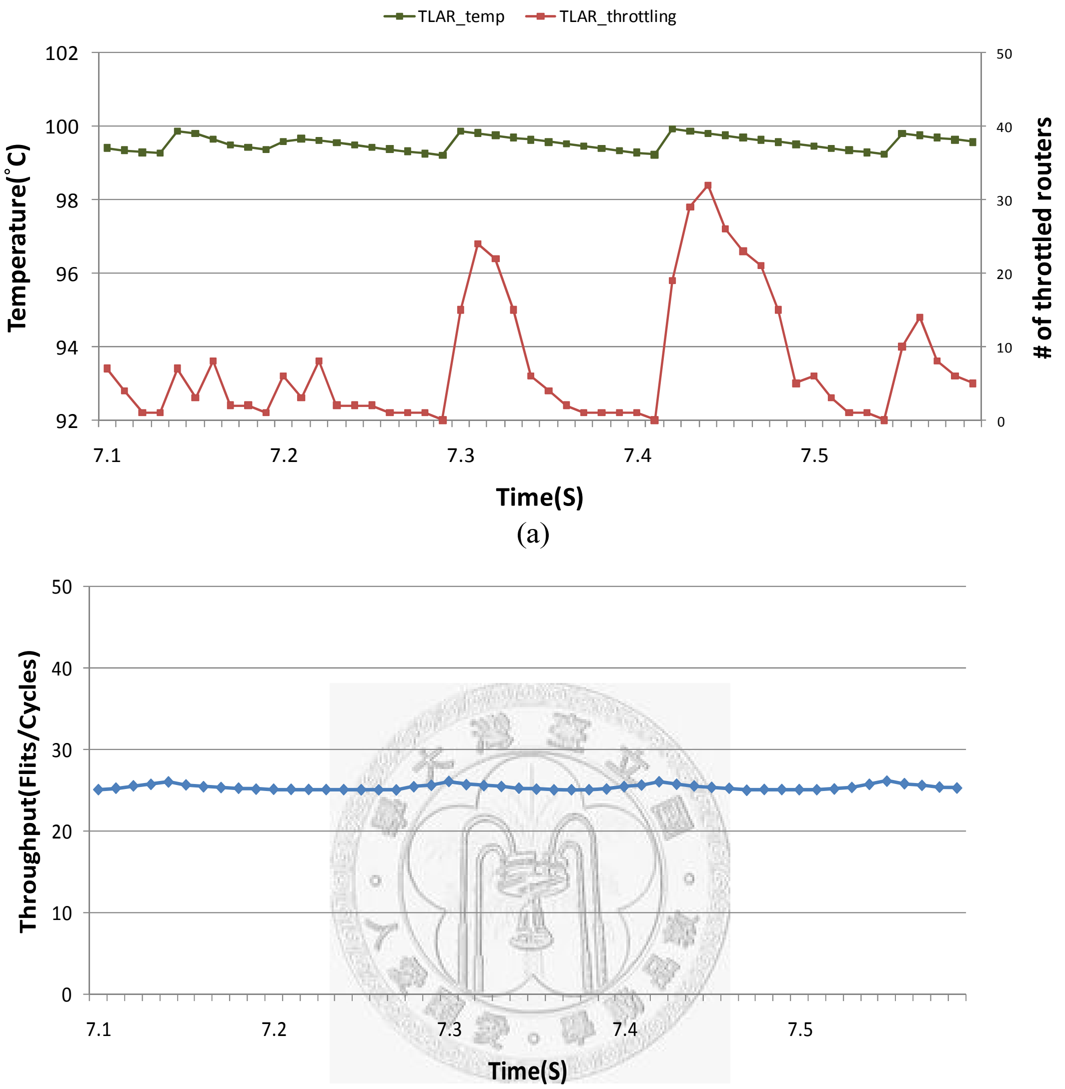
(a)

(b)

Downward Routing	
Avg. Temp.(°C)	99.35
Avg. # of Throttling	9.5
Avg. Throughput	15.3

(c)

Fig. 4-10 (a) Temperature and numbers of throttled routers of downward routing. (b) Throughput of downward routing. (c) Statistics of downward routing.



(a)

(b)

TLAR	
Avg. Temp.(˚C)	99.5
Avg. # of Throttling	7.6
Avg. Throughput	25.4

(c)

Fig. 4-11 (a) Temperature and numbers of throttled routers of TLAR (b) Throughput of TLAR. (c) Statistics of TLAR.

4.3 Summary

In this chapter, we see the performance results of TLAR. For traffic loading distribution, the proposed TLAR can balance vertical loading than downward routing because of more lateral packets used for routing. The throughput is also higher than downward routing for fixed or non-fixed throttled region of routers. Finally, we simulate TLAR on real case. Although TLAR has a little higher temperature, the throughput result is better than downward routing.

