Advertiser Elevator: A Fault Tolerant Routing Algorithm for Partially Connected 3D Network-on-Chips

Ebadollah Taheri, Mihailo Isakov, Ahmad Patooghy and Michel A. Kinsy
Adaptive and Secure Computing Systems (ASCS) Laboratory
Department of Electrical and Computer Engineering, Boston University

***Abstract*—In this paper, we propose an adaptive routing algorithm for vertically partially connected 3D NoCs to (1) overcome failures in vertical links, and (2) find the nearest available vertical link for rerouting of packets. To track the position of each vertical link and distance to the other nodes, the proposed routing algorithm, named Advertiser Elevator, indexes each vertical link and implements a mechanism for announcing and sharing these indexes with the other nodes of the network. Packets are routed toward the nearest vertical link based on received indexes. The routing algorithm tolerates vertical link failures by interpreting the absence of index messages from a vertical link node as a link failure at the node. Packets are rerouted around failed links based on collected messages. The performance of the Advertiser Elevator routing algorithm is evaluated using the Access Noxim NoC simulator under different network congestion levels and fault rates. The results show that the proposed routing algorithm (1) is able to deliver packets as long as there are at least four live vertical links in the network (e.g., corner links) and (2) improves the average network latency by 15% over the well-known Elevator-First routing algorithm.**

I. INTRODUCTION

Conventional communication frameworks such as point-to-point and bus-based communications do not scale with the increasing number of processing elements in multicore and system-on-chip (SoC) architectures [1]. As a consequence, network-on-chip (NoC) has emerged as an efficient, scalable communication infrastructure for these multi-processing element architectures. NoC-based chips, generally, have higher communication concurrency and performance with lower power utilization [2]. NoC-based architectures are often implemented using a tile-based approach and a mesh topology with logic-based dimensional-order routing due to their manufacturing and routing simplicity [2]. However, the continue increase in the number of processing elements in the 2D implementation of these architectures has resulted in higher average inter-node distance, longer routing delay, and more power consumption [3]. To mitigate these design issues, 3D IC technologies and 3D NoCs are being introduced. Under the same number of processing elements, 3D integration reduces the network diameter, routing delay, and power consumption of the chip [4]. Most 3D NoCs use *Through-Silicon Via* (TSV) as the vertical links to connect the different planes/layers of a 3D chip [5]. Although, TSV links have higher bandwidth than the electrical links, they have higher fabrication cost [6] and tend to have higher rates of failure [7]. Therefore, they must be used judiciously and optimally. As a solution to the fabrication costs of these vertical TSV links, system designers have proposed the use of vertically integrated, partially connected 3D NoC [6]. Different deadlock-free routing algorithms have been proposed for 3D NoCs [8], [9]. Currently, these routing algorithms have two major drawbacks: (1) the reduction in finding minimal paths, and (2) the need for extra hardware to prevent deadlock. Non-minimal path routing algorithms pose optimization problems in terms of the TSV link placements [10]. As mentioned above, reliability issues are more pronounced with the use of TSV links, not just for the links themselves but for the chip as a whole [4]. In order to compensate for these reliability problems, the use of TSV links in 3D NoCs must be tightly coupled with fault tolerant techniques [11]. In this work, we propose and evaluate a fault tolerant routing algorithm for vertically integrated partially connected 3D NoCs. The proposed routing algorithm is able to route network packets around failed TSV links and uses three virtual channels per physical channel to achieve deadlock-freedom.

II. RELATED WORK

The overall reliability of network-on-chip based architectures is closely related to that of their interconnected network. As a result, several research efforts have tried to address the reliability issue in NoC designs [12]. The proposed techniques fall in three categories: (1) fault avoidance, (2) fault masking, and (3) fault tolerance. The implementations of fault avoidance and fault masking schemes generally require considerable hardware overheads. Therefore, the common approach for NoC reliability is fault tolerance [13]. This design decision still holds for 3D NoCs. In 3D interconnect networks, in addition to detecting electrical link failures in the 2D planes, the fault tolerance mechanism must also try to detect failed TSVs and reroute traffic around them. To support the runtime rerouting decisions, adaptive routing algorithms are often adopted [6].

