Transport Layer Assisted Routing for Non-Stationary Irregular Mesh of Thermal-Aware 3D Network-on-Chip Systems

Chih-Hao Chao, Tsu-Chu Yin, Shu-Yen Lin, and An-Yeu (Andy) Wu

Graduate Institute of Electronics Engineering, National Taiwan University
Taipei, Taiwan R.O.C

Abstract
Thermal issue is important for 3D Network-on-Chip systems. To ensure thermal safety, 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 the Transport Layer Assisted Routing (TLAR) scheme and two algorithms for thermal-aware 3D NoC. Based on the experimental results, the proposed routing scheme can reduce the latency over 57.5% and improve the throughput above 1.47x.

I. INTRODUCTION
As the complexity of the System-on-Chip grows, on-chip interconnections gradually dominate the performance. Network-on-Chip (NoC) has been proposed as a novel and practical infrastructure. Recently, die-stacking three-dimensional (3D) IC technology is emerging for its ability to achieve higher network performance with lower power consumption [1][6]. However, routers have been proven to have comparable thermal impact as processors and contributes significant to overall chip temperature in 2D NoC systems [2]. Besides, it also has been shown that routers are sources generating thermal hotspots due to their higher switching activity [4][5]. In 3D NoC systems, the heat problem is severer because of the larger power density and the longer heat dissipation paths.

To keep temperature below thermal limit, run-time thermal management (RTM) is required. Shang et al. [2] proposes ThermalHerd, a distributive and collaborative RTM scheme for 2D NoC systems. As shown in Fig. 1 the router starts from the ambient temperature and heats up toward its steady state temperature. The monitoring unit senses the temperature of the router and reports to the temperature-aware controller. When the temperature rises above the trigger level, the controller changes the control policy to throttling. When the temperature falls beneath the trigger level, the controller stops throttling. The router recovers full bandwidth to digest the waiting packets. Chao et al. propose a traffic- and thermal-aware RTM scheme for 3D NoC in [5]. The cooling speed and network availability are improved by the thermal-aware vertical throttling technique, as shown in Fig. 2. The idea is to actively create a channel for faster heat dissipation.

The problem is that the topology changing in 3D NoC results in failed packet delivery and network stall. For fast cooling, the traffic quota of the near-overheat router has to be small. When the router exhausts the quota, it fully blocks the input packets. This behavior results in that the topology becomes time-varying, as shown in Fig. 5. When there are packets blocked in the network, the congestion tree expands soon and jams the entire network. We define such 3D topology as Non-Stationary Irregular mesh (NSI-mesh). For successful data delivery and performance consideration, routing in NSI-mesh needs topology information. The contribution of this paper is to solve these problems with the following proposed scheme and algorithms:

- **Transport Layer Assisted Routing (TLAR) Scheme**: utilizes the topology information in transport layer for layer selection in NSI-mesh.
- **Downward-Lateral Routing Algorithms**: guarantee successful packet delivery and try to balance the loading of the network.

The rest of paper is organized as the following. In Section II, we define the NSI-mesh and describe the routing problem. In Section III, we present the proposed transport layer assisted routing scheme and the downward-lateral routing algorithms. In Section IV, we show and discuss several experiments to support our claim. Finally we conclude this paper in Section V.
II. PRELIMINARY

A. Definition and Description of the Non-Stationary Irregular Mesh of Thermal-Aware 3D NoC

Non-stationary irregular mesh (NSI-mesh) is a time-varying irregular-mesh. The two key characteristics of NSI-mesh of thermal-aware vertical throttling 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.

In throttling, when the number of input packet has exceeded the quota, the throttled router will block the incoming packets and not serve anymore. 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 and response time of temperature sensing, we can view that the network changes its topology at 10ms interval, which is $10^7$ cycles when the NoC is operating at 1GHz.

B. Reactive Routing Problem for Packet Delivery in Non-Stationary Irregular Mesh

The small changing interval and large range of inactive number of throttled NSI-mesh make conventional irregular or fault-tolerant routing algorithms infeasible. In our previous work [5], we proposed to keep the bottom layer of the 3D NoC not throttled because the routers in the bottom layer have largest thermal conductance for cooling. As shown in Fig. 3, downward routing is applied for reactive routing when there are throttled routers in the network. The channels of the 2D mesh in the bottom layer compose the guaranteed lateral paths for routing. The traffic loading in bottom layer is very high, so this approach results in huge congestion and low throughput.

