

# Topology-Aware Adaptive Routing for Nonstationary Irregular Mesh in Throttled 3D NoC Systems

Kun-Chih Chen, *Student Member, IEEE*, Shu-Yen Lin, *Member, IEEE*, Hui-Shun Hung, and An-Yeu (Andy) Wu, *Senior Member, IEEE*

**Abstract**—Three-dimensional network-on-chip (3D NoC) has been proposed to solve the complex on-chip communication issues in future 3D multicore systems. However, the thermal problems of 3D NoC are more serious than 2D NoC due to chip stacking. To keep the temperature below a certain thermal limit, the thermal emergent routers are usually throttled. Then, the topology of 3D NoC becomes a *Nonstationary Irregular Mesh (NSI-Mesh)*. To ensure the successful packet delivery in the *NSI-Mesh*, some routing algorithms had been proposed in the previous works. However, the network still suffers from extremely traffic imbalance among lateral and vertical logic layer. In this paper, we propose a Topology Aware Adaptive Routing (*TAAAR*) to balance the traffic load for *NSI-Mesh* in 3D NoC. *TAAAR* has three routing modes, which can be dynamically adjusted based on the topology status of the routing path. In addition to increasing routing flexibility, the *TAAAR* also increases both vertical and lateral path diversity to balance the traffic load. Compared with the related adaptive routing methods, the experimental results show that the proposed *TAAAR* can reduce 19 to 295 percent traffic loads in the bottom logic layer and improve around 7.7 to 380 percent network throughput. According to our proposed VLSI architecture, the *TAAAR* only needs less than 24.8 percent hardware overhead compared with the previous works.

**Index Terms**—Network-on-chip, 3D IC, 3D NoC, transport layer assisted routing

## 1 INTRODUCTION

As the complexity of system-on-chip (SoC) grows with Moore's law, the considerations of interconnection complexity and wire delay become more crucial for chip multiprocessor (CMP) systems. The network-on-chip (NoC) has been proposed as a novel and feasible solution for efficient interconnections [1], [2], [3]. Recently, the three-dimensional integrated circuit (3D IC) technologies are promising to reduce the interconnect delays by die stacking [4], [5]. Hence, the 3D NoC-based CMP is expected to achieve higher performance with lower connection cost and data transmission power consumption. However, the cooling efficiency of interlayer is different due to die stacking. Besides, the routers have been shown as the sources of thermal hotspots due to the higher switching activity [7], [8], which leads to severer heat problems in 3D NoC-based CMP systems [9].

To keep the system temperature below a certain thermal limit, runtime thermal management (RTM) is required. The simplest RTM is to shut down whole network even if there is only one thermal emergent node. However, the performance impact of this kind of Global Throttling (GT) scheme is very serious [6]. In [7], Shang et al. proposed a

distributive throttling (DT) scheme, *ThermalHerd*, to regulate the network temperature. Because *ThermalHerd* distributively throttles the thermal emergent nodes, the performance impact is smaller than GT. However, this kind of DT scheme needs much longer cooling time because of the long heat dissipation path in 3D NoC. By considering both cooling speed and performance, Chao et al. proposed a *Thermal-Aware Vertical Throttling (TAVT)* scheme in 3D NoC [10], [17]. The control granularity of the TAVT is a pillar, which consists of the nodes with the same XY addresses. Based on the level of thermal emergency, TAVT determines different throttling states (i.e., the number of nodes in a throttling pillar), as shown in Fig. 1a. Because the heat dissipation capability of the bottom logic layer is good enough to avoid overheat, the bottom logic layer is always set as a nonthrottling layer (i.e., the logic layer contains no throttled node) [10], [17]. Due to the better cooling capability and smaller performance impact than DT, the TAVT is adopted as the RTM policy of this work. However, the TAVT makes the network topology become time-varying *Nonstationary Irregular Mesh (NSI-Mesh)* [11], [17], as shown in Fig. 1b. The routers in the throttled tiles cannot transmit any packet until the temperature becomes thermal safety. The *NSI-Mesh* causes many packets being blocked in the network for a long time and results in serious performance degradation.

To ensure the successful packet delivery in *NSI-Mesh*, many routing algorithms were proposed by using the spatial information [10], [12], [13], which make the packets detour the inactive nodes through the symmetric structure provided by NoC. In [12], [13], the fault-tolerant (FT) routings are applied to solve the routing problem in the

• The authors are with the Institute of Electrics Engineering, National Taiwan University, Taipei, Taiwan.  
E-mail: {kunchih, linyan, hfsken}@access.ee.ntu.edu.tw, andywu@cc.ee.ntu.edu.tw.

Manuscript received 9 May 2012; revised 22 Sept. 2012; accepted 25 Sept. 2012; published online 4 Oct. 2012.

Recommended for acceptance by M.E. Acacio.

For information on obtaining reprints of this article, please send e-mail to: tpsds@computer.org, and reference IEEECS Log Number TPDS-2012-05-0456. Digital Object Identifier no. 10.1109/TPDS.2012.291.

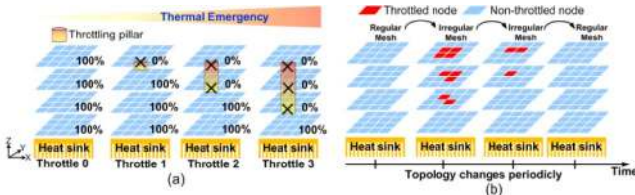


Fig. 1. (a) The throttling states of TAVT depends on the thermal emergency [10], [17], and (b) TAVT results in NSI-Mesh [11], [17].

NSI-Mesh. However, applying the FT routings directly may result in unbalanced traffic in NSI-Mesh [14]. For 3D NoC, the downward routing was proposed to pass through the bottom layer to detour the throttled nodes, as shown in Fig. 3a [10]. However, the downward routing leads to very heavy traffic congestion in the bottom logic layer, as demonstrated in Fig. 2a. To reduce the problem of traffic congestion, some routing algorithms were proposed to make the packets early detour the inactive nodes by using the temporal information in advance. In [16], Lin et al. proposed to use the buffer information and broadcast throttling information, which records the location of the throttled nodes, to early detour the throttled nodes. In [11] and [17], Chao et al. proposed the Transport Layer Assisted Routing (TLAR) scheme, which considers the information in both transport layer and network layer to deliver packets. It has better network throughput than the previous works, as shown in Fig. 2b. However, the TLAR still suffers from the heavy traffic congestion in the bottom logic layer and extremely traffic imbalance between vertical logic layers. The reason is that this routing algorithm has less lateral and vertical path diversity, as shown in Fig. 3b.

To solve the traffic imbalance in 3D NSI-Mesh and increase the routing flexibility, the contributions of this paper are summarized as follows:

1. *A Topology Aware Adaptive Routing (TAAR) algorithm.* Because of the larger path diversity provided by the proposed TAAR (see Fig. 3c), the traffic becomes more balanced in the NSI-Mesh, as illustrated in Fig. 2c. Compared with the previous related works, the proposed TAAR can improve around 7.7 to 380 percent network throughput. The TAAR consists of the following two techniques:
  - a. *Intralayer routing.* A novel cascaded routing, which can dynamically adjust the routing mode based on the network topology, is proposed to detour the throttled nodes. By this way, the packets can be properly routed at the same logic

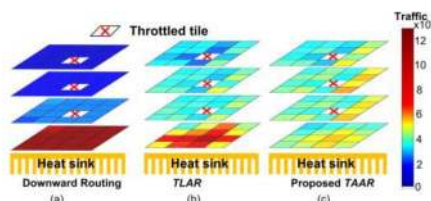


Fig. 2. The traffic distribution of (a) Downward routing, (b) TLAR, and (c) the proposed TAAR.

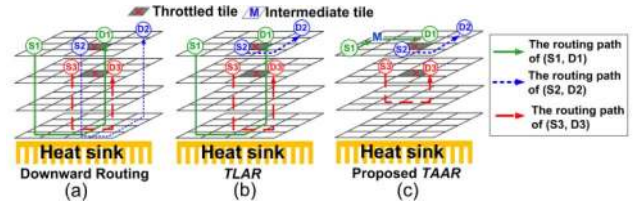


Fig. 3. The routing path of (a) Downward routing, (b) TLAR, and (c) the proposed TAAR.

layer, which reduces the traffic congestion in the bottom logic layer. In addition, we propose a queuing model to analyze the performance of this cascaded routing.

- b. *Interlayer routing.* The TAAR adopts multiple downward layer selections for vertical traffic balance. Therefore, the downward routing path is not restricted to the bottom logic layer. Based on the lateral traffic load, a queuing analysis is also proposed to derive a suitable downward ratio assignment.

2. *VLSI Architecture Design of TAAR.* We propose a corresponding architecture of TAAR. To reduce the area overhead, we use the information alignment of a specific region to reduce the table size. Compared with the previous related works, the proposed TAAR has 16.2 to 24.8 percent area overhead.

The rest of this paper is organized as follows: In Section 2, we briefly review some related routing algorithms for 3D NSI-Mesh. In Section 3, the proposed TAAR is introduced. In Section 4, the experiments and comparisons are discussed. In Section 5, we describe our proposed TAAR architecture implementation and memory reduction techniques. Finally, the conclusion is shown in Section 6.

