2018 30th International Teletraffic Congress

RB-OLITS: A Worst-case Reorder Buffer Size Reduction Method for 3D-NoC based on Ohm's Law-like Traffic Splitting

Zhenmin Li*, Renshi Fan†, Gaoming Du†, Yifan Cao†, Zhonghai Lu‡, Minglun Gao†,
* School of Microelectronics, Hefei University of Technology, Hefei, China. 230009.
† Institute of VLSI design, Hefei University of Technology, Hefei, China. 230009. Email: dugaoming@hfut.edu.cn
‡ Department of Electronic Systems, KTH Royal Institute of Technology, Stockholm, Sweden.

***Abstract*—Higher link bandwidth provided by three-dimensional Network-on-Chip (3D-NoC) relies on multi-path routing and traffic splitting, which inevitably leads to out-of-order packet delivery problem when facing unbalanced traffic congestion. Therefore, a reordering mechanism, often requiring reorder buffers, is necessary to guarantee in-order delivery of packets. The reorder buffers are expensive in terms of both resource and power consumption, hence it is of great significance to reduce the reorder buffer size in the worst case. In this paper, we propose a improved Ohm's Law-like traffic splitting method, named RB-OLITS, for reducing the worst-case reorder buffer size. The traffic splitting configuration is obtained according to the parallel resistance theory. Experimental results show that RB-OLITS has reduced the worst-case reorder buffer size by 19.09% in average compared with OLITS. Meanwhile, the worst-case delay bound is not degraded by using RB-OLITS, as the worst-case delay bound of the same target flow is slightly improved by 1.59% in average.**

I. INTRODUCTION

Recently, three-dimensional Network-on-Chip (3D-NoC) architecture paradigm is emerging as the most promising successor to the traditional bus interconnect architecture in Multiprocessor System-on-Chip (MPSoC) designs [1]. The packet-based 3D-NoC paradigm is highly scalable and fundamentally decouples communication from computation. A 3D-NoC is a collection of on-chip resources such as Processing Elements (PEs), Memory Elements (MEs) and peripheral components. These resources are typically interconnected using routers, to deliver the data (packets) from one node to another. A Network Interface (NI) is attached to each router, translating messages between resource element and router according to a standard communication protocol, hereby linking on-chip resources with network infrastructure.

In-order packet delivery is crucial for a majority of applications, for instance, multimedia or cache coherence protocols. However, due to the complexity of network congestion, the packets arriving at the destination may cause the out-of-order problem. Therefore, guaranteeing that the packets are transmitted in an orderly manner has become a pivotal research topic. Yang *et al.* [2] proposed a reordering mechanism based on a lookup table, but the reorder buffer is statically partitioned, leading to a low utilization rate. Murali *et al.* [3] used flow control method to achieve in-order delivery. However, this method requires high resource consumption and overestimates the traffic congestion. Ebrahimi *et al.* [4] proposed a dynamic buffer allocation structure to improve utilization and overall NoC performance. Another in-order packet delivery method utilizing reorder buffers has been proposed in [5], where network calculus is used to determine the worst-case reorder buffer size.

An Ohm's Law-like traffic splitting method, named OLITS, [6] has been proposed to determine the suitable traffic splitting configuration for each node along the route, which could result in an improved worst-case delay bound. However, this method does not address the issue of the worst-case reorder buffer size reduction. It is of great significance to economize on reorder buffer since they are expensive in terms of both resource and power consumption, without deteriorating the worst-case performance of NoC. This research gap motivates us to find an efficient traffic splitting method based on global congestion predication, in order to minimize the worst-case reorder buffer size.

In this paper, we propose a novel traffic splitting method, named Reorder Buffer - Ohm's Law-like Traffic Splitting (RB-OLITS), that results in reduced worst-case reorder buffer size. We redefine the flow resistance as the flow congestion factor extracted from the contention matrix, in order to indicate the relation between flow congestion and the worst-case reorder buffer size. To the best of our knowledge, this is the first work to find the traffic splitting configuration that results in minimized the worst-case reorder buffer size based on global congestion predication and network calculus for 3D-NoC. Note here, the deadlock freeness of the network is closely related with the routing algorithm and is beyond the scope of this paper.