III. PROPOSED TRANSPORT LAYER ASSISTED ROUTING (TLAR) SCHEME

A. Analysis for Guaranteed Packet Delivery

To ensure the success of packet delivery in the NSI-mesh network, we should prevent all the following four cases. First, the source router is fully throttled, as shown in Fig. 4(a). Second, the destination router is fully throttled, as shown in Fig. 4(b). Third, any one of the router on the routing path is fully throttled, as shown in Fig. 4(c). Last, the channel on the routing path is fully blocked by other long-term congested packets as shown in Fig. 4 (d). If source router is fully throttled, the packet will be blocked in the network interface. If any case of Fig. 4(b) or Fig. 4(c) occurs, the injected packets will be blocked somewhere on the routing path and form a congestion-tree.. Because the throttling state lasts about 10ms, the packet may be blocked for $10^7$ cycles. In this case, the congestion tree will expand and occupy entire network.

If the network interface of the thermal-aware 3D NoC contains the throttling state of the network, we can eliminate the two cases in Fig. 4(a) and Fig. 4(b). Fig. 4(d) is a flow control problem and the probability of occurrence can be reduced by applying virtual channel flow control or output buffering router architectures. However, the case of Fig. 4(c) is dependent on the routing path. We must guarantee that there is at least one non-fully throttled path toward destination router before we inject the packet, and the packet is routed on the guaranteed path.

To completely remove the case of Fig. 4(c), 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 overhead of source routing for optimizing performance of NSI-mesh is high. Besides, source routing cannot balance the loading of the network by adapting the network information as adaptive routing.
**B. Proposed Operation Flow and Architecture**

The proposed operation flow is shown in Fig. 5. The 3D NoC is switching between normal state and reconfiguration state. We assume a 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 operations. In normal state the 3D NoC works as usual irregular or regular mesh. After the N-cycle normal state, the network enters the R-cycle reconfiguration state. The reconfiguration state consists of three stages: (i) cleaning up and policy determination; (ii) synchronization of topology information; (iii) routing mode checking and throttling.

The network has to be cleaned up before topology changing. In reconfiguration state (i), the packetization of payloads from transport layer to network layer is paused. As shown in Fig. 6, the payloads stay in the transmitter payload queue. The transmitter packet queue becomes empty after a small period of time. In the meanwhile, the distributed thermal-aware controller in each tile determines the control policy (throttle or not) of the router within the tile for the next normal state. Application layer and transport layer both are aware of the throttling information of this tile. The proposed rules for throttling in the 3D NoC are introduced in section III.B.

Comparing to the cycle number in normal operation stage, the cycle number required for reconfiguration is much smaller. Assume the network is operated at 1GHz. In each 10ms interval, $10^5$ cycles is sufficient for a 8x8x4 3D NoC. The reconfiguration state only occupies 0.1% of the total time, as shown in Fig. 5. The setups overhead of TLAR, the reconfiguration stages, are negligible.

**C. Proposed Framework of TLAR**

To correctly select a path that makes packet delivery success, we propose the Transport Layer Assisted Routing (TLAR) scheme. Routing in TLAR is based on the reactive downward routing in [5], 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 decision of lateral routing in network layer. The decision results, what we defined as the routing mode, are saved in packet header. When the packet is injected to the network layer, and the routers follow the mode to route.

The key problem in TLAR is to choose routing layer, as shown in Fig. 7. Following the assumption in Section II.A, if source and destination routers are not throttled, the vertical path and lateral path through bottom layer will be guaranteed routable. However, the path in $Z_s$ layer is non-guaranteed routable because there may be some throttled routers blocking the packets. If there is no fully throttled router on the non-guaranteed path in $Z_s$ layer, TLAR chooses this path for lateral routing. Due 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...
above source router is forbidden owning to the limitation of turn model. As the proof in [5][5], the combined lateral and vertical routing is deadlock-free if the lateral routing is deadlock-free, and we forbid the (UN, UE, US, UW) turns.