Chapter 5
Architecture Design for Transport-Layer Assisted Routing

For successful data delivery and performance consideration, we propose transport layer assisted routing (TLAR) scheme. Here we propose a low cost and low latency architecture of TLAR.

5.1 Traditional Architecture and Dataflow

NoC is composed of five layers [33]: Application layer, Transport layer, Network layer, Data link layer and Physic layer. For architecture design, we focus on transport layer and network layer, because we use transport layer to assist network layer for routing decision, and we should consider are overhead comparing to traditional design. The implementation is composed of a router, which transfer hop by hop, and a network interface (NI), which implements the interface to the IP modules. The traditional architecture is show in Fig. 5-1.

Network interface is allocated in transport layer, as shown in Fig. 5-2. Network interface (NI) is the component that provides the conversion of the packet-based communication of the NoC to the higher-level, which communicate between application layer and network layer. Ni packetize message from application layer to packet transmitting to network layer. In Fig. 5-1, this is the traditional NI + router, and it can maintain traditional function of network interface, but cannot maintain the transport-layer assisted routing.

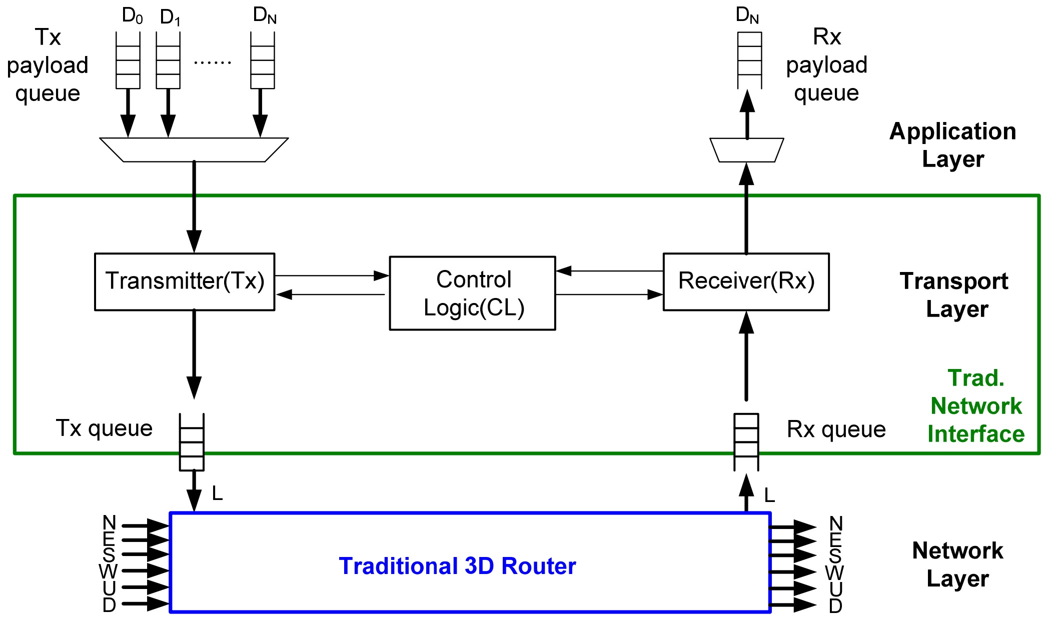


Fig. 5-1 Architecture of network interface and router in the traditional 3D NoC.

Routers deal with packet form network interface, and choose routing path to route packet to destination router, which transmit flits rather than packets, as shown in Fig. 5-2. Because the transport layer provides packets to network layer, routers should find routing path to destination hop by hop. Because traditional NI and router cannot maintain TLAR scheme, we propose TLAR network interface and dual mode router architecture for TLAR scheme, and we discuss them in following sub-chapter.