This rest of this paper is organized as follows. Section II describes the background needed for the sequel. Section III gives the details of RB-OLITS method for reducing the worst case reorder buffer size. The experimental results are shown and discussed in Section IV. Finally, we conclude the paper in Section V.

II. PRELIMINARIES

In this section, we firstly introduce the target architecture of 3D-NoC used in this paper, followed by the introduction of network calculus and worst-case reorder buffer calculation required to present our method.

978-0-9883045-5-0/18/$31.00 ©2018 ITC
DOI 10.1109/ITC30.2018.00024

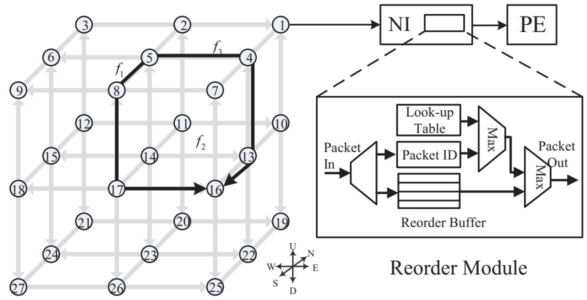

Fig. 1. 3D-NoC architecture model with a running example

A. 3D-NoC Architecture Model

We present the 3D-NoC architecture with 3D mesh topology in Fig. 1. Each circle in the 3D mesh architecture denotes a router that is capable of performing multi-path minimal routing function. Each router is connected to a local component, typically a PE, through NI. The reorder module is located within NI, which is responsible of reordering the packets in a orderly manner before forwarding them to the PE. Therefore, the PEs are ignorant about the reordering procedure, so that the modularity and scalability of our 3D-NoC architecture are maintained. An example of traffic splitting is shown in Fig. 1, with two different sub-flows from node 5 (source) to node 16 (destination) with minimal multi-path routing.

The reorder module performs reordering by fetching the packets in the correct order, each at a time, from the one with the smallest packet sequence number. When the reorder module receives a new data packet, it compares the packet sequence number with the one that is required. If matched, the packet will be directly sent to PE. Otherwise, it will be stored in the reorder buffer. The corresponding address of this packet, together with its packet sequence number, are stored in the look-up table. Once the newly received packet matches with the requiring packet sequence number, a search of the next required packet inside the reorder buffer is triggered, with the help of the look-up table to accelerate the searching process. Finally, all the packets are sent to the PE in the correct order. The details of the reorder module circuit are beyond the scope of this paper and are left for the future work.

B. Network Calculus Basics

A traffic flow in a NoC, denoted by f, represents an infinite flow of unicast traffic sent from the source node to the destination node. The reception and forwarding of data packets of f from the perspective of each router can be characterized by the arrival and service curve, respectively. The former defines the arrival process of a service flow, and the latter describes the output behaviour of a network node. The arrival curve is given by $\alpha(t) = rt + b$, where b indicates the burstiness and r is the average packet generation rate. The service curve is give by a latency-rate function for NoC routers, i.e., $\beta_{R,T} = R(t - T)^+$, where R is the minimum service rate, and T is the maximum processing latency of the network element. Here x^+ equals 0 if $x \leq 0$, otherwise x. Finally, the upper delay bound is calculated by [7]:

$$\bar{D} = \frac{b}{R} + T \tag{1}$$

C. Worst-case Reorder Buffer Size Calculation

According to [5], with fixed packet delay and uniform injection rate, let the transient delays of packets for flow f_1 and f_2 be D_1 and D_2, and the packet injection intervals be Δt_1 and Δt_2, respectively. Hence, the buffer size S_{rb} can be calculated as follows:

$$S_{rb} = \frac{|D_1 - D_2|}{\Delta t}. \tag{2}$$

If $D_1 > D_2$, then $\Delta t = \Delta t_2$; if $D_1 < D_2$, $\Delta t = \Delta t_1$; otherwise, S_{rb} equals zero, meaning no packet reordering is required.