## 2 BACKGROUND AND RELATED WORKS

The topology of the thermal-aware 3D NoC transforms while some thermal emergent routers are throttled and results in NSI-Mesh, which results in rapid performance degradation. To solve the routing problem in NSI-Mesh, the routing algorithms use spatial information [10], [12], [13] or temporal information [11], [16], [17] to detour these inactive nodes. The routing algorithms, which use spatial information, make the packets detour the inactive nodes through the symmetric structure provided by NoC. On the other hand, the routing algorithms were proposed to make the packets early detour the inactive nodes by using the temporal information in advance. These approaches are reviewed in Sections 2.1 and 2.2, respectively.

### 2.1 Routings for NSI-Mesh with Spatial Information

#### 2.1.1 Thermal-Aware Downward Routing [10]

The bottom logic layer of a 3D NoC system is set as a nonthrottling layer due to the highest thermal conductance toward heat sink. In [10], Chao et al. applied downward routing to deliver the packet through the bottom logic layer while some routers are throttled in the upper logic layer. However, this approach results in heavy traffic load in the bottom logic layer (as shown in Fig. 2a), which results in rapid network performance degradation.

### 2.1.2 Fault-Tolerant Routing [12], [13]

The fault-tolerant (FT) routings can be applied to solve the routing problems in the throttled NoC systems, because the faulty nodes are similar to the throttled nodes in the same location. In [12], Wu proposed *Extended XY* routing algorithm based on the XY routing and odd-even (OE) turn model [18]. In [13], Chen and Chiu proposed a FT routing for irregular mesh, *Chen and Chiu's*. In [36], Flich and Duato proposed a Logic-Based Distributed Routing (LBDR) to detour the faulty block without routing table supporting. However, the FT routing is not feasible to directly apply in the thermal-aware 3D NoC systems due to the following two problems:

1. Due to the major concern for FT routings, the traffic loads on the boundaries of faulty blocks are heavy and unbalanced [14].
2. Before packet delivery, the location of the faulty blocks must be marked. However, to facilitate the FT routing, the faulty blocks may contain some non-faulty nodes, which leads to low-channel utilization and significant performance impact.

### 2.1.3 Traffic-Load Balanced Cascaded Routing [20], [21], [22]

Because of the larger lateral path diversity [20], [21], [22], the cascaded routing algorithm may be applied to detour the throttled nodes. However, the conventional cascaded routing algorithms are not feasible to directly apply in the thermal-aware 3D NoC systems due to the following three problems:

1. These approaches were proposed to solve the routing problem in a regular topology.
2. Deadlock cannot be avoided without virtual channel (VC), which may result in significant area overhead [31].
3. The identical routing mode in each routing phase may restrict the routing path diversity.

## 2.2 Routings for *NSI-Mesh* with Temporal Information

### 2.2.1 Traffic- and Thermal-Aware Routing (TTAR) [16]

In [16], Lin et al. proposed TTAR, which adopts OE turn model [18] and one-dimensional Regional Congestion Awareness (RCA) selection criterion [19]. The TTAR can detour the throttled nodes and route properly at the same logic layer based on the propagated throttling information. However, the following two problems of TTAR cause the heavy traffic congestion in the bottom logic layer:

1. The OE turn model restricts the routing paths, which leads to heavy traffic load in the specific columns [14].
2. The vertical path diversity is too small (i.e., only downward route to the bottom logic layer). Hence, the vertical traffic load becomes very unbalanced.

### 2.2.2 Transport Layer Assisted Routing [11], [17]

Chao et al. classified the unsuccessfully deliveries in *NSI-Mesh* into four problems [11], [17]: 1) the source node is fully

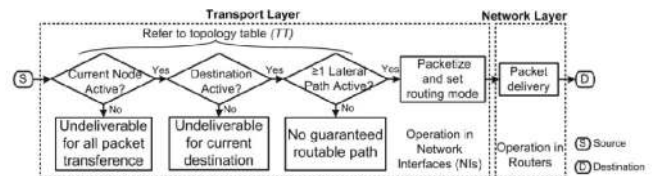


Fig. 4. Operation flowchart of a transport layer and a network layer.

throttled, 2) the destination node is fully throttled, 3) any one of the router on the routing path is fully throttled, and 4) the channel on the routing path is fully blocked by other long-term congested packets. For successful packet delivery, the authors proposed to use the information of both transport layer and network layer to ensure there is at least one routable path from source to destination. If the source layer (i.e., the logic layer where the source node locates at) is routable, TLAR will decode to use lateral routing first. Otherwise, the downward routing is used instead. To realize this scheme, the authors proposed three routing algorithms:

- *Downward Lateral Deterministic Routing (DLDR)*. It uses XY routing as the lateral routing algorithm.
- *Downward Lateral Adaptive Routing (DLAR)*. It uses adaptive routing as the lateral routing algorithm.
- *Downward Lateral Adaptive-Deterministic Routing (DLADR)*. To increase the path diversity, the adaptive routing and XY routing are combined for lateral routing.

Although TLAR has better performance than previous works, it still suffers from heavy traffic load from other logic layers, as shown in Fig. 2b. To solve the problem of traffic congestion around the throttled nodes [10], [12], [13] and less path diversity [11], [16], [17], the TAAR is proposed in this paper, which is introduced in Section 3.

## 3 PROPOSED TOPOLOGY-AWARE ADAPTIVE ROUTING (TAAR)

To prevent the packet from being blocked in the throttled node, the throttling information, which is stored in the Topology Table (TT), is checked before sending the packet to the network, as shown in Fig. 4. The throttling information is updated during each change of topology. To realize the scheme to update TT, the previous proposed RTM framework is adopted [11], [17], as shown in Fig. 5. In reconfiguration stage, based on the temperature sensing results provided by the distributed thermal sensor, the TT is updated by the propagating throttling information along the X-, Y-, and Z-dimension sequentially. For a network which operates at 1 GHz, and the total period of temperature-traffic control is 10 ms, Chao et al. showed that the total reconfiguration stage only needs less than 0.1 percent of the total period, which has negligible timing overhead [11], [17]. By following the RTM framework in Fig. 5, the *Topology Aware Adaptive Routing* is proposed to balance the traffic of whole 3D NoC and introduced in this section.

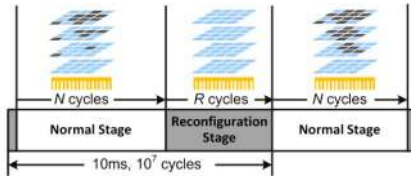


Fig. 5. The framework of the adopted *RTM* [11], [17].

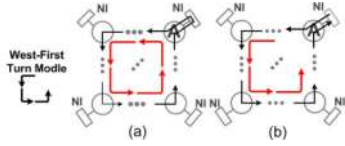


Fig. 6. (a) Deadlock is occurred without store-and-forward policy, and (b) Deadlock is prevented with store-and-forward policy.

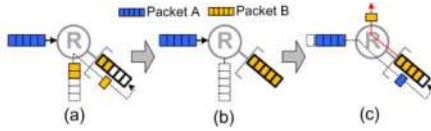


Fig. 7. (a) Packet B is waiting for Packet A, (b) Packet A is prepared to be retransmitted, and (c) Packet B start to enter the temporal storage.

### 3.1 Lateral Cascaded Routing: Topology-Aware Multiple Routing Adjustment (*TAMRA*)

#### 3.1.1 Topology-Aware Multiple Routing Adjustment

For lateral path diversity, the *TAMRA* delivers the packets to the destinations through cascaded routing. Because of the channel dependency, the *TAMRA* may lead to deadlock, as shown in Fig. 6a. The *VC* was proposed to avoid deadlock in [20], [21], [22]. Though the system performance is improved by using *VC*, the large buffer requirement requires large area cost [31]. In addition to *VC*, the restricted routing function can perform deadlock avoidance with low area cost, which is the major concern for hardware implementation. To consider the cost issue, the Store-and-Forward (*SAF*) policy is adopted along with the wormhole flow control, as shown in Fig. 6b. Because the violated turn is cut by the *SAF* policy, the deadlock is avoided. To reduce the latency overhead caused by *SAF* policy, the *TAMRA* adopts two phases cascaded routing (i.e., one intermediate node). By following the transport layer assistance (as shown in Fig. 4), the rerouted packets (i.e., the packet arrive at its destination through cascaded routing) are temporally stored in network interface (*NI*).

The switching contention is happened, if there are two more packets request the same output direction. For the rerouted packets, the switching contention results in extra latency overhead, as shown in Fig. 7. In the worst case, if each router can transmit one flit per cycle with fair arbitration, the extra latency overhead of the rerouted packet is the same as (the biggest packet size)  $\times$  3 cycles while the rerouted packet is the last one to get the grant. Compared with the blocking time (i.e., at least  $10^7$  cycles), which the packets stick at the throttled node, the latency overhead of the *SAF* policy is negligible.