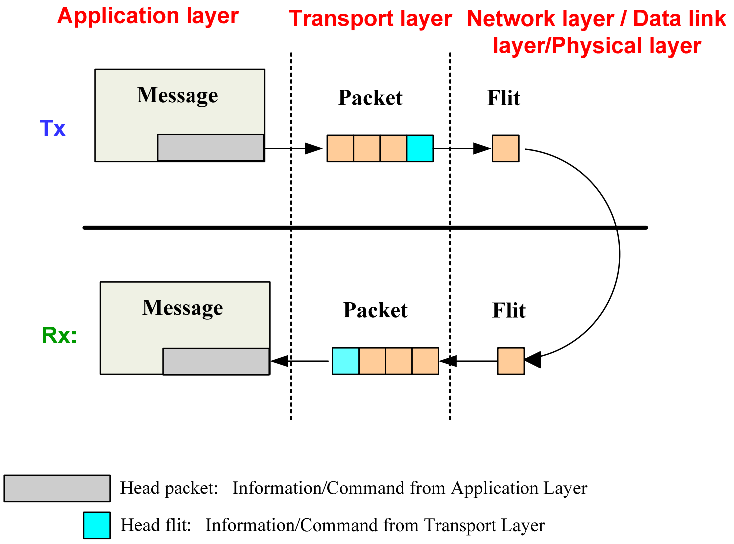

Application layer
Transport layer
Network layer / Data link layer/Physical layer
Tx
Message
Packet
Flit
Rx:
Message
Packet
Flit
Head packet: Information/Command from Application Layer
Head flit: Information/Command from Transport Layer

Fig. 5-2 Data flow between application layer, transport layer and network layer.

5.2 Network Interface Design

5.2.1 Control Logic and FSM

We know network interface is established in transport layer, and we should maintain original function of network interface and our proposed transport-layer assisted routing. Because traditional network interface cannot maintain TLAR scheme, we propose TLAR network interface, as shown in Fig. 5-3. We can divide our network interface to four major 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 1-bit throttling information of each destination, and updates on each topology change. Application layer and

transport layer share this information. TT is required for all NSI-mesh networks to solve the problem of (i) Source router is not serving and (ii). Destination router is not serving. 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.

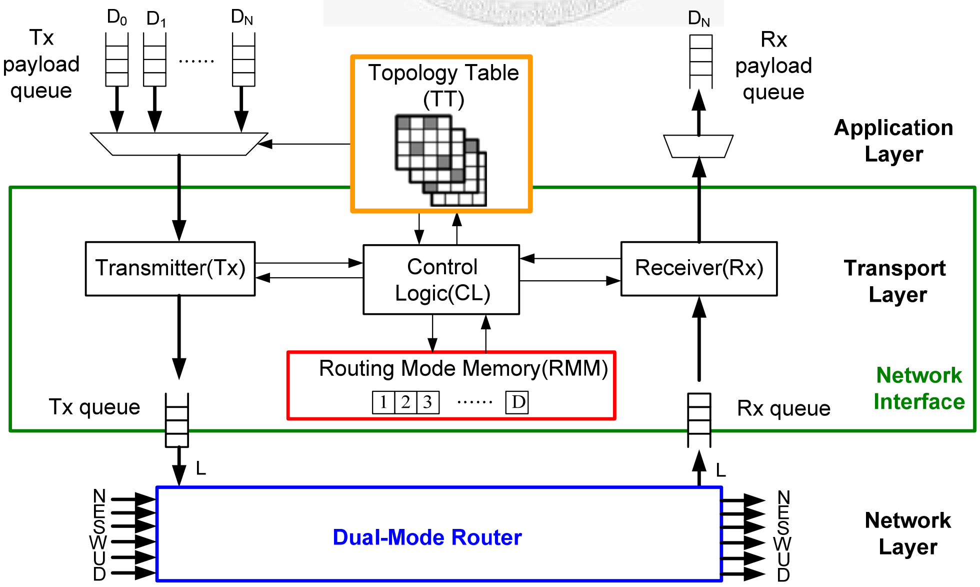


Fig. 5-3 Proposed architecture of TLAR.

5.2.2 Techniques of Memory Reduction

To prevent area overhead, we propose two memory reduction techniques to reduce area. The proposed memory reduction techniques are 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. 5-4(a). Therefore, the number of bits can be reduced from XYZ to XY $\log_2(Z)$, as shown in Fig. 5-4(b) and Fig. 5-4 (c). For example, for (x,y,z) direction = (N,M,K) of 3D NoC, we originally need N*M*K bits to store topology table(only one bit to indicate throttling or not). By introducing the characteristic of throttling, we only need N*M*$\log_2(K)$ bits. There are some examples shown in Table 5-1.