Fault Tolerant Routing Algorithms for Fully Connected 3D NoCs: Akbari et al. [11] introduced a fault tolerant routing algorithm, *AFRA*, for mesh based 3D NoCs with fully connected vertical links. The routing algorithm tries to tolerate failures on vertical links by bypassing failed vertical links. This is done by packet rerouting without using an extra virtual channel. However, this routing algorithm only tolerates faults of vertical links in one direction, i.e., the routing can only tolerate failure in upward or downward vertical links but not both. Although there are several other papers [4] proposing different fault tolerant routing algorithms for 3D NoCs, the

978-1-5090-6389-5/17/$31.00 ©2017 IEEE

current trend in prototyped or commercial 3D chips points to vertically partially connected topologies.

Fault Tolerant Routing Algorithms for Partially Connected 3D NoCs: Jiang et al. in [12] proposed a deadlock free routing algorithm that uses 2 virtual channels for vertically partially connected 3D NoCs. In this routing scheme, all the routers at a layer are aware of locations of all vertical links in that layer. However, the runtime computations of best vertical links lead to longer routing times and higher hardware overheads. In [14], authors proposed a routing algorithm without an extra virtual channel. This proposed algorithm imposes some constraints on the locations of vertical links to achieve deadlock-freedom. One downside to this approach is the fact that with a small number of TSVs, location constraints can render the algorithm unusable. Using two virtual channels, *Elevator-First* routing algorithm [6] improves the network performance while simplifying the routing complexity associated with vertically partially connected 3D NoCs. *Elevator-First* routing algorithm supports different topologies and TSV arrangement in the layer. However, lack of path diversity can be an issue leading to low redundant routes and in turn to no tolerance to TSV failure.

III. Advertiser Elevator routing algorithm

A common routing approach in vertically partially connected 3D NoCs is to first route packets in the horizontal plane toward the vertical links, called *elevators*, then second, route them vertically to reach the destination node/layer using these elevators. A key design decision in these routing algorithms is the assignment of the elevator links to packets. Assigning a fixed elevator to each source and destination pair, e.g., *Elevator-First* routing algorithm [6], has many advantages but it can also lead to lower path diversity and network reliability in the case of vertical link failures. To tolerate vertical link failures, one can build in the routing protocol mechanisms to (i) detect failed elevators and (ii) reroute packets around such elevators. However, runtime packet rerouting severely complicates the deadlock freedom policy. Another approach is to increase the number of elevators that a router may use in routing packets. Under this design, a router has more than one elevator address in its routing table or logic. When a faulty elevator is detected, routers start rerouting packets through other links. The detection of faulty elevators and the propagation on that information may add to the average routing delay and hardware costs.

A. Adaptive Selection of Elevator Links to Improve Reliability

In the proposed *Advertiser Elevator* algorithm, elevator links are assigned indexes based on their state of operability. Each router maintains its own copy of the index table. Through the elevator announcement process of the algorithm, elevator link locations, i.e., indexes are shared in the routers. Figure 1 shows the vertical link index sharing process. The nodes with healthy elevators are assigned the largest elevator index, in this example, 4. The elevator indexes are shared among routers using dedicated links between neighboring routers. After receiving elevator indexes from neighbors, each router selects the maximum index among the received indexes and its own elevator index and updates its index table accordingly. In the following cycle, the routers send their maximum received elevator indexes minus one to their neighbors. The liveness checking of the elevator links is done in a distributed fashion where each node informs on its local elevator link status. As shown in Algorithm 1, a node with a functioning elevator labels the elevator index as a 4 and sends the value to its neighbors. If a node receives an index value, e.g., 4, from a neighbor and its own local vertical link is faulty, then the node decrements the received index value, e.g., 4 becomes 3. In this way, nodes that are close to fault-free elevators have higher elevator indexes. The elevator indexes assist the routing algorithm find the best elevator for a given source and destination pair. In order to implement a fully adaptive routing without deadlock in the layers, packets are classified as southward and northward packets. The elevator indexes are shared in two directions, south and north directions, to distinguish southward and northward routing.

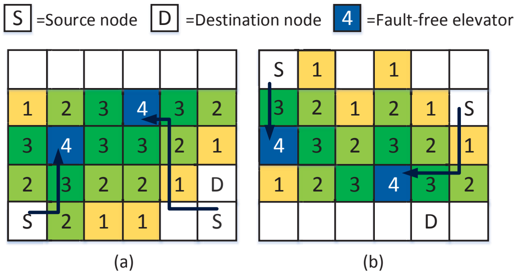

Fig. 1. Elevator indexes of each node for packet with (a) north destination and (b) south destination.