To increase the lateral path diversity, the adaptive routing is a practical way. However, the computational complexity of checking all adaptive routing paths in *NSI-Mesh* is very large. For a minimal adaptive routing with  $V$

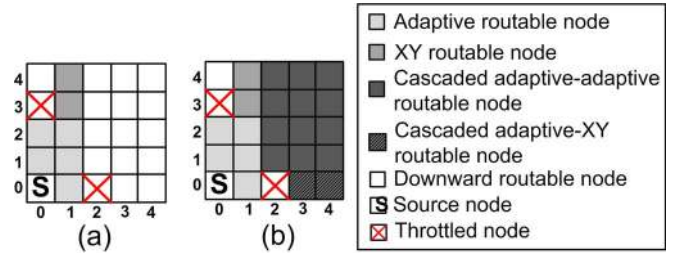


Fig. 8. (a) Reachable destinations in *TLAR*, and (b) Reachable destinations in the proposed *TAMRA*.

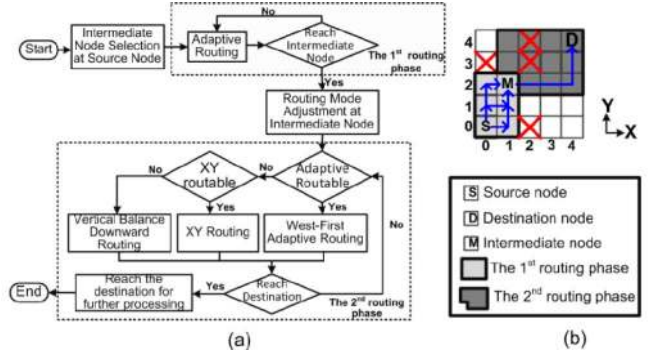


Fig. 9. (a) Operation flow of *TAMRA*, and (b) An example.

and  $H$  diversity in  $x$ - and  $y$ -direction, respectively, there are  $\frac{(V+H)!}{V!H!}$  possible paths should be checked, and the computational complexity is  $O(4^N)$ . To reduce the computational complexity, we only check the throttled nodes in the minimal-path region. For the minimal-path region containing one or more throttled nodes, the packets may be blocked, and the routing region is set as nonguaranteed adaptive routing region. In this case, the *XY* routing path will be checked to increase the routable path through lateral routing, which reduces the computational complexity to  $O(N)$ . If there is no routable lateral routing path, the downward routing will be considered instead. Therefore, in this work, the *TAMRA* can select three routing modes: west-first adaptive routing, *XY* routing, and downward routing. As the proof in [17], this kind of hybrid lateral routing algorithm is deadlock-free.

Based on the topology information, the proposed *TAMRA* can dynamically adjust the applied routing mode of each packet, which can significantly increase the lateral path diversity. Fig. 8 is an example of *TAMRA* in the source layer. Because the lateral path diversity of *TLAR* is not enough, there are many node cannot be achieved without downward routing, as shown in Fig. 8a. If we set the adaptive routing region of *TLAR* as the first routing phase region of *TAMRA* (i.e., the light gray area in Fig. 8b), there are more nodes can be achieved through cascaded routing. The flowchart of *TAMRA* is shown in Fig. 9a, and Fig. 9b is an example. In the first routing phase, the adaptive routing is adopted to deliver the rerouted packet from source node  $S$  to the predecided intermediate node  $M$ . The routing mode of the packet will be adjusted depends on the topology situation of the second routing phase at node  $M$ . The updated routing mode (i.e., *XY* routing in this case) will be adopted in the second routing phase until reaching the destination node  $D$ .

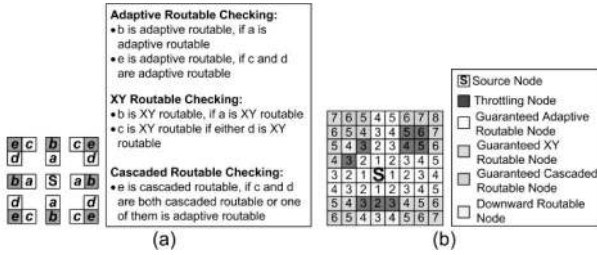


Fig. 10. (a) Dependency of routability checking of TAAR, and (b) An example of routing mode checking in TAAR.

### 3.1.2 Decision of Intermediate Node and Routing Mode

In addition to the  $TT$  updating, the checking of routing mode and selection of intermediate node can be done during the reconfiguration stage of  $RTM$  [17]. The routing mode and the intermediate node can be decided based on the updated information in  $TT$ . The decision of routing mode and intermediate node will be stored in routing mode memory ( $RMM$ ) and intermediate node memory ( $INM$ ), respectively.

Based on the dependency of routability illustrated in Fig. 10a, the three routing modes provided by TAAR can be decided by using the incremental checking flow as the sequence number shown in Fig. 10b. If the packet can be directly routed without cascaded routing, the intermediate node will be equal to the destination, which reduces the unnecessary cascaded routing. If the destination is lateral nonroutable through adaptive or XY routing, cascaded routing or downward routing will be considered.

For those cascaded routable nodes, the selection of intermediate node is the critical issue. We use Fig. 9b to illustrate an example. If the intermediate node is  $(0, 4)$ , the packet will reach the destination through downward routing. However, the destination is routable though cascaded adaptive-XY routing while the intermediate node is  $(1, 2)$ . It is difficult to find an optimal intermediate node. If  $X_d$  and  $Y_d$  are the distances between source and destination in  $x$ -direction and  $y$ -direction, respectively;  $V$  and  $H$  are the distances between source and intermediate node in  $x$ -direction and  $y$ -direction, respectively, there are

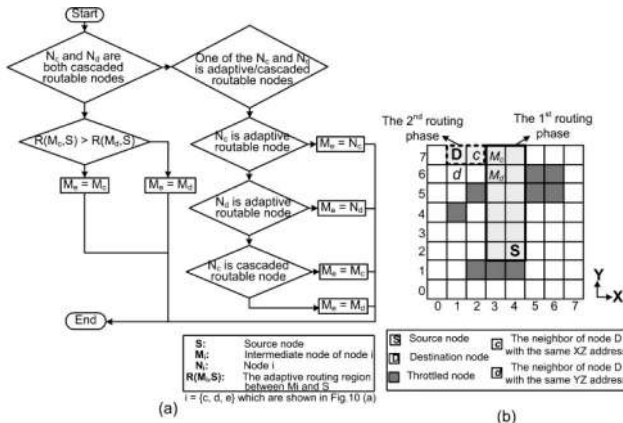


Fig. 11. (a) The operation flow of intermediate node decision, and (b) An example of the proposed intermediate node.

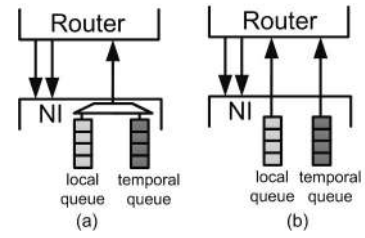


Fig. 12. (a) The two queues in  $NI$  share the same output channel to router, and (b) The two queues own dedicated output channels.

$$\sum_{V=1}^{X_d-2} \sum_{H=1}^{Y_d-2} \frac{(V+H)!}{V!H!} \cdot \frac{[(X_d-V)+(Y_d-H)]!}{(X_d-V)!(Y_d-H)!} \quad (1)$$

possible paths to be checked by minimal adaptive routing. If the lateral mesh size is  $N \times N$ , the total computational complexity is  $O(2^{N^2})$ , which results in high computational complexity. Considering the computational complexity and lateral path diversity maximization, we only check the maximum minimal-path region with no throttled node to find the intermediate node through incremental checking, as shown in Fig. 10b. Therefore, the computational complexity is reduced to  $O(N)$ . Fig. 11a shows the decision flow of intermediate node, and Fig. 11b is an example. Based on the rules of Fig. 10a, the destination node  $D$  is a cascaded routable node, and the two neighbors of node  $D$  (i.e., node  $c$  and  $d$ ) are both cascaded routable nodes. Besides, the intermediate nodes of node  $c$  and  $d$  are  $M_c$  and  $M_d$ , respectively. The minimal-path region between  $M_c$  and  $S$  is defined as  $R(M_c, S)$ . In this case,  $R(M_c, S)$  is equal to 12 (i.e.,  $(|X_{M_c} - X_S| + 1) \times (|Y_{M_c} - Y_S| + 1) = 12$ ). On the other hand, the minimal-path region between  $M_d$  and  $S$  is 10. The intermediate node of node  $D$  is referred as  $M_c$  to maximize the lateral path diversity (i.e.,  $R(M_c, S) > R(M_d, S)$ ).

### 3.1.3 Performance Analysis of Different Interconnection between Router and NI

The propagation delay is the critical issue. For the input interconnection of each  $NI$ , we use two dedicate input channels to prevent the normal packets (i.e., the packet can arrive at its destination without cascaded routing) and rerouted packets from being blocked by each other, as shown in Fig. 12. For the output interconnection of each  $NI$ , through the M/M/1 queuing analysis, it is obvious to show that the performance with two dedicated output channels is better. The involved notations of analysis are shown in Table 1.