In order to obtain the worst-case buffer size, let the upper bounds of D_1 and D_2 be $\bar{D_1}$ and $\bar{D_2}$, and the lower bounds be $\underline{D}_1$ and $\underline{D}_2$, respectively. Then the worst-case buffer size, denoted by S_{rb}^{max}, with uniform injection rate, can be given by:

$$S_{rb}^{max} = max\left\{\frac{(\bar{D_1} - \underline{D}_2)}{\Delta t_2}, \frac{(\bar{D_2} - \underline{D}_1)}{\Delta t_1}\right\} \tag{3}$$

According to network calculus, the entire on-chip network is further abstracted into integration of sub-systems, and each sub-flow corresponds to a sub-system S. Let the input arrival curves of sub-systems S_1 and S_2 be $\alpha_1(t)$ and $\alpha_2(t)$, respectively. Similarly, let the output arrival curves be $\alpha_1^*(t)$ and $\alpha_2^*(t)$, the service curve be $\beta_1(t)$ and $\beta_2(t)$, and the delay bound be $\bar{D_1}$, $\bar{D_2}$, respectively. Here, $\bar{D_1} = h(\alpha_1, \beta_1)$, $\bar{D_2} = h(\alpha_2, \beta_2)$. If $\Delta t_1 = h(\alpha_2, \beta_2) - \underline{D}_1$, $\Delta t_2 = h(\alpha_1, \beta_1) - \underline{D}_2$, then the worst-case reorder buffer size can be calculated as:

$$S_{rb}^{max} = max\{\alpha_1^*(\Delta t_1), \alpha_2^*(\Delta t_2)\} \tag{4}$$

The calculation of $h(\alpha_m, \beta_m)$ are given in [8], which requires deriving the Equivalent Service Curve (ESC) for multi-path routing NoC. In order to generalize, let the arrival curves of two sub-flows be $\alpha_1(t) = \rho_1 r_1' t + \rho_1 b_1'$ and $\alpha_2(t) = \rho_2 r_2' t + \rho_2 b_2'$, and the corresponding service curves be $\beta_1 = R_1(t - T_1)^+$ and $\beta_2 = R_2(t - T_2)^+$, respectively. Here, ρ_1 and ρ_2 represent the splitting proportions of sub-flow f_1 and f_2. For simplicity, assuming $r_1 = \rho_1 r_1'$, $r_2 = \rho_2 r_2'$, $b_1 = \rho_1 b_1'$ and $b_2 = \rho_2 b_2'$, then the output arrival curves of both flows can be given by $\alpha_1^*(t) = \alpha_1 \oslash \beta_1(t) = r_1 t + b_1 + r_1 T_1$ and $\alpha_2^*(t) = \alpha_2 \oslash \beta_2(t) = r_2 t + b_2 + r_2 T_2$, respectively. According to Equation 1, the upper delay bounds are $\bar{D_1} = h(\alpha_1, \beta_1) = T_1 + b_1/R_1$ and $\bar{D_2} = h(\alpha_2, \beta_2) = T_2 + b_2/R_2$, and the lower delay bounds are $\underline{D}_1 = T_1$ and $\underline{D}_2 = T_2$. By rewriting Equation 4, the worst-case reorder buffer size is determined by:

$$\begin{aligned} S_{rb}^{max} \quad = \quad & max\{r_1(T_2 + b_2/R_2) + b_1, \\ & r_2(T_1 + b_1/R_1) + b_2\}. \end{aligned} \tag{5}$$

III. RB-OLITS TRAFFIC SPLITTING METHOD

The OLITS method [6] employs the adjacency matrix to derive a contention matrix containing the congestion information from the perspective of a designated target flow. Then the congestion factors can be calculated. The congestion factors, considered as a series of parallel resistances, are used to predicate the traffic state. Based on the parallel resistance theory, a proper target flow splitting configuration can be calculated leading to better worst-case reorder buffer size. Based on OLITS, we propose RB-OLITS by redefining the congestion factor to improve the effect on worst-case reorder buffer size reduction. Firstly, we give the definitions of adjacency matrix, contention matrix, and congestion factor, which are required for presenting RB-OLITS.