B. Virtual Channel (VC) Assignment

In vertically partially connected 3D NoCs, use of extra virtual channels is widely proposed to improve the network performance and provide deadlock freedom [6], [8], [9], [12]. To reach fully adaptive and deadlock-freedom, the *Advertiser Elevator* needs at least three virtual channels; two virtual channels to be fully adaptive in XY planes without any deadlock, plus one virtual channel for deadlock-freedom among packets which use X/Y channels after Z channel. The algorithm judiciously selects high index elevator routers in the source layer to send packets in the XY plane. Packets are routed through the southward and northward virtual networks based on their destinations. Two of the three VCs associated with each physical link are used to form the southward and northward virtual networks. Northward packets are put on the first virtual network and use VC 0. Similarly, southward packets are assigned to the second virtual network and VC 1. Packets with destinations outside the source layer are routed in the source layer using VCs 0 and 1, and are routed in the destination layer through VC 2. Based on the turn model for the proposed routing algorithm, when a packet leaves its source layer for a lower layer or enters its destination layer for an upper layer, it switches to the third VC.

Algorithm 1 Update the elevator indexes for each node.

```
$S\_U$: South elevator Index to Up
$S\_D$: South elevator Index to Down
$N\_U$: North elevator Index to Up
$N\_D$: North elevator Index to Down
 1: all local elevator indexes $\leftarrow 0$
 2: if (router has fault-free elevator to UP) then
 3:     $S\_U$ and $N\_U \leftarrow 4$
 4: end if
 5: if (router has fault-free elevator to Down) then
 6:     $S\_D$ and $N\_D \leftarrow 4$
 7: end if
 8: send local elevator index minus 1 to the neighbor nodes
 9: while (5 cycles after the update starts) do
10:     if (received $S\_U >$ local $S\_U$) then
11:         local $S\_U \leftarrow$ the received $S\_U$
12:     end if
13:     if (received $S\_D >$ local $S\_D$) then
14:         local $S\_D \leftarrow$ the received $S\_D$
15:     end if
16:     if (received $N\_U >$ local $N\_U$) then
17:         local $N\_U \leftarrow$ the received $N\_U$
18:     end if
19:     if (received $N\_D >$ local $N\_D$) then
20:         local $N\_D \leftarrow$ the received $N\_D$
21:     end if
22:     send local elevator index minus 1 to the neighbor nodes
23: end while
```

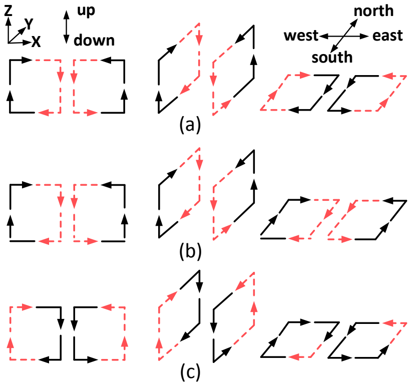

Fig. 2. Turn model of the first virtual channel (a), the second virtual channel (b), and the third virtual channel (c).

C. Advertiser Elevator Algorithm and Deadlock Discussion

As presented in Algorithm 2, the *Advertiser Elevator* algorithm is a two-stage routing protocol. In the first stage, packets are routed to the destination layer (cf. lines 7 to 25). In the second stage, packets are routed within the destination layer to destination node (cf. lines 1 to 6). If a packet is coming from an upper layer, it needs to use the third VC in its destination layer (cf. lines 2-4). Upward turns in the XY plane (i.e., Up-North, Up-East, Up-South, and Up-West) are prohibited when a packet is assigned to the third VC. In the first and second VCs, for deadlock freedom purposes, the *Advertiser Elevator* algorithm prohibits lateral-downward turns (i.e., North-Down, East-Down, South-Down, West-Down) - cf. line 10. The algorithm routes packets upward or downward depending on the availability of up and down elevators (cf. lines 7-11). Packets in XY plane are routed according to their virtual networks and the position of fault-free elevators. The routing directions are based on elevator indexes (cf. line 24) and allowed virtual network turns (cf. lines 12-18). Similar to other papers [12], [14], the *Advertiser Elevator* algorithm views communication links between two nodes $n(x, y, l_i)$ and $n(x, y, l_j)$ in adjacent horizontal planes/layers l_i and l_j as up/down elevators and all the elevators at $n(x, y, \_)$ locations in the network are part of the (x, y) pillar. Unlike AFRA routing algorithm [11], where an elevator failure renders the whole pillar unusable, in the proposed approach the liveness of the up/down elevators forming the pillar are decoupled. Each node is aware of the failure or liveness of its up/down elevator. If an elevator in a pillar is faulty, the *Advertiser Elevator* algorithm marks that elevator as unusable, but keeps the other elevators in the pillar available for routing traffic. This optimization of the protocol can lead to two special routing cases. In the first scenario, a packet is either in the first or second VC (a downward packet) and needs to be routed through some middle layers (non-source and non-destination layers). This particular situation arises when a packet encounters a faulty elevator on its path. In such a case, the algorithm selects a routing path using the same set of rules as in the source layer. In the second scenario, the packet is already using the third VC (an upward packet), in this case, west-first routing mode is used to forward the packet (cf. lines 12-13).