Fig. 8(a) shows the flow chart of path selection in TLAR. The checking of lateral path is done during the reconfiguration state for each destination in the transport layer. For the packet which is going to a lateral routable destination, the lateral-first path is selected. Otherwise downward path is selected because it is guaranteed routable. Fig. 8(b) shows an example of check order. The order is based on bread-first-search (BFS) style and can be done incrementally in \(O(N^2)\) for an \(N\times N\) layer.

As shown in Fig. 6, 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. The checking functions are dependent to the routing algorithms and are described in the following sections.

**D. Proposed Algorithm 1: Downward with Lateral Deterministic Routing (DLDR)**

The proposed algorithm 1 in TLAR is the combination of downward routing and a deterministic routing. The downward routing is used for moving packets up and down in the vertical direction. The lateral deterministic routing (LDR) 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 routability, we adapt XY routing, a dimension-ordered routing (DOR), as the deterministic routing.

Checking is done in incremental style for each destination in the reconfiguration state. The transport layer controller checks if there is any throttled router on the paths in \(Z_S\) layer based on the table that stores the throttling information. The checking of all XY locations can be done in \(O(N^2)\) by using the incremental checking flow, as shown in Fig. 8(b). The dependency of routability is shown in Fig. 9(a). Because lateral routing is XY routing, node \(a\) will be routable if \(d\) is routable. Node \(b\) is routable if node \(a\) is routable. As shown in Fig. 8(a), if the packet is routable at the source layer \(Z_S\), it is routed lateral-first. The packet goes through the lateral path in the source layer and then going up or down to its destination. Otherwise it is downward-first. The packet traverses laterally in the bottom layer.

**E. Proposed Algorithm 2: Downward with Lateral Adaptive Routing (DLAR)**

Here we introduce the proposed algorithm 2, an improved version of the algorithm 1: downward with lateral adaptive routing (DLAR). The problem of the algorithm 1 is that the path diversities in source layer and in bottom layer are too small. For any source-destination pair, once a router is fully throttled on the routing path in the source layer, only downward path is available. Therefore, many packets are forced to be routed in the bottom layer. In this case the throughputs will be very limited and the latency increases fast. To increase the path diversity, we replace the deterministic routing by a throttling- and traffic-aware adaptive routing, which tries to balance the loading of the network by detouring packets away from throttled routers. We follow [7] for the details of the lateral routing. We adapt odd-even turn model in the routing function of the proposed lateral adaptive routing (LAR) algorithm. The selection function is consisted of the
the simulation period, and we modify the calculation latency in Noxim is based on the received packets during simulator to verify our idea. The statistics of packet use Noxim [3], a SystemC-based cycle-accurate NoC of the simulator are required for modeling the TLAR. We make the comparison as fair as possible, several modifications and the link level flow control protocol is request-ack. The queue depth of each input channel is 16 flits, 8×8×4, and the packet length is randomly from 2 to 10 flits. The network size is a Wormhole flow control is adopted, and random arbitration is used for switch scheduling. The network size is a 8×8×3 pillars throttled in the upper three layers of the network, the packets in the upper layers are more because the congestion in the bottom layer is relaxed by adopting TLAR. Besides, our previously proposed traffic- and throttling-aware adaptive routing algorithm also balances the loading of traffic in each layer.

B. Routing for Non-Stationary Irregular Mesh

In this section we show the performance of the proposed TLAR algorithms. We use the case of two 2x2x3 throttling regions. As shown in Fig. 11, there are two empty square regions in L1, L2, L3. First, we use the statistical traffic load distribution [8] to show the network loading. As shown in Fig. 11(a), though there is only two 2x2x3 pillars throttled in the upper three layers of the network, many packets have to be routed downward through the bottom layer. The congestion degree of the conventional downward in the bottom layer is the largest because all the packets are traversed in the bottom layer. Besides, the DOR in downward routing cannot balance the loading of the bottom layer. In Fig. 11(b), the loading of the work is more balanced by introducing the TLAR-DLDR. The difference of TLAR-DLDR and downward is that TLAR tries to route packet if the source layer is guaranteed routable. In Fig. 11(c), more packets are routed laterally in the source layer, so the network is more balanced vertically. The packets in the upper layers are more because the congestion in the bottom layer is relaxed by adopting TLAR. Besides, our previously proposed traffic- and throttling-aware adaptive routing algorithm also balances the loading of traffic in each layer.