1) Adjacency Matrix: In order to capture the local traffic splitting state, the adjacency matrix is defined as [6]:

$$A = (a_{ij})_{v\times 6} \tag{6}$$

$$a_{ij} = \begin{cases} p_{ij}, & \text{splitting propotion in } j \text{ direction at node } v_i \\ 0, & \text{else} \end{cases} \tag{7}$$

2) Contention Matrix: We employ a contention matrix to capture the global network congestion information. In a contention matrix, it is assumed that there are k flows, $f_{\langle s_1,d_1\rangle}$, $f_{\langle s_2,d_2\rangle}$, $\cdots$, and $f_{\langle s_k,d_k\rangle}$, where s_i and d_i are the source and destination node, respectively. Take $f_{\langle s_i,d_i\rangle}$ as the target flow, and the other flows are contention flows. The final contention matrix C_{s_i,d_i} is given as:

$$\begin{aligned} C_{s_i,d_i} =& [(\varepsilon r_1 + \beta b_1)A_{s_1,d_1} + (\varepsilon r_2 + \beta b_2)A_{s_2,d_2} + \cdots \\ &+ (\varepsilon r_j + \beta b_j)A_{s_j,d_j} + \cdots \\ &+ (\varepsilon r_k + \beta b_k)A_{s_k,d_k}] \wedge A_{s_i,d_i}, j \neq i \end{aligned} \tag{8}$$

where $\wedge$ means the minimum operation, i.e., $f \wedge g = \min\{f,g\}$. For simplicity,

$$\begin{aligned} C_{s_i,d_i} =& [B_{s_1,d_1} + B_{s_2,d_2} + \cdots \\ &+ B_{s_j,d_j} + \cdots B_{s_k,d_k}] \wedge A_{s_i,d_i}, j \neq i \end{aligned} \tag{9}$$

Here, r_i and b_i are flow rate and burstiness of sub-flow i, respectively. And ε and β are the impact factor of r_i and b_i respectively, which are used to describe the congestion state more accurately. Thus, B_{s_i,d_i} is the improved adjacency matrix containing global traffic congestion information:

$$B_{s_j,d_j} = (\varepsilon r_j + \beta b_j)A_{s_j,d_j} \tag{10}$$

3) Congestion Factor: Based on the contention matrix, we use a congestion factor to predicate the congestion state of a sub-flow. Therefore, the congestion factor of sub-flow f_i, termed λ_{f_i}, is defined as the total contention on f_i, which can be calculated as the sum of the contention matrix elements that the sub-flow traverses:

$$\lambda_{f_i} = \sum_{i=1}^{N} C(node\_i, direction\_j) \tag{11}$$

Here, N is the total number of nodes in the sub-flow, $node\_i$ is the i-th node along the sub-flow, and $direction\_j$ is the direction next to $node\_i$.

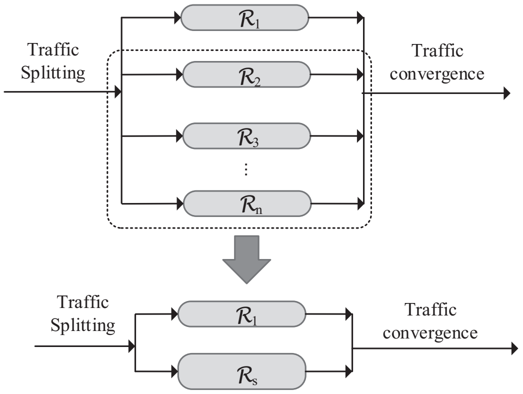

Fig. 2. Illustration of RB-OLITS traffic splitting

4) RB-OLITS: According to Equation 5, the worst-case reorder buffer size is affected by r_1T_2 or r_2T_1. Both items contain factor ρ, which is the splitting proportion of the sub-flow, indicating that the worst-case reorder buffer size is affected by the square of the congestion factor. Based on this observation, we intuitively replace the original congestion factor with its square when calculating the traffic splitting proportions. As Fig. 2 shows, $\mathcal{R}_1 \cdots \mathcal{R}_n$ denote the congestion factor λ_{f_1} to λ_{f_n}, representing the resistance of sub-flow f_1 to f_n, respectively. Based on parallel resistance theory, we can calculate the equivalent resistance of $\mathcal{R}_2 \cdots \mathcal{R}_n$, denoted by $\mathcal{R}_s$, as follows:

$$\mathcal{R}_s = \sqrt{1/((\frac{1}{\mathcal{R}_2})^2 + (\frac{1}{\mathcal{R}_3})^2 + \cdots (\frac{1}{\mathcal{R}_n})^2)} \tag{12}$$