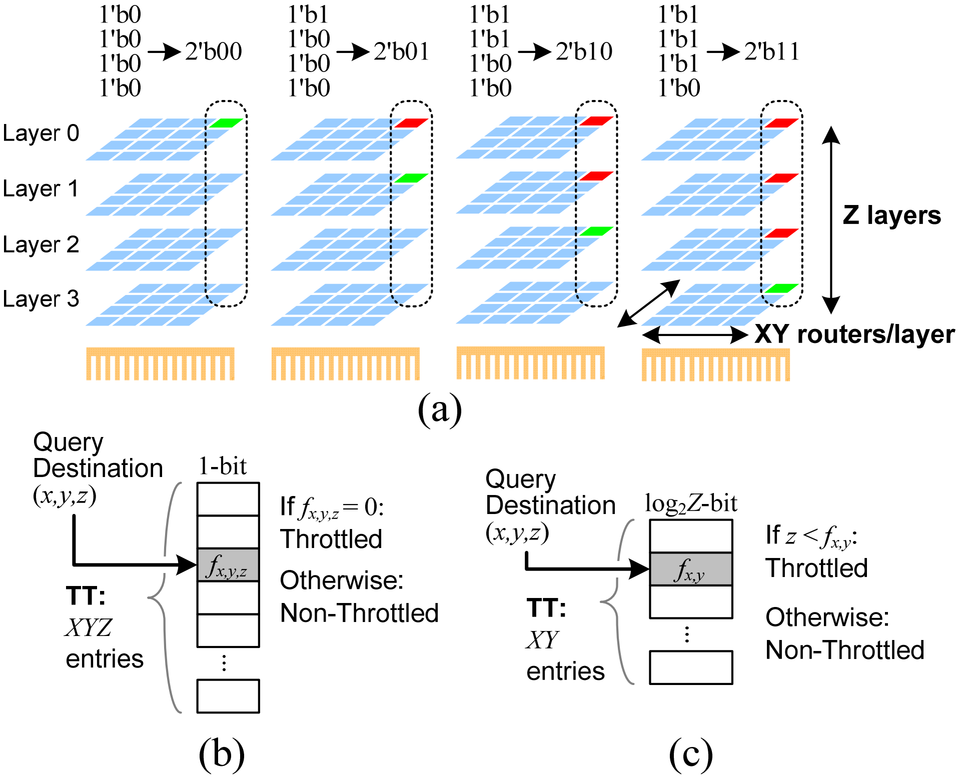


Fig. 5-4 (a) Reduce the size of topology table by storing the first non-throttled layer for each XY location. (b) Direct implementation. (c) Implementation with proposed TT reduction technique.

Table 5-1 Topology table comparison.

For NxMxK 3D NoC	Original topology table (N*M*K)	Improved topology table (N*M*(log$_2$k))	Reduction
8x8x2 3D NoC	128	64	50%
8x8x4 3D NoC	256	128	50%
8x8x8 3D NoC	512	192	62.5%

The other memory reduction is about routing mode memory. For routing mode memory (RMM), TLAR only requires bits to store the routing modes for all destinations. We use the example in Fig. 5-5(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. 5-5(b). Therefore, we originally need N*M*K bits to store topology table for an N*M*K 3D NoC. However, we now only need N*M bits to store routing mode decision by introducing the memory reduction technique. There are some examples shown in Table 5-2.

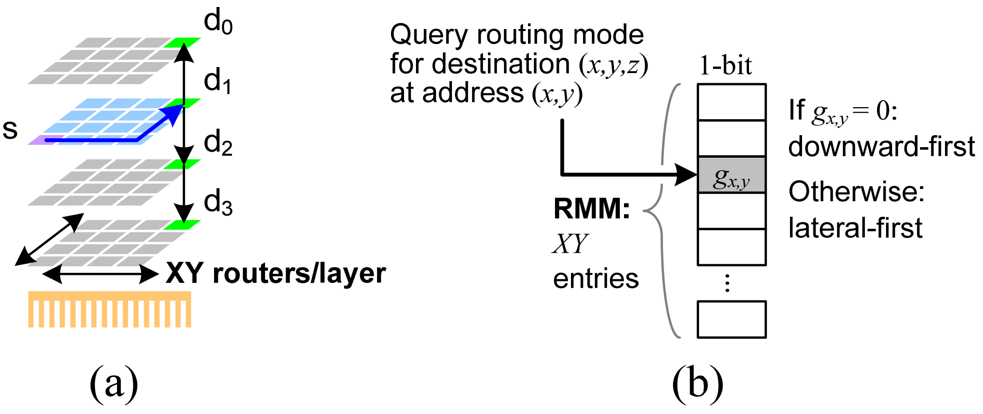


Fig. 5-5 (a) The source destination pairs (s,d_0), (s,d_1), (s,d_2), (s,d_3) has the same source layer for lateral-first path, so their routing modes are identical. (b) For an X-Y-Z network, the size of RMM is XY bits.

Table 5-2 Routing mode memory comparison.

For NxMxK 3D NoC	Original routing mode memory (N*M*K)	Improved routing mode memory (N*M)	Reduction
8x8x2 3D NoC	128	64	50%
8x8x4 3D NoC	256	64	75%
8x8x8 3D NoC	512	64	87.5%

5.3 Dual-Mode Router Design

The proposed dual-mode router is shown in Fig. 6. The router is based on the wormhole flow control, 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. In comparison to the 2D router, the 3D router requires extra two physical channels 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 complexity of routing logic and arbitration logics also increase due to the additional channels. The overhead of the proposed TLAR is in RC. To support the functionality of downward routing and XY routing, RC has to be dual-mode, as shown in Fig. 5-6. As 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, as shown in Fig. 5-7. 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 Chapter 3.