The statistics of transferred flits is shown in TABLE I. With TLAR, the mean packet number in the most congested bottom layer is largely decreased by 33.9% if we adopt the DLDR algorithm. If we adopt DLAR in TLAR, the mean packet number is reduced by 49.1%. Not only the mean packet number is decreased, the total variance and inter-layer variance are both decreased. With TLAR-DLDR, the total standard deviation and inter-layer standard deviation are reduced by 54.6% and 54.9%. With TLAR-DLDR, the total standard deviation and inter-layer standard deviation are reduced by 75.8% and 76.3%.

IV. EXPERIMENTS

In this section, we first introduce our simulation settings and define the performance indices in section IV.A. The performance of the proposed TLAR algorithms for NSI-mesh are shown and compared in section IV.B.

A. Settings and Clarification of Performance Indices

For all the following experiments, if there is no special description, we use the following settings. Wormhole flow control is adopted, and random arbitration is used for switch scheduling. The network size is a 8×8×4, 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 request-ack.

To keep the performance indices representative and the comparison as fair as possible, several modifications of the simulator are required for modeling the TLAR. We use Noxim [3], a SystemC-based cycle-accurate NoC simulator to verify our idea. The statistics of packet latency in Noxim is based on the received packets during the simulation period, and we modify the calculation method of throughput. We use the network injection rate that makes the average latency to 400 as the throughput. Because only the deliverable payloads are packetized and injected to the network, the packet generation and injection process are modified. Originally the injection rate is simulated by generating Poisson arrived packets of given traffic distribution. In this work, we try to match the given injection rate by escaping the packet toward the fully-throttled destination and regenerating the packet toward non-fully-throttled destinations. Because we assume application layer and transport layer share the topology information, the packet injection process of the fully throttled router is paused until it is not fully 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 performance indices are not affected by the topology changing of the NSI-mesh.
Table I. Statistics of passing flits for 20,000 cycles

<table>
<thead>
<tr>
<th></th>
<th>Downward</th>
<th>TLAR-DLDR</th>
<th>TLAR-DLAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>149.68</td>
<td>1113.54</td>
<td>1381.71</td>
</tr>
<tr>
<td>L2</td>
<td>147.43</td>
<td>1100.68</td>
<td>1398.16</td>
</tr>
<tr>
<td>L3</td>
<td>148.16</td>
<td>1110.43</td>
<td>1366.34</td>
</tr>
<tr>
<td>L4</td>
<td>4960.81</td>
<td>3280.92</td>
<td>2523.75</td>
</tr>
<tr>
<td>Total</td>
<td>1475.98</td>
<td>1707.58</td>
<td>1697.02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Stdv.</th>
<th>Stdv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>12.82</td>
<td>74.91</td>
<td>91.10</td>
</tr>
<tr>
<td>L2</td>
<td>11.98</td>
<td>70.75</td>
<td>92.20</td>
</tr>
<tr>
<td>L3</td>
<td>12.23</td>
<td>97.27</td>
<td>87.53</td>
</tr>
<tr>
<td>L4</td>
<td>280.24</td>
<td>185.30</td>
<td>142.57</td>
</tr>
<tr>
<td>Total</td>
<td>2160.52</td>
<td>980.44</td>
<td>522.49</td>
</tr>
</tbody>
</table>

Fig. 12 shows the latency versus total network injection rate of the proposed three algorithms of TLAR. Because of the more balanced loading of the network, the TLAR-DLDR and TLAR-DLAR both have better performance than the baseline reactive downward routing algorithm. Quantitatively the average latency of downward routing is reduced 57.5% and 63.7% respectively by TLAR-DLDR and TLAR-DLAR. The throughput of the network is improved to 1.47x and 1.67x by TLAR-DLDR and TLAR-DLAR.

V. CONCLUSION

In this paper, we introduce a new problem by defining the non-stationary irregular mesh. To improve the performance of reactive routing, we propose a new transport-layer assisted routing scheme and two routing algorithms. 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 for only one bit is required for each destination. From our experiments, the proposed TLAR can effectively balance the vertical load distribution. The average latency is reduced around 60%, and the network throughput is improved above 47%.

REFERENCE