Then, we can calculate the splitting proportion of sub-flow $f_{\langle s,d\rangle}$, termed $\mathcal{P}_{\langle s,d\rangle}$, as follows:

$$\mathcal{P}_{\langle s,d\rangle} = \frac{\mathcal{R}_s^2}{\mathcal{R}_1^2 + \mathcal{R}_s^2} \tag{13}$$

The splitting configuration is captured by $\mathcal{P}_{\langle s,d\rangle}$ for sub-flow $f_{\langle s,d\rangle}$. The traffic splitting proportions of other sub-flows can be determined in the same way. Since the proposed method is based on OLITS, we name it Reorder Buffer OLITS (RB-OLITS). This simple modification leads to significant improvement in terms of the worst-case reorder buffer size, without degrading the worst-case delay bound. The experimental results are shown in the next section.

IV. EXPERIMENTS AND RESULTS

In order to evaluate the proposed RB-OLITS method, both synthetic benchmarks and industry patterns are used. The results obtained using RB-OLITS are compared with those with OLITS [6]. We employ the same $3 \times 3 \times 3$ 3D-NoC with mesh topology in Fig. 1 as our target architecture. For synthetic pattern, two target flows are injected into the network. For target flow $f(8, 12)$, an reorder buffer is placed at node 12, and its buffer size is calculated using various configurations.

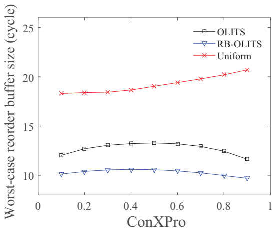

Fig. 3. Worst-case reorder buffer size comparison with varying splitting proportion in X direction for contention flows

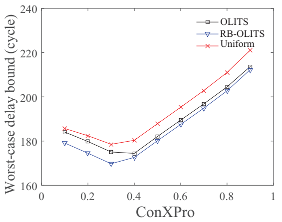

Fig. 4. Worst-case delay bound comparison with varying splitting proportion in X direction for contention flows

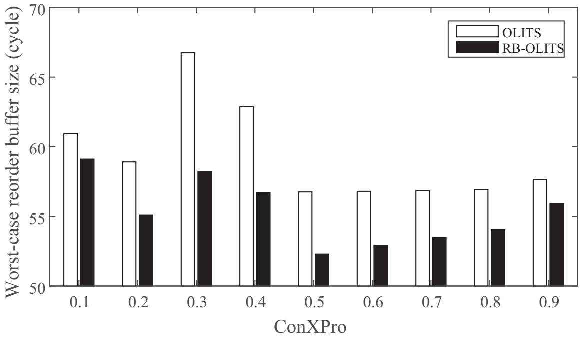

Fig. 5. Worst-case reorder buffer size comparison with varying splitting proportion in X direction for contention flows of DVOPD

A. *Synthetic Patterns*

We prepared two synthetic test cases where the contention flows are split using different splitting strategies. Besides the worst-case reorder buffer size, the worst-case delay bound is also calculated to evaluate our method in terms of worst-case performance.

Firstly, we set up three contention flows in the network, namely $f(4, 20)$, $f(8, 19)$, $f(4, 23)$. We alter the splitting proportion of the contention flow from 0.1 to 0.9 of X axis, whereas the Y and Z directions get half of the rest packets. The target flow are split on every viable node in the network, creating 30 sub-flows. The traffic splitting configuration is decided using RB-OLITS, OLITS and uniform paradigm, respectively. The results of worst-case reorder buffer sizes are given in Fig. 3. Compared with the uniform splitting paradigm, the RB-OLITS obtains a maximum improvement of 53.20%, and an average improvement of 46.35%. Compared with OLITS, RB-OLITS achieves a maximum improvement of 20.88%, with 19.09% improvement on average. In addition, the results of worst-case delay bounds are shown in Fig. 4. On average, RB-OLITS obtains an improvement of 4.11% compared with uniform splitting method, and 1.59% compared with OLITS.