In Fig. 12, the local queue is used to store the local generated packet temporally, and the temporal queue is used to store and forward those rerouted packets. For each output queue of  $NI$  of option 1 (i.e., Fig. 12a), the effective service rate is  $\mu_{eff1}/2$ , if there is no starvation. Because of

TABLE 1  
The Parameter Notation

Parameter	Description
$\lambda$	The average arrival rate
$\mu$	The average service rate of each router
$\mu_{eff1}$	Effective service rate of Fig. 12 (a)
$\mu_{eff2}$	Effective service rate of Fig. 12 (b)

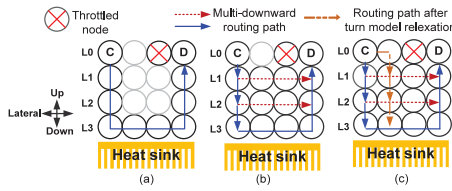


Fig. 13. (a) Conventional downward routing, (b) Multidownward routing, and (c) Lateral-downward turn relaxation.

the switching contention, the range of  $\mu_{eff1}$  can be obtained. By Little's formula, the average waiting time of option 1 is

$$W_{op1} = \frac{2}{\mu_{eff1} - 2\lambda}, \quad \text{where } \mu/6 \leq \mu_{eff1} \leq \mu. \quad (2)$$

With similar analysis, the average waiting time of each output queue of NI in option 2 (i.e., Fig. 12b) is

$$W_{op2} = \frac{1}{\mu_{eff2} - \lambda}, \quad \text{where } \mu/7 \leq \mu_{eff2} \leq \mu. \quad (3)$$

With (2) and (3), it is obvious to find the range of average waiting time of option 1 and option 2:

$$\frac{2}{\mu/6 - 2\lambda} \geq W_{op1} \geq \frac{2}{\mu - 2\lambda} \quad (4)$$

and

$$\frac{1}{\mu/7 - \lambda} \geq W_{op2} \geq \frac{1}{\mu - \lambda}. \quad (5)$$

By (4) and (5), the interconnection method between NI and router will be design based on option 2 due to the minimization of average waiting time in worst case.

### 3.2 Vertical Routing: Vertical Traffic Balancing Routing (VTBR)

To solve the problem of small vertical path diversity (i.e., Fig. 13a) and balance the vertical traffic, the TAAR adopt the Vertical Traffic Balancing Routing as the vertical routing, which will be introduced in this section. The VTBR can permit the downward packet pass through any nonthrottling layer to the destination, as shown in Fig. 13b. Because the downward routing path is not restricted to the bottom logic layer, the vertical traffic becomes more balanced.

#### 3.2.1 Selection of Multiple Downwarding Layers

To find the feasible downward ratio of each nonthrottling layer, an M/M/1 queuing analysis is proposed. We use Fig. 14a as an example and derive the downward ratio. While a 3D NoC system has two throttled nodes at L0 and L1, the probability of downward routes through the L2 and L3 from L0 are  $\alpha_{L2}$  and  $\alpha_{L3}$ , respectively. The  $\alpha_{L3}$  is equal to  $1 - \alpha_{L2}$ . The corresponding effective queuing model is shown in Fig. 14b. To generalization, the average lateral arrival rates of the two nonthrottling layers are assumed as  $\lambda_{L2}$  and  $\lambda_{L3}$ , which are different in  $\varepsilon_2$  (i.e.,  $\varepsilon_2 = \lambda_{L2} - \lambda_{L3}$ ). The effective finite queuing lengths of L2 and L3 are the same as  $L$ . By Little's formula, the average waiting time of L2 and L3 are

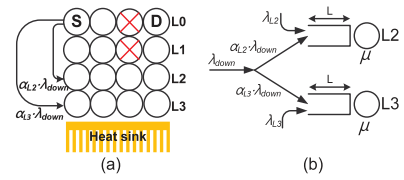


Fig. 14. (a) Downward layer selection with a ratio,  $\alpha$ , and (b) Effective queuing system of the two downward layers.

$$W_{L2} = \frac{L}{\lambda_{L2} + \alpha_{L2} \cdot \lambda_{down}} = \frac{L}{\lambda_{L3} + \lambda_{down} + (\alpha_{L2} - 1) \cdot \lambda_{down} + \varepsilon_2} \quad (6)$$

and

$$W_{L3} = \frac{L}{\lambda_{L3} + (1 - \alpha_{L2}) \cdot \lambda_{down}} = \frac{L}{\lambda_{L3} + \lambda_{down} - \alpha_{L2} \cdot \lambda_{down}}, \quad (7)$$

where  $\lambda_{down}$  is the arrival rate of the downward packets. By Poisson Arrival See Time Averages (PASTA) property, the average waiting time means the total processing time for the total amount of data  $L$  in the current queue, if the coming packet is the last one for this queuing system. Therefore, the total processing time  $T$  to handle all data in L2 and L3 is the larger one between  $W_{L2}$  and  $W_{L3}$ . Obviously,  $T$  is smallest when  $W_{L2}$  and  $W_{L3}$  are identical. Therefore, the feasible downward ratio of L2 and L3 are

$$\alpha_{L2} = \frac{-\varepsilon_2 + \lambda_{down}}{2\lambda_{down}} \quad \text{and} \quad \alpha_{L3} = \frac{\varepsilon_2 + \lambda_{down}}{2\lambda_{down}}. \quad (8)$$

With the similar derivation, for a four chip stacking 3D NoC, the downward ratio of each logic layer for every throttling state of TAVT [10], [17], which are the throttled state 0, 1, 2, and 3 in Fig. 1a, are shown in Table 2. The  $\alpha_{L1}$ ,  $\alpha_{L2}$ , and  $\alpha_{L3}$  are the downward ratio for L1, L2, and L3 for each throttling state of TAVT [10], [17]. The  $\varepsilon_1$  is the amount of difference between  $\lambda_{L1}$  and  $\lambda_{L3}$ . Because there is no throttled node in Throttle 0 state, the downward ratio of each logic layer is zero. On the other hand, the downward ratio of L3 is one in Throttle 3, because L3 is the only one nonthrottling layer.

To simplify the problem in this paper, we assume that the network is homogeneous, the effective service rate of each router is the same as  $\mu$ , and the average arrival rates (i.e.,  $\lambda_{Ln}$ ,  $n = \{1, 2, 3\}$ ) of each logic layer are equal to  $\lambda$ . We can obtain the following Lemma.

TABLE 2  
The Feasible Downward Ratio of Each Throttling State of TAVT

	Throttle 0	Throttle 1	Throttle 2	Throttle 3
$\alpha_{L1}$	0	$\frac{\lambda_{down} - 2\varepsilon_1 + \varepsilon_2}{3\lambda_{down}}$	0	0
$\alpha_{L2}$	0	$\frac{\lambda_{down} + \varepsilon_1 - 2\varepsilon_2}{3\lambda_{down}}$	$\frac{-\varepsilon_2 + \lambda_{down}}{2\lambda_{down}}$	0
$\alpha_{L3}$	0	$\frac{\lambda_{down} + \varepsilon_1 + \varepsilon_2}{3\lambda_{down}}$	$\frac{\varepsilon_2 + \lambda_{down}}{2\lambda_{down}}$	1

**Lemma 1.** *If the packet arrival is a Poisson distribution and the network is homogeneous, the uniform random downward layer selection can obtain the maximum throughput.*

The proof, which uses Throttle 2 as an example, is shown as follows, and the same result will be obtained in Throttle 1 state. Because the network is homogeneous, the  $\varepsilon_2$  of (6) and (7) is zero. Therefore,  $T$  is smallest ( $T_{opt}$ ), if  $\alpha_{L2}$  is 0.5 by

$$T_{opt} = \frac{L}{\lambda + 0.5\lambda_{down}} \leq \max \left\{ \frac{L}{\lambda + \alpha_{L2} \cdot \lambda_{down}}, \frac{L}{\lambda + (1 - \alpha_{L2}) \cdot \lambda_{down}} \right\}. \quad (9)$$

In (10), if each L2 and L3 has total  $L$  data to process, the throughput ( $\theta$ ) is the maximum ( $\theta_{opt}$ ), when the downward ratio of L2 and L3 are identical (i.e.,  $\alpha_{L2} = 0.5$ ):

$$\theta_{opt} = \frac{2L}{T_{opt}} \geq \frac{2L}{T} = \theta. \quad (10)$$

Therefore, in this paper, we use uniform random downward layer selection policy to achieve high network throughput and balanced interlayer traffic.

### 3.2.2 Relaxation in Lateral-Downward Turn

To further increase the path diversity in the source layer, we release the lateral-downward turns (i.e., North-Down, East-Down, South-Down, West-Down), as shown in Fig. 13c. As the proof in [10], it is deadlock-free, if there is no upward turn (i.e., Up-North, Up-East, Up-South, and Up-West) and the lateral routing is deadlock-free.