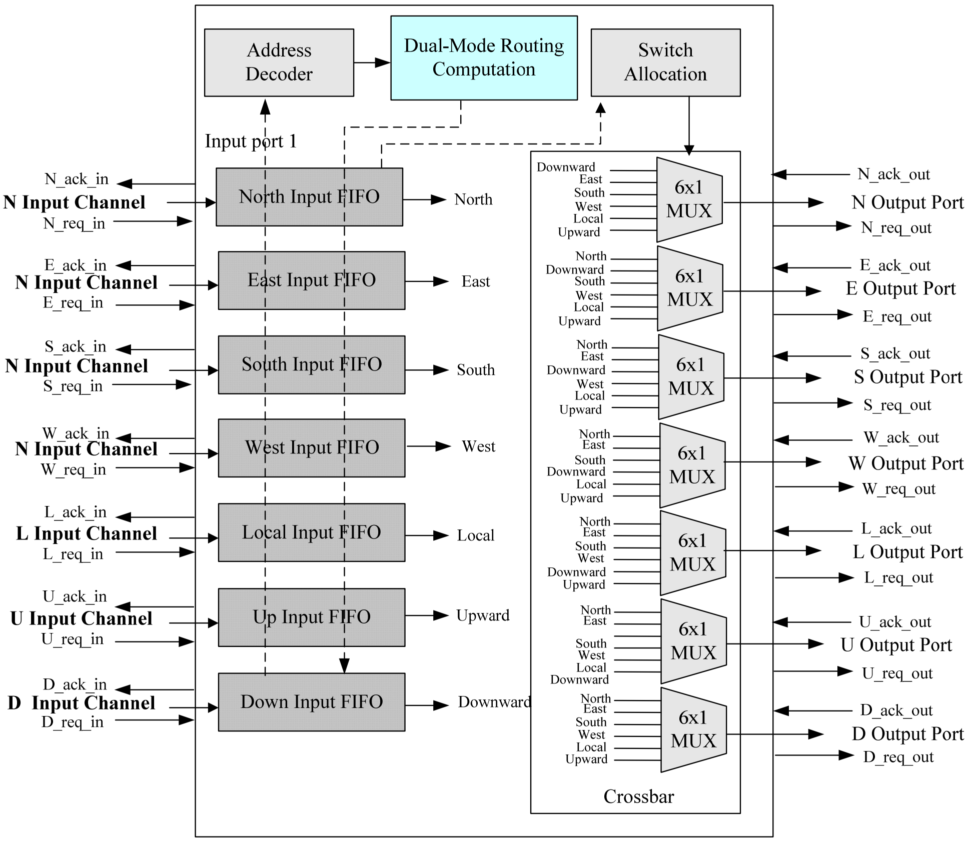


Fig. 5-6 Our proposed 3D router architecture.

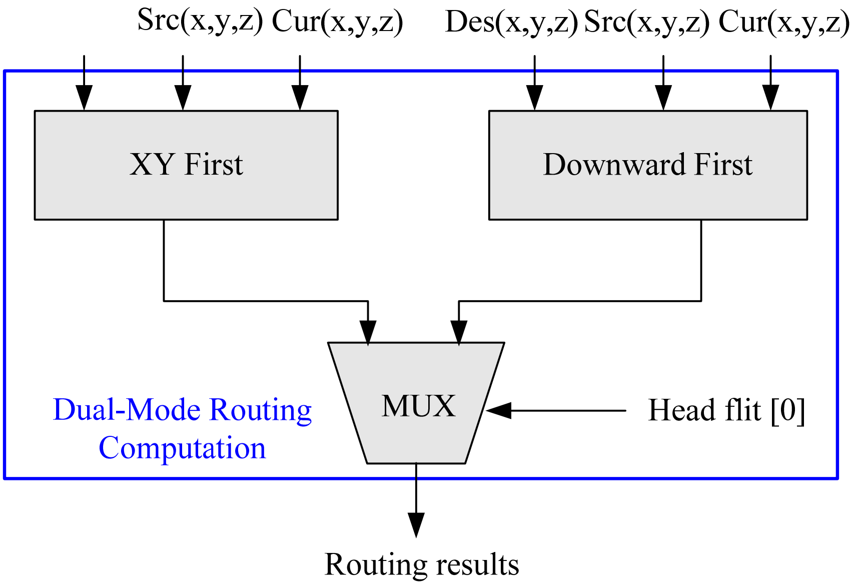


Fig. 5-7 Architecture of dual-mode routing computation.