B. *Industry Pattern*

We use an industry pattern DVOPD [9], as an example to further gauge the effectiveness of RB-OLITS. We mapped DVOPD application onto our 3D-NoC architecture. There are 43 contention flows in total, which establish a more complex on-chip traffic scenario. Uniform splitting strategy is also adopted for all contention flows, i.e., the splitting proportion is 0.33 in any direction. The service curve of the router is $\beta(t) = 0.33 \cdot (t - 3)^+$, where 0.33 is the traffic flow rate, meaning that the router is able to process one packet in every three cycles.

We change the splitting proportion of the contention flow from 0.1 to 0.9 of X axis, whereas the Y and Z directions get half of the rest packets. The target flow is split into 12 sub-flows, with the splitting configuration generated by RB-OLITS and OLITS, respectively. The results of the worst-case reorder buffer size are shown in Fig. 5. Compared with OLITS, the improvement when using RB-OLITS reaches 12.80% when the splitting proportion in X direction is 0.3, which is maximum. On average, RB-OLITS outperforms OLITS by 6.76%.

V. CONCLUSION

In this work we propose a traffic splitting method, named RB-OLITS, to reduce worst-case reorder buffer size. The traffic splitting configuration is obtained according to the parallel resistance theory. The experimental results show that RB-OLITS effectively reduces the worst-case reorder buffer size, with improvement of 19.09% in average compared with OLITS. Moreover, our proposed method is capable of maintaining the worst-case delay bounds for 3D-NoC, with an improvement of 1.59% in average.

REFERENCES

[1] B. S. Feero and P. P. Pande, “Networks-on-chip in a three-dimensional environment: A performance evaluation,” *IEEE Transactions on computers*, vol. 58, no. 1, pp. 32–45, 2009.

[2] X. Yang, Z. Qing-li, F. Fang-fa, Y. Ming-yan, and L. Cheng, “Nisar: An axi compliant on-chip ni architecture offering transaction reordering processing,” in *ASIC, 2007. ASICON’07. 7th International Conference on*, pp. 890–893, IEEE, 2007.

[3] S. Murali, D. Atienza, L. Benini, and G. De Micheli, “A method for routing packets across multiple paths in nocs with in-order delivery and fault-tolerance gaurantees,” *VLSi DeSign*, vol. 2007, 2007.

[4] M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila, and H. Tenhunen, “A high-performance network interface architecture for nocs using reorder buffer sharing,” in *Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on*, pp. 546–550, IEEE, 2010.

[5] G. Du, M. Li, Z. Lu, M. Gao, and C. Wang, “An analytical model for worst-case reorder buffer size of multi-path minimal routing nocs,” in *Networks-on-Chip (NoCS), 2014 Eighth IEEE/ACM International Symposium on*, pp. 49–56, IEEE, 2014.

[6] G. Du, Y. Ou, X. Li, P. Song, Z. Lu, and M. Gao, “Olits: An ohm’s law-like traffic splitting model based on congestion prediction,” in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2016*, pp. 1000–1005, IEEE, 2016.

[7] L. Lenzini, L. Martorini, E. Mingozzi, and G. Stea, “Tight end-to-end per-flow delay bounds in fifo multiplexing sink-tree networks,” *Performance Evaluation*, vol. 63, no. 9-10, pp. 956–987, 2006.

[8] G. Du, C. Zhang, Z. Lu, A. Saggio, and M. Gao, “Worst-case performance analysis of 2-d mesh nocs using multi-path minimal routing,” in *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, pp. 123–132, ACM, 2012.

[9] P. K. Sahu and S. Chattopadhyay, “A survey on application mapping strategies for network-on-chip design,” *Journal of Systems Architecture*, vol. 59, no. 1, pp. 60–76, 2013.