To detour the throttled nodes, we use the topology information in  $NI$  to prevent the packet from being blocked in the throttled node. If the neighboring node, which is in the routing path, is throttled, the packets must be delivered by using multidownward routing. In contrast, if the neighboring node is nonthrottled, the output direction will be selected with the channel information on the output sides. It can adopt any buffer level-based selection [23], [24]. To realize the throttling aware selection, each router use a 4-bit register, Neighbor Throttling Status ( $NTS$ ), to record the throttling state of its neighboring nodes of lateral four directions (i.e., North, East, South, and West). The  $NTS$  can be updated by using the information of the  $TT$  in  $NI$ . The architecture of router and  $NI$  will be introduced in Section 5.

## 4 EXPERIMENTAL RESULTS AND DISCUSSION

To evaluate the proposed  $TAAR$ , the synthetic traffic and the data flow of Low-Density Parity-Check (LDPC) [34], [35] are simulated through the traffic-thermal cosimulation platform [9], which uses  $Noxim$  [26] and  $HotSpot$  [27] to simulate the 3D NoC platform and the distributed thermal sensor, respectively. For each router, the channel depth of the buffer is four flits without VC for low cost issue [31]. The network size is an  $8 \times 8 \times 4$  3D NoC, and the packet length is randomly from 2 to 10 flits. Because the  $SAF$  policy and the wormhole flow control scheme are both adopted, the depth of the temporal queue of Fig. 12 is the same as the maximum packet size in the network (i.e., 10 flits in this work) to prevent from deadlock. Through statistical traffic

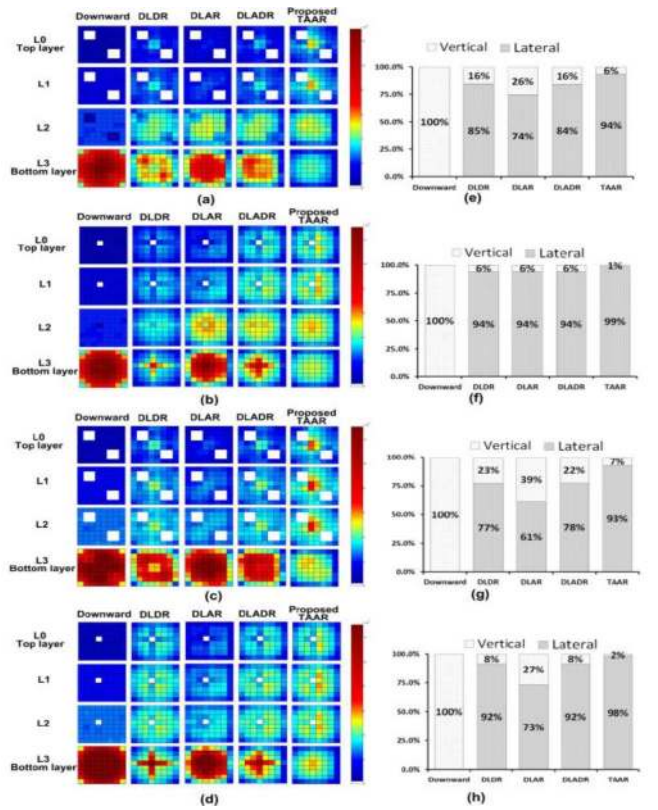


Fig. 15. (a)-(d) Statistical traffic load distribution of Cases 1 to 4 under uniform traffic pattern, and (e)-(h) the distribution of each routing mode of Cases 1 to 4.

load distribution ( $STLD$ ) [14], we can compare the network loading distribution.

### 4.1 Performance Evaluation in Synthetic Traffic Pattern

To evaluate the performance by using the synthetic traffic pattern, we use four throttling cases of  $TAVT$ : (Case 1) two  $2 \times 2 \times 2$  throttling regions, (Case 2) one  $1 \times 1 \times 2$  throttling region, (Case 3) two  $2 \times 2 \times 3$  throttling regions, and (Case 4) one  $1 \times 1 \times 3$  throttling region, which are shown in Fig. 15. To demonstrate the general traffic behavior with complex communications in NoC systems, we use three popular synthetic traffic patterns: uniform, hotspot, and transpose 2 [15], [16], [17], [18], [19].

Fig. 15a shows the  $STLD$  of Case 1 under uniform traffic pattern, which uses the same packet injection rate that makes average latency of downward routing twice the zero load latency. Compared with the previous works [10], [11], [17], the proposed  $TAAR$  can balance more vertical traffic. The reason is that the proposed  $TAAR$  have more path diversities in source layer and multiple vertical downward layer selection. Fig. 15e shows the distribution of the routing mode decision. Because of the larger lateral path diversity provided by the proposed  $TAAR$ , the total ratio of downward routing is reduced to 6 percent. The results of Case 2 to Case 4 are similar to Case 1, which are shown in Fig. 15.

The statistics of transferred flits are shown in Table 3. In Case 1, compared with the previous works in uniform traffic, the  $TAAR$  can reduce the traffic load in the congestion bottom layer from 58 to 227 percent. The similar

TABLE 3  
Statistics of Passing Flits for 50,000 Cycles in Uniform Traffic

Case 1: two 2x2x2 throttling regions					
	Downward	DLDR	DLAR	DLADR	TAAR
L0	6,794	16,940	11,809	16,951	22,999
L1	13,440	21,324	17,084	21,413	26,793
L2	20,911	33,704	35,386	33,484	33,644
L3	91,023	44,028	55,833	44,263	27,813
I Stdv.	39,087	12,279	19,951	12,326	4,405
Case 2: one 1x1x2 throttling regions					
	Downward	DLDR	DLAR	DLADR	TAAR
L0	7,645	24,503	17,382	24,358	26,735
L1	15,083	28,737	22,440	28,713	30,984
L2	22,464	32,274	34,599	32,225	31,563
L3	95,525	33,338	49,989	33,623	27,524
I Stdv.	40,683	3,991	14,517	4,134	2,426
Case 3: two 2x2x3 throttling regions					
	Downward	DLDR	DLAR	DLADR	TAAR
L0	6,785	17,290	11,933	17,312	23,131
L1	13,401	21,603	17,243	21,710	27,043
L2	19,884	24,019	20,904	23,926	27,818
L3	88,070	51,107	68,932	51,030	30,951
I Stdv.	37,737	15,323	26,378	15,273	3,216
Case 4: one 1x1x3 throttling region					
	Downward	DLDR	DLAR	DLADR	TAAR
L0	7,637	24,516	17,245	24,614	26,931
L1	15,070	28,735	22,539	28,752	30,923
L2	22,346	29,732	25,165	29,542	30,867
L3	95,350	36,318	60,788	36,485	28,138
I Stdv.	40,613	4,883	19,845	4,924	2,002

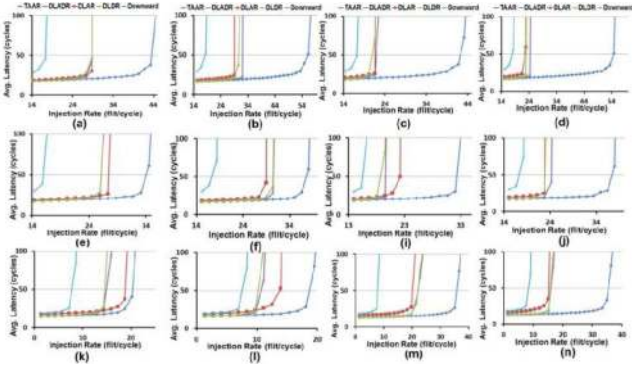


Fig. 16. Performance evaluation of Cases 1 to 4 in (a)-(d) uniform traffic pattern, (e)-(j) hotspot traffic, and (k)-(n) transpose 2 traffic.

results are also shown in Case 2 to Case 4. Because the standard deviations of interlayer traffic load ( $I\_Stdv.$ ) are decreased by using the proposed TAAR scheme, the proposed TAAR can achieve more balanced traffic in the vertical direction. The results of *STLD* and statistics of transferred flits are similar to uniform traffic pattern in hotspot and transpose 2 traffic patterns.

The performance evaluation of proposed TAAR in the uniform traffic, hotspot traffic, and transpose 2 traffic are shown in Fig. 16. Compared with the previous works [10], [11], [17], the system throughput improvement of the proposed TAAR is shown in Table 4.

#### 4.2 Case Study of Real Traffic Data

To evaluate the proposed TAAR in the real application, we refer to the Low-Density Parity-Check and Fast Fourier Transform (FFT). The LDPC and FFT are widely used in most of the advanced wireless technologies.