In general, livelock may occur when a non-minimal routing algorithm like *Advertiser Elevator* is used. Therefore, for livelock avoidance, *Advertiser Elevator* algorithm (a) prohibits 180° turns in the east-west and west-east routing modes and (b) does not allow the corner nodes to have both up and down elevators. As shown in Figure 2 the proposed algorithm by using three VCs is deadlock free. In summary, packets using VC 0 or 1 may switch to the third VC (2). But if a packet is already using VC 2, then there is no alternative switching VC. This approach ensures that there are no cyclic dependencies in the virtual channel allocation process. It is worth noting that although the algorithm is presented with three virtual channels $0-2$, any number of VCs greater than three can be used and the deadlock-freedom property of the algorithm is still preserved if those VCs are divided into three sets. Without lost of generality, we assume an orthogonal topology network and XY-planes of four corners. Through different heuristics and simulation settings, we examine a number of optimal or near-optimal elevator placement schemes. Although a detailed report of the study is beyond the scope of this paper, it is worth noting that the average probability of finding the best elevator is 99.2% for an $8 \times 8 \times X$ network using 2-bit index transmission bandwidth.

IV. Evaluations

To evaluate the *Advertiser Elevator* routing algorithm, we use the *Access Noxim* [15] simulator. We compare our *Advertiser Elevator* algorithm with the *Elevator-First* routing algorithm [6] using an $8 \times 8 \times 4$ network. Three networks with different number of elevators were considered in our evaluations. We set the packet size to 8 flits, the VC depth to

Algorithm 2 Proposed Routing Algorithm.

```
if (current is in the destination layer) then
    if (packet is not in the third virtual channel) then
        switch to the third virtual channel
    end if
    return XY routing to destination
else
    if (upward routing is needed and an upward fault-free elevator exists) then
        return Up direction
    else if (downward routing is needed and a downward fault-free elevator exists) then
        switch to the third virtual channel
        return Down direction
    else if (packet is in the third virtual channel) then
        Directions ← east, south and north (if exist)
    else if (packet is in the first virtual channel) then
        Directions ← south, east and west (if exist)
    else if (packet is in the second virtual channel) then
        Directions ← north, east and west (if exist)
    end if
    if (direction in previous routing is west) then
        remove east from Directions
    else if (direction in previous routing is east) then
        remove west from Directions
    end if
    return direction with largest index in Directions
end if
```

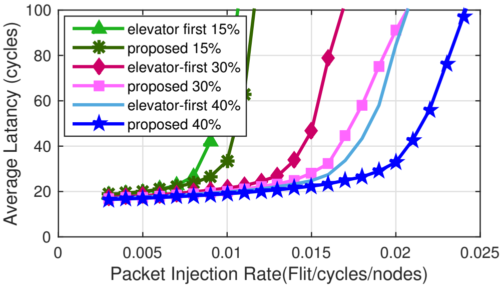

Fig. 3. Average latency of the proposed routing algorithm.

4, the fault injection and elevator placements to random, and the traffic pattern to random uniform. The results presented in Figure 3 show that the *Advertiser Elevator* routing algorithm improves the average latency by 14-16% in three networks. As shown in Figure 4, the proposed routing is able to tolerate vertical link faults at a lower average latency. Injected faults disable vertical links between two adjacent layers. If a vertical link becomes faulty, the vertical links in other layers belonging to the same pillar are not necessarily faulty. For fault injection rates of 7%, 14% and 28% the average network latencies are 17%, 21% and 42% lower when compared to the *Elevator-First* routing algorithm.

V. Conlusion

This paper proposes a fault tolerant routing algorithm for vertically partially connected 3D NoCs, named *Advertiser Elevator*. The proposed algorithm utilizes the assigned elevator indexes to find the best fault-free vertical link for routing packets. Using dedicated links/extra bits, fault-free elevators share their elevator indexes with the rest of the nodes in the network. With only two bits for the elevator index sharing, the probability of selecting the best elevator is above 98%. Simulation results show an average latency improvement of ∼14% over the *Elevator-First* algorithm.