5.4 Implementation Results and Comparison

Here we present the synthesis results of our proposed network interface and router design. Table 5-3 shows the design parameters. We implement 16 flits, which we reference similar to Intel 80 tiles [30] with no virtual channel. We have implemented a 34-bit bandwidth router using 0.13 μm CMOS standard cells. The router has local port and other ports providing 7 input and 7 output network ports. Each flit contains 34 bits.

The NI and the router are designed for an 8x8x4 3D NoC. The network interface contains two 16-flit queues for transmitter and receiver (Tx/Rx), a topology table (TT), a routing mode memory (RMM), and necessary control logic (CL). The post synthesis simulation shows the NI and the router is able to operate at 360 MHz. In topology table, the original algorithm needs 8*8*4 = 256 bits. After applying the improved algorithm, we only 8*8*2 = 128 bits, which save 50% of original bits. For routing mode memory, we originally need 8*8*4 = 256 bits. Then we now only need 8*8 = 64 bits, which we save 75 % of original bits.

Table 5-3 Design parameters.

Technology	UMC 90nm
Clock period/frequency	2.4 ns/420 MHz
Topology Table Size	8x8x4 3D Mesh
Number of ports per router	7
Buffer depth per router port	16
Flit size	32 bits + 2 bits(control bits)
Message Queue in Tx	32 bits*16 flits = 512 bits
Packet Queue in Rx	34 bits *16 flits = 544 bits

In network interface and router, we should consider how much area overhead of transport-layer assisted routing (TLAR) and memory overhead is decreased by improving algorithm. First, as shown in Fig. 5-8, we show effectiveness of the proposed table and memory reduction techniques by the synthesis result of the TLAR NI. The area of traditional NI + router is 36,832μm^2. If we directly implement TLAR NI + router, the area is 83,094μm^2, which the area overhead is 125.6%. Applying the proposed TT reduction technique, the area of topology table is decreased from 23,180μm^2 to 11,222μm^2, and the total area of the NI is decreased to 71,496μm^2, which is reduced by 31.5%. Applying the proposed RMM reduction technique, the area of routing mode memory is decreased from 21,650μm^2 to 5,286μm^2. The combination of the proposed TT reduction and RMM reduction can reduce the total area of NI to 55,972μm^2, which is only 52% area overhead.

Second, we show the synthesis result of the traditional NI + router and the synthesis result of the proposed TLAR NI + router in Table 5-4. 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 5-4, 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 %.

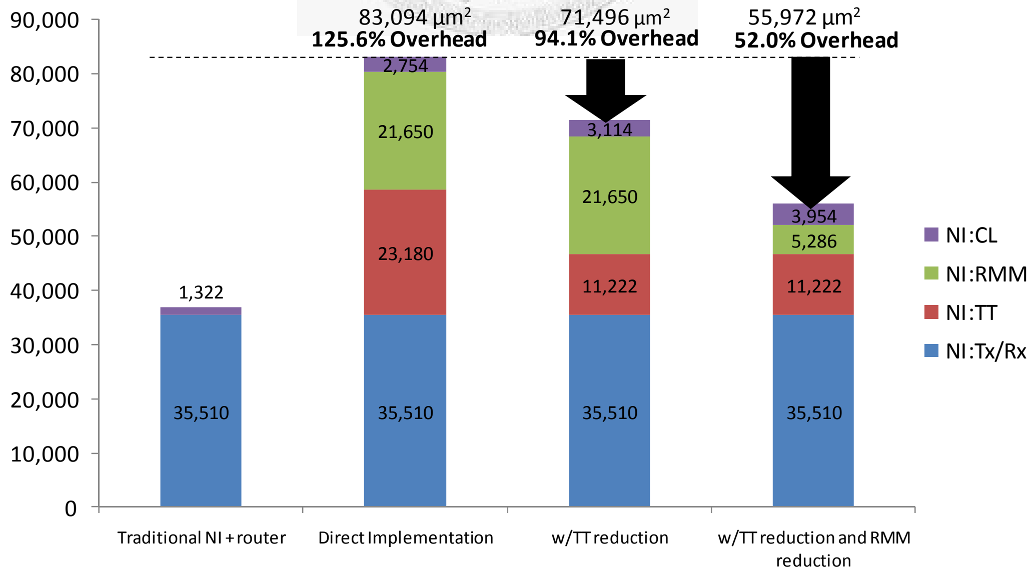


Fig. 5-8 Synthesis area (μm^2) of TLAR NI.

Table 5-4 Synthesis Area of NI and Router (μm^2).