In general, the LDPC decoding algorithm can be represents as a parity  $H$  matrix or a bipartite graph, as shown in Fig. 17. As the communication technology

TABLE 4  
The Throughput Improvement of the TAAR Toward Each Cases and the Previous Works in Different Traffic Patterns

Downward [10]			
	Uniform	Hotspot	Transpose 2
Case 1	172.7%	109.1%	133.3%
Case 2	227.3%	118.2%	380.0%
Case 3	163.6%	109.1%	125.6%
Case 4	218.2%	118.2%	378.1%
Avg. Improvement	195.5%	113.7%	254.3%
DLDR [11][17]			
	Uniform	Hotspot	Transpose 2
Case 1	57.9%	27.8%	40.0%
Case 2	80.0%	20.0%	60.0%
Case 3	107.1%	76.9%	100.0%
Case 4	133.3%	60.0%	118.2%
Avg. Improvement	94.6%	46.2%	79.6%
DLAR [11][17]			
	Uniform	Hotspot	Transpose 2
Case 1	50.0%	21.1%	7.7%
Case 2	89.5%	26.3%	84.6%
Case 3	93.3%	53.3%	40.0%
Case 4	150.0%	60.0%	140.0%
Avg. Improvement	95.7%	40.2%	68.1%
DLADR [11][17]			
	Uniform	Hotspot	Transpose 2
Case 1	50.0%	27.8%	27.3%
Case 2	71.4%	20.0%	60.0%
Case 3	93.3%	64.3%	75.0%
Case 4	118.8%	50.0%	118.2%
Avg. Improvement	83.4%	40.5%	70.1%

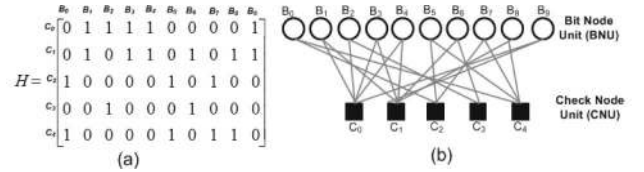


Fig. 17. The representation of a (10, 5) LDPC code in (a) H matrix, (b) Bipartite graph.

TABLE 5  
The Mapping Results of a (10, 5) LDPC Code for a 3 x 3 NoC

Node	Mapped Results	Node	Mapped Results	Node	Mapped Results
N0	B0, B9, C0	N3	B3, C3	N6	B6
N1	B1, C1	N4	B4, C4	N7	B7
N2	B2, C2	N5	B5	N8	B8

development, it is necessary to have a flexible LDPC codes, which results in difficulty in hardware realization. To increase the efficiency of the hardware implementation of the LDPC, an NoC-based wire connection was proposed to realize the flexible LDPC decoder because of the characteristics of regularity and scalability [34].

Similar to the methods in [34], [35], we evaluate the proposed TAAR with the data flow of the (1,944, 972) LDPC codes, which is defined in IEEE 802.11n [32], and the (960, 480) LDPC codes, which is defined in IEEE 802.16e [33]. In our simulation, the mapping of *BNU*s and *CNU*s onto *PE*s is done by

$$\text{Mapped node number} = (\text{BNU (or CNU) number}) \bmod (\text{Total number of NoC nodes}). \quad (11)$$



TABLE 6

The Comparison of the System Performance of LDPC Codes

		(1944, 972) LDPC Codes in 802.11n Standard				
		Downward	DLDR	DLAR	DLADR	TAAR
Throughput (flit/cycle)	Max.	20.58	30.76	13.95	15.68	53.39
	Min.	19.50	14.74	11.83	13.02	25.84
Avg. latency (cycles)		13,914	8,682	27,676	27,567	2,605
		(960, 480) LDPC Codes in 802.16e Standard				
		Downward	DLDR	DLAR	DLADR	TAAR
Throughput (flit/cycle)	Max.	21.20	30.81	38.36	34.94	58.52
	Min.	19.61	19.89	22.64	22.82	50.26
Avg. latency (cycles)		10,038	3,664	6,459	5,547	456

TABLE 7

The Comparison of the Performance of FFT

		SPLASH-2 benchmark (FFT)				
		Downward	DLDR	DLAR	DLADR	TAAR
Throughput (flit/cycle)	Max.	38.18	98.28	76.84	83.21	109.98
	Min.	37.68	54.30	40.71	57.34	75.50
Avg. latency (cycles)		42,107	39,794	40,189	40,241	39,596

Table 5 shows an example of mapping results, which map a (10, 5) LDPC code on a  $3 \times 3$  NoC platform. The comparison of system throughput in a real data flow of LDPC between the proposed TAAR and previous related works [10], [11], [17] is shown in Table 6. Compared with the previous works, the proposed TAAR can reduce the average latency by around 70 to 95 percent and obtain better throughput.

In addition to LDPC code, we refer to the FFT benchmark in SPLASH-2 [37] to evaluate the proposed TAAR. By following the proposed mapping algorithm in [38], the 64k-points FFT of SPLASH-2 can be mapped to our  $8 \times 8 \times 4$  3D NoC platform. Table 7 shows the comparison between the previous works and the proposed TAAR. Obviously, the proposed method still validates the effectiveness of the system performance.

## 5 VLSI ARCHITECTURE DESIGN OF TAAR SCHEME

For a thermal-aware architecture design for multicore system, embedded thermal sensor for each router is a popular practical method [29], [30]. Therefore, the conventional thermal-aware architecture designs of a 3D NoC system only focus on the router architecture design, as shown in Fig. 18a. However, the architectures cannot support TAAR because the NI cannot provide the suitable information of topology and the intermediate node. In this section, a new architecture design of NI and router for supporting the proposed TAAR is provided, as shown in Fig. 18b.

### 5.1 Architecture Design of Network Interface

#### 5.1.1 Transmitter and Receiver (Tx/Rx) Unit

Tx deals with the message from application layer and packetizes the payloads to packets which will be injected to network layer. In contrast, Rx receives packet from network layer and sent to application layer. To prevent from deadlock, the depth of temporal queue in Tx is set as the maximal packet size in the network.

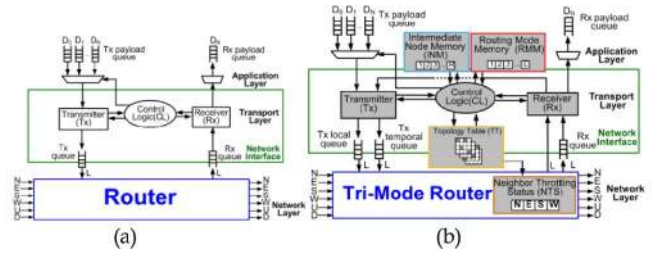


Fig. 18. (a) The conventional architecture and (b) TAAR architecture.

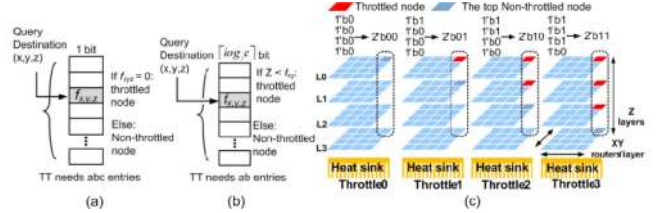


Fig. 19. (a) Direct TT implementation, (b) The proposed TT implementation, and (c) An example of the TT reduction.

#### 5.1.2 Topology Table

For each router, the throttling information for each destination is stored in TT and updated by CL while topology changing. Directly implement TT for an  $a \times b \times c$  3D NoC needs  $abc$  bits to store the throttling information for each node, as shown in Fig. 19a. Hence, table reduction is needed to reduce the overhead. However, the throttling status of each node in the same horizontal layer should be known to maximize the lateral routable paths. Therefore, we propose a method to reduce the table size of TT based on the two characteristics of TAVT [10]: 1) if a router is throttled, all the routers above it are throttled; 2) if a router is not throttled, all the routers below it are not throttled. By this way, we can only store which layer is the top of the nonthrottling routers, as shown in Fig. 19b. Fig. 19c shows an example. Because there are four logic layers, each table entry needs only 2 bits to store the top of the nonthrottling routers. In general, the number of bits can be reduced from  $abc$  bits to  $ab \lceil \log_2 c \rceil$  bits. For an  $N \times N \times N$  3D NoC system, we reduce the total complexity of the table scalability from  $O(N^3)$  to  $O(N^2 \log_2 N)$ .

#### 5.1.3 Routing Mode Memory

RMM is required to reduce the timing overhead of checking routing mode for each packet. The routing mode for each destination is stored in RMM and updated by CL while topology changing. Before packet injection, the corresponding routing mode is queried from RMM.

Directly using TAAR for an  $a \times b \times c$  3D NoC, we should need  $2abc$  bits to record the applied routing mode for each destination. Because there is no upward turn in TAAR, the size of RMM can be reduced through identical information in vertical dimension (i.e., RMM can only store the applied routing mode of each node in the source layer). Therefore, RMM only needs  $2ab$  bits to store the routing mode of every destination. In general, for an  $N \times N \times N$  3D NoC system, we reduce the total complexity of the table from  $O(N^3)$  to  $O(N^2)$ .

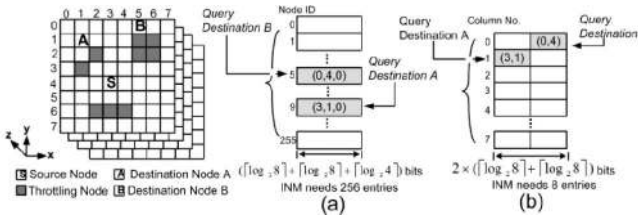


Fig. 20. (a) Direct *INM* implementation, and (b) The proposed *INM* implementation can reduce the table size.