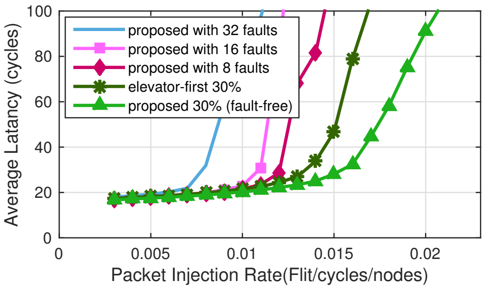

Fig. 4. Average latency with fault injection.

References

[1] W. Dally and B. Towles, *Principles and practices of interconnection networks*. Morgan Kaufmann, 2004.

[2] Z. Zhang, A. Greiner, and S. Taktak, "A reconfigurable routing algorithm for a fault-tolerant 2d-mesh network-on-chip," in *Proceedings of the 45th annual Design Automation Conference*. ACM, 2008, pp. 441–446.

[3] J. H. Lau, "Evolution, challenge, and outlook of tsv, 3d ic integration and 3d silicon integration," in *Advanced Packaging Materials (APM), 2011 International Symposium on*. IEEE, 2011, pp. 462–488.

[4] S. Pasricha and Y. Zou, "A low overhead fault tolerant routing scheme for 3d networks-on-chip," in *Quality Electronic Design (ISQED), 2011 12th International Symposium on*. IEEE, 2011, pp. 1–8.

[5] E. Beyne, P. D. Moor, W. Ruythooren, R. Labie, A. Jourdain, H. Tilmans, D. S. Tezcan, P. Soussan, B. Swinnen, and R. Cartuyvels, "Through-silicon via and die stacking technologies for microsystems-integration," in *Electron Devices Meeting, 2008. IEDM 2008. IEEE International*, Dec 2008, pp. 1–4.

[6] F. Dubois, A. Sheibanyrad, F. Petrot, and M. Bahmani, "Elevator-first: A deadlock-free distributed routing algorithm for vertically partially connected 3d-nocs," *Computers, IEEE Transactions on*, vol. 62, no. 3, pp. 609–615, 2013.

[7] L. Jiang, Q. Xu, and B. Eklow, "On effective tsv repair for 3d-stacked ics," in *2012 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2012, pp. 793–798.

[8] R. Salamat, M. Khayambashi, M. Ebrahimi, and N. Bagherzadeh, "A resilient routing algorithm with formal reliability analysis for partially connected 3d-nocs," *IEEE Transactions on Computers*, vol. PP, no. 99, pp. 1–1, 2016.

[9] E. Taheri, A. Patooghy, and K. Mohammadi, "Cool elevator: A thermal-aware routing algorithm for partially connected 3d nocs," in *Computer and Knowledge Engineering (ICCKE), 2016 6th International Conference on*. IEEE, 2016, pp. 111–116.

[10] S. Foroutan, A. Sheibanyrad, and F. Petrot, "Assignment of vertical-links to routers in vertically-partially-connected 3-d-nocs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 8, pp. 1208–1218, Aug 2014.

[11] S. Akbari, A. Shafiee, M. Fathy, and R. Berangi, "Afra: A low cost high performance reliable routing for 3d mesh nocs," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012*. IEEE, 2012, pp. 332–337.

[12] X. Jiang, L. Zeng, and T. Watanabe, "A sophisticated routing algorithm in 3d noc with fixed tsvs for low energy and latency," *IPSJ Transactions on System LSI Design Methodology*, vol. 7, no. 0, pp. 101–109, 2014.

[13] P. Ren, X. Ren, S. Sane, M. A. Kinsy, and N. Zheng, "A deadlock-free and connectivity-guaranteed methodology for achieving fault-tolerance in on-chip networks," *IEEE Transactions on Computers*, vol. 65, no. 2, pp. 353–366, Feb 2016.

[14] H. Ying, A. Jaiswal, and K. Hofmann, "Deadlock-free routing algorithms for 3-dimension networks-on-chip with reduced vertical channel density topologies," in *High Performance Computing and Simulation (HPCS), 2012 International Conference on*. IEEE, 2012, pp. 268–274.

[15] K.-Y. Jheng, C.-H. Chao, H.-Y. Wang, and A.-Y. Wu, "Traffic-thermal mutual-coupling co-simulation platform for three-dimensional network-on-chip," in *VLSI Design Automation and Test (VLSI-DAT), 2010 International Symposium on*. IEEE, 2010, pp. 135–138.