	Traditional NI + router	Direct Implementation of TLAR NI + Router	TLAR NI + Router w/ TT and RMM reduction
NI: Tx and Rx	35,510	35,510	35,510
NI: TT	N/A	23,180	11,222
NI: RMM	N/A	21,650	5,286
NI: CL	1,322	2,754	3,954
Router	95,527	96,226	96,226
Total (NI + Router)	132,359	179,320(+**35.5**%)	152,198(+**14.9**%)

5.5 Summary

In this chapter, we implement TLAR scheme by implementing network interface and router. The TLAR network interface has additional topology table and routing mode memory than traditional network interface, and we proposed memory reduction techniques to reduce area overhead. For router design, we propose dual-nmode router architecture, and the area overhead is little. Finally, by using memory reduction techniques, the area overhead is 14.9% comparison to traditional NI + router.

Chapter 6 Conclusion

In this thesis, we introduce a new problem by defining the non-stationary irregular mesh (NSI-mesh). To solve NSI-mesh problem, we need reactive routing to detour throttled routers. To improve the performance of reactive routing, we propose a new transport-layer assisted routing (TLAR) and three routing algorithms based on it. Our NoC system is divided to normal work step and reconfiguration step. The computation overhead of checking for TLAR is negligible and can be easily done in the reconfiguration state. The storage overhead depends on the number of destination of each source tile, but only one bit for topology table and one bit for routing mode memory are required for each destination. From our experiments, the proposed TLAR can effectively balance the vertical load distribution. The throughput is improved up to 90%. The proposed TLAR can averagely improve the sustainable throughput from 48% to 85.5%. Finally, we implement TLAR NI + router. The area overhead of direct implementation of TLAR is 35.5%. With the proposed table and memory reduction techniques, the area overhead of TLAR can be reduced to 14.9%, which is relative small in comparison with the throughput improvement.

References

[1] ITRS, International Technology Roadmap for Semiconductors, [Online]. Available: http://public.itrs.net.

[2] J. A. Davis, R. Venkatesan, A. Kaloyeros, M. Beylansky, S. J. Souri, K. Banerjee, K. C. Saraswat, A. Rahman, R. Reif and D. Meindl, "Interconnect limits on Gigascale Integration (GSI) in the 21st century," in *Proc. IEEE*, vol. 89, pp. 305-324, Mar. 2001.

[3] R. Ho, K. W. Mai, and M. A. Horowitz, "The future of wires," in *Proc. IEEE*, vol. 89, pp. 490-504, April. 2001.

[4] D. Sylvester and K. Keutzer, "A global wiring paradigm for deep submicron design," *IEEE Trans. CAD/ICAS*, vol. 19, pp. 242-252, Feb. 2000.

[5] L. Benini and G. De Micheli, "Networks on chips: A new SoC paradigm," *IEEE Computer*, vol. 35, pp. 70–78, Jan. 2002.

[6] A. W. Topol, D. C. La Tulipe, Jr., L. Shi, D. J. Frank, K. Bernstein, S. E. Steen, A. Kumar, G. U. Singco, A. M. Young, K. W. Guarini and M. Ieong, "Three-dimensional integrated circuits," *IBM J. Res. Develop.*, vol. 50, no. 4/5, pp. 491-506, 2006.

[7] B. Black, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. H. Loh, D. McCaule, P. Morrow, D. W. Nelson, D. Pantuso, P. Reed, J. Rupley, S. Shankar, J. Shen and C. Webb, "Die stacking (3D) microarchitecture," in *Proc. Int. Symp. Microarchitecture*, pp. 469-479, Dec. 2006.

[8] Samsung. [Online]. Available: http://www.samsung.com/

[9] Tezzaron. [Online]. Available: http://www.tezzaron.com/technology/FaStack.htm

[10] W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. D. Franzon, "Demystifying 3D ICs-- the pros and cons of going vertical," *IEEE Design & Test of Computers*, pp.498-510, 2005.

[11] R. M. Lea, I. P. Jalowiecki, D. K. Boughton, J. S. Yamaguchi, A. A. Pepe, V. H. Ozguz, and J. C. Carson, "A 3D stacked chip packaging solution for miniaturized massively parallel processing," *IEEE Trans. Advanced Packaging*, vol. 22, no. 3, pp. 424-432, Aug. 1999.

[12] R. J. Gutmann, J. Q. Lu, Y. Kwon, J. F. McDonald and T. S. Cale, "Three-dimensional (3D) ICs: A technology platform for integrated systems and opportunities for new polymeric adhesives," in *Proc. Int. IEEE Conf. Polymers and Adhesives in Microelectronics and Photonics*, pp. 173-180, 2001.

[13] A. Y. Zeng, J. J. L¨u, K. Rose, and R. J. Gutmann, "First-order performance prediction of cache memory with wafer-level 3D integration," *IEEE Design & Test of Computers*, pp. 548–555, 2005.