### 5.1.4 Intermediate Node Memory (*INM*)

The intermediate node for each destination is store in *INM*. For a rerouted packet, the corresponding intermediate node is set by referring the *INM*. Directly implementing *INM* for an  $a \times b \times c$  3D NoC needs  $(abc(\lceil \log_2 a \rceil + \lceil \log_2 b \rceil + \lceil \log_2 c \rceil))$  bits to record the location of intermediate node toward each destination, as shown in Fig. 20a. As shown in Fig. 11, the intermediate node will be the same as the one of the two neighboring nodes toward the source node. Therefore, we can store the intermediate node of the neighboring node for the destination in the same row.

Because of the characteristics of *TAVT*, we can only store the intermediate node information in the same logic layer. In addition to the vertical information alignment, the table size of *INM* can be further reduced by using horizontal information alignment. Therefore, *INM* only need  $(2ab(\lceil \log_2 a \rceil + \lceil \log_2 b \rceil + \lceil \log_2 c \rceil))$  bits to store the candidate of intermediate node, as shown in Fig. 20b. In general, for an  $N \times N \times N$  3D NoC system, we reduce the total complexity of the table scalability from  $O(N^3)$  to  $O(N^2)$ .

### 5.1.5 Control Logic (*CL*)

Fig. 21 shows the flowchart of *CL*. As mentioned in Section 3, the network has two stages: 1) normal stage and 2) reconfiguration stage. The *CL* has different functions in the two stages, which are described as follows:

1. If the network enters the reconfiguration stage, the topology will be changed based on the temperature from the embedded thermal sensors. Therefore, the *TT* will be updated by *CL*. Based on the information of updated *TT*, the *RMM* and *INM* are updated simultaneously.
2. If the network enters the normal stage, *Tx* will be checked first. If the packet in *Tx* is a new generated one, the routing mode and intermediate node will be assigned. If the packet in *Tx* belongs to rerouted

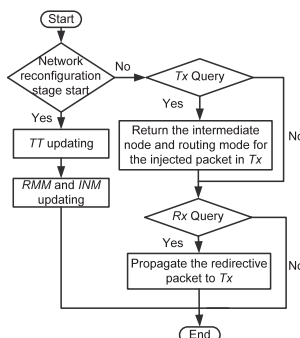


Fig. 21. The operation flow of *CL*.

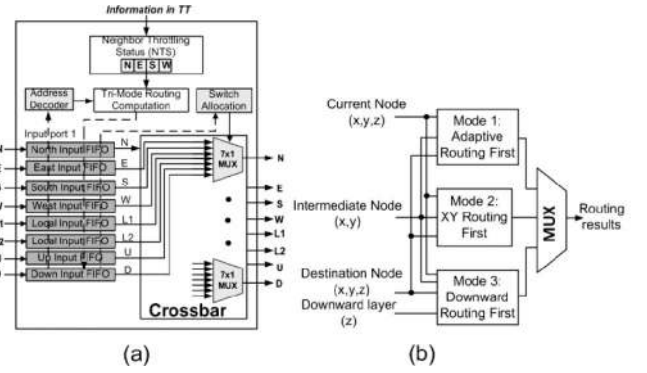


Fig. 22. The architecture of (a) The proposed 3D trimode router, and (b) Trimode routing computation logic.

packet, the new routing mode will be assigned based on the topology information in *TT*. After checking the *Tx*, the *Rx* will be checked whether any rerouted packet is received. If the *Rx* receives any rerouted packet, the *CL* will propagate it to the *Tx* for further processing.

## 5.2 Architecture Design of Trimode Router

Based on the router design in [25], we propose a trimode 3D NoC router architecture to support the *TAAR*, as shown in Fig. 22a. The 3D router requires extra two physical channels (*U* and *D*) for vertical connections. Besides, there are two dedicated channels (*L1* and *L2*) to the connection between the *NI* and the router. Therefore, the size of crossbar increases from  $5 \times 5$  to  $8 \times 8$ . To support the three routing modes of *TAAR*, the architecture of trimode routing computation logic is shown in Fig. 22b.

## 5.3 VLSI Implementation and Analysis of Hardware Overhead

We implement the *NI* and router which will be designed for an  $8 \times 8 \times 4$  3D NoC. We assume each flit has 64 bits [5] that contains 22-bit header and 42-bit payload. The header uses 2 bits to represent three different flit types: head flit, body flit, or tail flit. Besides, we use 1-bit to indicate the normal packet or rerouted packet. Because this design will be used for an  $8 \times 8 \times 4$  3D NoC, 9 bits are used to indicate the destination, and 2 bits are used to indicate the assigned downward layer. Because the intermediate node is always located in the source layer, we use 6 bits to represent the location of the intermediate node. Finally, 2 bits are used to represent three different routing modes. Based on UMC 90-nm technology, the area overhead comparison between the previous works in [10], [11], [17] and the proposed *TAAR* is shown in Table 8.

TABLE 8  
Area Comparison of the *TAAR* and the Related Works

	Downward	DLDR	DLAR	DLADR	TAAR
<i>NI:</i>					
<i>Tx/Rx/CL</i>	42,993	42,993	42,993	42,993	49,631
<i>NI: RMM</i>	0	2,448	2,448	4,476	4,476
<i>NI: TT</i>	0	2,448	2,448	2,448	2,448
<i>NI: INM</i>	0	0	0	0	2,978
<i>NI: CL</i>	241	619	5,386	4,663	6,387
Router	152,898	155,635	155,833	155,901	178,758
<b>Total</b>	<b>196,132</b>	<b>204,143</b>	<b>209,108</b>	<b>210,481</b>	<b>244,678</b>

Unit:  $\mu\text{m}^2$

TABLE 9  
Throughput/Area Comparison After Normalization

	Downward	DLDR	DLAR	DLADR	TAAR
Throughput improvement (T)	1	1.681	1.728	1.753	2.878
Hardware cost overhead (H)	1	1.041	1.066	1.073	1.248
Area efficiency (T/H)	1	1.615	1.621	1.633	2.306

Because there are two dedicated input/output channels in *NI*, the hardware cost of Tx/Rx of *TAAR* is bigger than the previous works. The hardware cost of router of *TAAR* is bigger than the previous works, because the *CS*, *SA*, and *SD* are more complex than the previous works. Besides, there is one *NTS* to record the throttling status of the four lateral neighboring nodes, which leads to extra hardware overhead. Because there are three routing modes in *DLADR* and the proposed *TAAR*, the hardware cost of *RMM* of the two approaches are identical and bigger than *DLDR* and *DLAR*. For the fair comparison, the proposed memory reduction techniques are applied to implement the *RMM*, *TT*, and *INM* of each compared approach.

To compare the tradeoff between the area cost and performance improvement, we normalize the average throughput improvement and hardware cost overhead of each approaches based on the baseline design, downward routing [10]. Table 9 shows the throughput/area comparison after normalization. Obviously, the proposed *TAAR* can significantly improve the throughput with acceptable hardware cost overhead.

## 6 CONCLUSION

To increase the intra- and interlayer path diversity, we propose a *TAAR* for the *NSI-Mesh*. *TAAR* can balance the lateral and vertical traffic load, respectively. Compared with the related works, the proposed *TAAR* can reduce around 19 to 295 percent traffic loads in the congestion bottom layer, and improve around 7.7 to 380 percent network throughput with acceptable hardware cost overhead.

## ACKNOWLEDGMENTS

This work was supported by the National Science Council of Taiwan under Grant NSC 100-2220-E-002-013 and NSC 100-2220-E-002-016.