[14] I. Loi, S. Mitra, T. H. Lee, S. Fujitaand L. Benini, "A low-overhead fault tolerance scheme for TSV-based 3D Network on Chip links," in *Proc. IEEE/ACM Int. Conference on Computer-Aided Design*, pp. 598-602, 2008.

[15] V. Pavlidis, E. Friedman, "3D topologies for Networks-on-Chip," *IEEE Trans. VLSI*, pp. 1081-1090, 2007.

[16] J. Joyner, P. Zarkesh-Ha, and J. Meindl, "A stochastic global net-length distribution for a three-dimensional System-on-Chip (3D-SoC)," *in Proc. IEEE Int. ASIC/SOC Conference,* pp. 147-151, Sept. 2001.

[17] D. Park, S. Eachempati, R. Das, A. Mishra, Y. Xie, N. Vijaykrishnan, and C. Das, "MIRA: A multi-layered on-chip interconnect router architecture," in *Proc. IEEE Int. Symp. Computer Architecture (ISCA)*, pp. 251-261, 2008.

[18] C. Zhu, Z. Gu, L. Shang, R. P. Dick, and R. Joseph, "Three-dimensional chip-

multiprocessor run-time thermal management," *IEEE Trans. Computer Aided Design of Integrated Circuits and Systems*, vol. 27, no. 8, pp. 1479-1492, Aug. 2008.

[19] 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,* pp. 67-78, 2004.

[20] J. Torresola, Chia-Pin Chiu, G. Chrysler, D. Grannes, R. Mahajan, R. Prasher and A. Watwe, "Density factor approach to representing impact of die power maps on thermal management." *IEEE Trans. Advanced Packaging*. vol. 28, no. 4, pp. 659-664. Nov. 2005.

[21] C. Addo-Quaye, "Thermal-aware mapping and placement for 3-D NoC designs," in *Proc. IEEE Int. SOC Conference*, pp.25-28, Sept. 2005.

[22] 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," to appear in *Int. Symp. Networks-on-Chip, May* 2010.

[23] J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, N. Vijaykrishnan , M. Yousif, and C. Das, " A novel dimensionally-decomposed router for on-chip communication in 3D architectures," in *Proc. IEEE Int. Symp. Computer Architecture (ISCA)*, pp. 138-149, 2007.

[24] G. M. Chiu, "The odd-even turn model for adaptive routing," *IEEE Trans. Parallel and Distributed Systems*, vol. 11, no. 7, pp. 729-738, July, 2000.

[25] G. Ascia, V. Catania, M. Palesi and D. Patti, "Neighbors-on-Path: A new selection strategy for on-chip networks," in *Proc. IEEE/ACM/IFIP Workshop on Embedded Systems For Real Time Multimedia*, Oct. 2006.

[26] P. Gratz, B. Grot, and S. W. Keckler, "Regional congestion awareness for load balance in networks-on-chip," in *Proc. IEEE Symp. High Performance Computer Architecture (HPCA)*, pp.203-214, Feb. 2008.

[27] J. Hu and R. Marculescu, "Application-specific buffer space allocation for networks-on-chip router design," in *Proc. IEEE/ACM Int. Conf. on Computer-*

Aided Design, Nov. 2004.

[28] Noxim: Network-on-Chip Simulator [On-line]. Available: http://sourceforge.net/projects/noxim

[29] W. Huang, K. Sankaranarayanan, R. J. Ribando, M. R. Stan, and K. Skadron, “An improved HotSpot block-based thermal model with granularity considerations,” in *Workshop on Duplicating, Deconstructing, and Debunking (WDDD), in conjunction with Intl. Symp. on Computer Architecture (ISCA)*, June 2007.

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

[31] C. J. Glass and L. M. Ni, “The turn model for adaptive routing,” *J. ACM*, vol. 41, no. 5, pp. 874-902, Sep. 1994.

[32] 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,” to appear in *International Symposium on VLSI Design, Automation & Test (VLSI-DAT)*, Apr. 2010.

[33] Benini, L., and De Micheli, G.: ‘Networks on chips: A new SoC paradigm’, *IEEE Comput.*, 2002, 35, (1), pp. 70–80

[34] 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. Computers, vol. 52*, pp. 1156–1168, Sept. 2008.

[35] Shu-Yen Lin, Tzu-Chu Yin, Hao-Yu Wang, and An-Yeu Wu “Traffic-and Thermal-Aware Routing for Throttled Three-Dimensional Network-on-Chip Systems,” i*n Proc. IEEE Int. Symp. VLSI-DAT-2011*, Hsinchu, TAIWAN, April 2011.