## REFERENCES

- [1] S. Kumar et al., "A Network on Chip Architecture and Design Methodology," *Proc. Int'l Symp. Very Large Scale Integration*, pp. 105-112, 2002.
- [2] Y. Jin et al., "Communication-Aware Globally-Coordinated On-Chip Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 23, no. 2, pp. 242-254, Feb. 2012.
- [3] R. Marculescu et al., "Outstanding Research Problems in NoC Design: System, Microarchitecture, and Circuit Perspectives," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 1, pp. 3-21, Jan. 2009.
- [4] A.W. Topol et al., "Three-Dimensional Integrated Circuits," *IBM J. Research Development*, vol. 50, no. 4/5, pp. 491-506, 2006.
- [5] B.S. Feero and P.P. Pande, "Networks-On-Chip in a Three Dimensional Environment: A Performance Evaluation," *IEEE Trans. Computers*, vol. 58, no. 1, pp. 32-45, Jan. 2009.
- [6] Mobile Intel Pentium 4 Processor - M Datasheet, <http://www.intel.com>, 2013.
- [7] L. Shang et al., "Thermal Modeling, Characterization and Management of On-Chip Networks," *Proc. IEEE/ACM Int'l Symp. Microarchitecture (Micro)*, pp. 67-68, Dec. 2004.
- [8] K. Lahiri et al., "Performance Analysis of Systems with Multi-Channel Communication," *Proc. Int'l Conf. VLSI Design*, pp. 530-537, 2000.
- [9] K.Y. Jheng et al., "Traffic-Thermal Mutual-Coupling Co-Simulation Platform for Three-Dimensional Network-on-Chip," *Proc. IEEE Int'l Symp. VLSI Design, Automation, and Test (VLSI-DAT)*, Apr. 2010.
- [10] C.H. Chao et al., "Traffic- and Thermal-Aware Run-Time Thermal Management Scheme for 3D NoC System," *Proc. IEEE Int'l Symp. Network-on-Chip (NOCS)*, pp. 223-230, May 2010.
- [11] C.H. Chao et al., "Transport Layer Assisted Routing for Non-Stationary Irregular Mesh of Thermal-Aware 3D Network-on-Chip Systems," *Proc. IEEE Int'l SOC Conf. (SOCC)*, Sept. 2011.
- [12] M.K.F. Schafer et al., "Deadlock-Free Routing and Compinent Placement for Irregular Mesh-Based Network-on-Chip," *Proc. IEEE/ACM Int'l Conf. Computer-Aided Design*, pp. 238-245, 2005.
- [13] R. Holsmark and S. Kumar, "Design Issues and Performance Evaluation of Mesh NoC with Regions," *Proc. Norchip Conf.*, pp. 40-43, 2005.
- [14] S.Y. Lin et al., "Traffic-Balanced Routing Algorithm for Irregular Mesh-Based On-Chip Networks," *IEEE Trans. Computers*, vol. 57, no. 9, pp. 1156-1168, Sept. 2008.
- [15] P. Mahdavinia et al., "An Efficient Routing Algorithm for Irregular Mesh NoCs," *Proc. IEEE Int'l Symp. Circuit and System (ISCAS)*, 2010.
- [16] S.Y. Lin et al., "Traffic-and Thermal-Aware Routing for Throttling Three-Dimensional Network-on-Chip System," *Proc. IEEE Int'l Symp. VLSI Design, Automation, and Test (VLSI-DAT)*, pp. 135-138, Apr. 2011.
- [17] C.H. Chao et al., "Transport Layer Assisted Routing for Run-Time Thermal Management of 3D NoC Systems," *to be Appeared in, ACM Trans. Embedded Computing Systems*.
- [18] 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.
- [19] P. Gratz et al., "Regional Congestion Awareness for Load Balance in Network-on-Chip," *Proc. IEEE Symp. High Performance Computer Architecture (HPCA)*, pp. 203-214, Feb. 2008.
- [20] L.G. Valiant, "A Scheme for Fast Parallel Communication," *SIAM J. Computing*, vol. 11, pp. 350-361, 1982.
- [21] T. Nesson and S.L. Johnsson, "ROMM Routing on Mesh and Torus Networks," *Proc. ACM Symp. Parallel Algorithms and Architectures (SPAA)*, 1995.
- [22] A. Singh et al., "GOAL: A Load-Balanced Adaptive Routing Algorithm for Torus Networks," *Proc. Int'l Symp. Computer Architecture (ISCA)*, 2003.
- [23] G. Ascia et al., "Implementation and Analysis of a New Selection Strategy for Adaptive Routing in Network-on-Chip," *IEEE Trans. Computers*, vol. 57, no. 6, pp. 809-820, June 2008.
- [24] G. Ascia et al., "Neighbors-on-Path: A New Selections Strategy for on-Chip-Networks," *Proc. IEEE/ACM/IFIP Workshop Embedded Systems for Real Time Multimedia*, Oct. 2006.
- [25] J. Hu and R. Marculescu, "Application-Specific Buffer Space Allocation for Networks-on-Chip Router Design," *Proc. IEEE/ACM Int'l Conf. Computer Aided Design (ICCAD)*, pp. 354-361, Nov. 2004.
- [26] Noxim: <http://sourceforge.net/projects/noxim/>, 2013.
- [27] W. Huang et al., "HotSpot: A Compact Thermal Modeling Methodology for Early-Stage VLSI Design," *IEEE Trans. Very Large Scale Integrated Systems*, vol. 14, no. 5, pp. 501-513, May 2006.
- [28] Y. Hoskote, "A 5-GHz Mesh Interconnect for a Teraflops Processor," *IEEE Micro*, vol. 27, no. 5, pp. 51-61, Sept./Oct. 2007.
- [29] X. Wang et al., "Adaptive Power Control with Online Model Estimation for Chip Multiprocessors," *IEEE Trans. Parallel and Distributed Systems*, vol. 22, no. 10, pp. 1681-1696, Oct. 2011.
- [30] A. Bartolini et al., "Thermal and Energy Management of High-Performance Multicores: Distributed and Self-Calibrating Model-Predictive Controller," *IEEE Trans. Parallel and Distributed Systems*, vol. 24, no. 1, pp. 170-183, Jan. 2013.

- [31] A. Jankes et al., "Benefits of Selective Packet Discard in Networks-on-Chip," *ACM Trans. Architecture Code Optimization*, vol. 9, no. 2, article 12, 2012.
- [32] <http://www.ieee802.org/11/>, 2013.
- [33] <http://www.ieee802.org/16/tge/>, 2013.
- [34] C. Cardo et al., "A Flexible NoC-Based LDPC Code Decoder Implementation and Bandwidth Reduction Methods," *Proc. Conf. Design and Architecture for Signal and Image Processing (DASIP)*, pp. 1-8, Nov. 2011.
- [35] W.H. Hu et al., "Parallel LDPC Decoding on a Network-on-Chip Based Multiprocessor Platform," *Proc. Int'l Symp. Computer Architecture High Performance Computers (SBAC-PAD)*, pp. 35-40, Oct. 2009.
- [36] J. Flich and J. Duato, "Logic-Based Distributed Routing for NoCs," *IEEE Computer Architecture Letters*, vol. 7, no. 1, pp. 13-16, June 2008.
- [37] S.C. Woo et al., "The Splash-2 Programs: Characterization and Methodological Considerations," *Proc. Int'l Symp. Computer Architecture*, pp. 24-36, 1995.
- [38] J.H. Bahn et al., "Parallel FFT Algorithms on Network-on-Chips," *Proc. Int'l Conf. Information Technology: New Generations (ITNG)*, pp. 1087-1093, 2008.



**Kun-Chih Chen (S'10)** received the BS degree from National Taiwan Ocean University, Taiwan, in computer science and engineering in 2007, the MS degree from National Sun Yat-sen University, Taiwan, in computer science and engineering in 2009, and is currently working toward the PhD degree at the Graduate Institute of Electronics Engineering, National Taiwan University, Taiwan. His research fields include the architecture and algorithm design for three-

dimensional Networks-on-Chip (3D NoC), advance arithmetic unit design, and fault-tolerant system designs. He is a student member of the IEEE.



**Shu-Yen Lin (M'13)** received the BS degree in electronic engineering from Fu Jen Catholic University, Taiwan, in 2002, and the MS degree in electronic engineering from National Taiwan University of Science and Technology, Taiwan, in 2004, and the PhD degree in Graduate Institute of Electronics Engineering, National Taiwan University, Taiwan, in 2009. He was an engineer at Information and Communications Research Laboratories (ICL), Industrial Technology Research Institute (ITRI), Hsinchu, Taiwan, from 2011 to 2013. He is currently an assistant professor in department of electrical engineering, Yuan Ze University, Taoyuan, Taiwan. His research fields include the algorithm and architecture design for 2D/3D interconnection, VLSI system, and 3D-IC system. He is a member of the IEEE.



**Hui-shun Hung** received the BS degree from National Taiwan University, Taipei, Taiwan, in 2010, and the MS degree from National Taiwan University, Taipei, in 2012, both in electronic engineering. He is currently an engineer at MediaTek Inc., Hsinchu, Taiwan. His research interests include the design of very large-scale integration architectures and routing algorithm on thermal-aware 3D network-on-chip systems.



**An-Yeu (Andy) Wu (S'91-M'96-SM'12)** received the BS degree from National Taiwan University in 1987, and the MS and PhD degrees from the University of Maryland, College Park in 1992 and 1995, respectively, all in electrical engineering. In August 2000, he joined the faculty of the Department of Electrical Engineering and the Graduate Institute of Electronics Engineering, National Taiwan University (NTU), where he is currently a professor. His research interests include low-

power/high-performance VLSI architectures for DSP and communication applications, adaptive/multirate signal processing, reconfigurable broadband access systems and architectures, and System-on-Chip (SoC)/Network-on-Chip (NoC) platform for software/hardware co-design. Dr. Wu had served as the Associate Editors of *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* from 2003 to 2005, *IEEE Transactions on Circuits and Systems I: Regular Papers* in 2007, *IEEE Transactions on Circuits and Systems II: Express Briefs* from 2008 to 2009, and *IEEE Transactions on Signal Processing* from 2008 to 2010. He also served on the technical program committees of many major IEEE International Conferences, such as SiPS, AP-ASIC, ISCAS, ISPACS, ICME, SOCC, and A-SSCC. From August 2007 to December 2009, he was on leave from NTU and served as the Deputy General Director of SoC Technology Center (STC), Industrial Technology Research Institute (ITRI), Hsinchu, TAIWAN, supervising Parallel Core Architecture (PAC) VLIW DSP Processor and Multicore/Android SoC platform projects. In 2010, Dr. Wu received "Outstanding EE Professor Award" from The Chinese Institute of Electrical Engineering (CIEE), Taiwan. He is a senior member of the IEEE and the IEEE Computer Society.

► For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).