Automated Power and Latency Management in Heterogeneous 3D NoCs

Awet Yemane WeldeZion†; Masoumeh Ebrahimi†*; Masoud Daneshtalab†; Hannu Tenhunen†*
†KTH Royal Institute of Technology, Sweden; *University of Turku, Finland
{aywe;masebr;masdan;hannu}@kth.se

ABSTRACT

Beside different core sizes in many-core Systems-on-Chip, the cost and reliability issues of TSVs move 3D NoCs toward heterogonous designs. Such heterogeneity introduces design complexity and new challenges for obtaining a high performance, low power, low area, and a reliable design. By taking all these factors into account, we propose a design as a combination of Q-Learning and deflection routing in a heterogeneous 3D NoCs. This design enables the routing algorithm to dynamically adjust itself to the underlying traffic condition and topology arrangement at run time. Thereby, the network can reach its optimal performance and minimum power consumption shortly after a reconfiguration either because of an occurred fault in the network or a traffic change.

1. INTRODUCTION

Networks-on-Chip consists of an interconnection of routers to enable a large number of cores to communicate with each other. The parallelization offered by NoC combined with the possibility of stacking heterogeneous cores in a 3DIC allows a wide range of applications to be applied to the heterogeneous 3D NoC platform. One of the case examples is the 3D processor-memory stacking with Wide-IO standard using TSVs [1]. Designing deadlock-free routing algorithms for NoCs has been always a major topic as it is a main factor in achieving efficiency. To prove deadlock freeness in wormhole switching network, some routing limitations are applied by the means of turn models. These limitations, even small, strongly limit the flexibility of routing packets. For example, a single forbidden turn prevents packets to take some minimal paths and consequently a large number of non-minimal routes [2]. Routing algorithms are specially become very challenging in irregular networks or when some routers or links are disabled due to faults [3]. In 3D NoCs, more complexities are introduced as cycles can be formed between and within layers [4]. To overcome the complexity of designing high-performance and fault-tolerant routing algorithm in 3D NoCs, we have directed our efforts toward deflection routing, enjoying an inherent fault-tolerant capability. In deflection routing, a message is divided into packets, and each packet is a single-flit long. Once injected into the network, it is by default sent to an output with the minimum distance to the destination. If this output is not available, the packet is deflected to the next best output option. Because of shape and size variability of cores, the connectivity between cores may take a form of an irregular mesh [1]. Here, heterogeneity in our context refers to the irregularity of the networks due to the different sizes of cores integrated in the system. Despite the necessity of heterogeneous 3D NoC designs, the main issue is that they cannot be easily optimized regarding the performance and power.

This implies that heuristic approaches should be applied for each topology configuration not only to provide a deadlock-free communication between the cores but also to reach a satisfactory level of performance and low power consumption. Proposing a heuristic approach for different configuration of heterogeneous 3D NoCs imposes huge costs and engineering efforts. On top of all difficulties, a heuristic approach leading to a high performance design may not be always possible. For example, a proper routing algorithm for a heterogeneous 3D NoC cannot be easily developed as the routing paths are not known at the design time. The reason is that the topological structures of different chip layers are not known by a chip vendor. Even by assuming such heuristic approaches, a fault may suddenly disturb the configuration and thus having a significant impact on the correct functionality, performance and power. In other words, conventional routing strategies cannot be efficiently applied to heterogeneous 3D NoCs, demanding intelligent methods to cope with the issues.

In this paper, we offer a general solution relaxing the main design challenges of obtaining an optimal performance, low power, low area and high reliability in heterogeneous 3D NoC platforms. Specifically, we address the challenges of routing in irregular networks, and runtime power and performance optimization of 2D/3D NoCs. By motivating the usage of deflection routing in heterogeneous 3D NoCs and applying the Q-Learning algorithm on top of that, we propose a method satisfying the following features:

- **Reliability:** We achieve reliability in a relatively simple way by applying deflection routing. Deflection routing allows packets to be delivered through any possible output channels at a router for as long as at least one link is functioning. This includes both horizontal and vertical connections. Thereby packets can reach to upper or lower layers regardless of the TSVs arrangement. Faulty routers can be easily tolerated as it can be seen as a new topology configuration.
- **Optimal performance and low power consumption:** The applied Q-Routing algorithm is able to adjust itself with the underlying topology at run time without a prior knowledge on the configuration. The Q-Routing algorithm can reach the near optimal performance and low power consumption shortly after the reconfiguration. The reconfiguration may happen for different reasons such as a fault in the network, a new mapped application, traffic change, or testing purposes. Extra power saving, around 39%, is obtained by removing buffers from the network compared to a conventional on-chip network [6].
- **Area constraint:** We choose a buffer-less platform where the most area-hungry part of the network, buffers, is removed from the router architecture. The proposed approach is integrated with buffer-less flow control fabric, reporting a large area savings up to 60% compared to conventional buffered networks [6].

2. PRELIMINARIES AND RELATED WORK

2.1 Deflection Routing

Deflection routing is a well-studied topic in the on-chip network field. Originally the term was used in [7], where the deflection routing is utilized in distributed communication systems to decrease the contention by sending packets through redundant links. In [6] has shown the power/performance aspect of buffer-less routers using deflection routing as compared to buffered ones. The main properties of deflection routing can be described as:

Deadlock and livelock freedom: Deflection routing is inherently deadlock free. In case of packets' contention, the priority is always given to the output ports leading to the shortest paths. However if not possible, the packet can be sent out through any other output ports. The priority of output ports is stored in a table which can be filled out at design time and changed at run time. As packets are never blocked, a deadlock-free network is guaranteed. Priority schemes in the arbitration unit avoid the livelock situation. In the age-based priority scheme, if a packet takes a longer time than the threshold value, its priority increases.

Deflection routing is inherently fault tolerant: Irregularity is often indirectly discussed in the fault-tolerant domain where by occurring faults in routers or links the network topology may change from a regular to an irregular one. In general, fault-tolerant methods offer solutions to protect the network in the case of faults but they make less attention to the performance of the irregular network. For example, turn models may cause a router to be unreachable even though the router is still connected to the neighboring routers through some channels. Thanks to the characteristics of deflection routing, as long as a router has at least one remaining physical connection, the router is reachable. The presented approach in [8] suggests solutions to design hardware-efficient routing methods for an irregular mesh. The focus of this approach is on static shortest-path routing while in our proposed approach, packets are able to choose both minimal and non-minimal with the given priority on minimal paths.

Compatibility with heterogeneous networks: Deflection routing is fully compatible with irregular networks. In heterogeneous systems with unequal core sizes and links length, the type of network topology that can be mapped is irregular. The deflection routing can be directly implemented to such irregular networks without any particular modifications. Under any topology configurations, all packets can be delivered to the destinations. However, to improve performance, the priority of output channels can be adjusted according to the shortest paths.

2.2 Q-Routing Methods

Q-Routing based models have been utilized in different domains [9][10], but there are few methods investigated in the context of on-chip networks [5][12][11][12][13][14]. HARAQ [5] takes advantages of Q-Routing in wormhole switching network to achieve traffic balance. FTDR-H [11] applies Q-Routing methods in the network with the aim of tolerating faults. The learning approach has been utilized in [12] to handle communication among modules on a reconfigurable NoCs. In the C-Routing approach [13], the clustering model is studies to reduce the Q-table sizes. Bi-LCQ [14] applies the Q-Routing method on a cluster-based NoCs. All of the aforementioned works are presented in the realm of a 2D network.

2.3 Router micro-architecture

A generic router micro-architecture that accommodates Q-routing for deflection routing in irregular networks is shown in Figure 1. It is buffer-less and enacts a fully adaptive, non-minimal deflection routing algorithm, also known as hot-potato routing. A packet is only a single flit long and comprises control and payload bits. A hop is counted when a packet traverses the link from one router to the next. In the case where two or more packets compete for the same link, the router honors an oldest-first priority scheme. No packets are dropped from the network and when the network is congested, packets are accumulated in FIFO buffer in the network interface (NI) situated between each router and its local processing element. A router can be pipelined into stages. For a regular network, a relative addressing scheme is implemented which simplifies the duplication of identical routers when network structures of varying sizes are concerned. More details of the routing protocol and router micro-architecture are given in [15]. For irregular networks, look-up-table (LUT) based addressing scheme with Q-learning algorithm is implemented as described in Section 3.

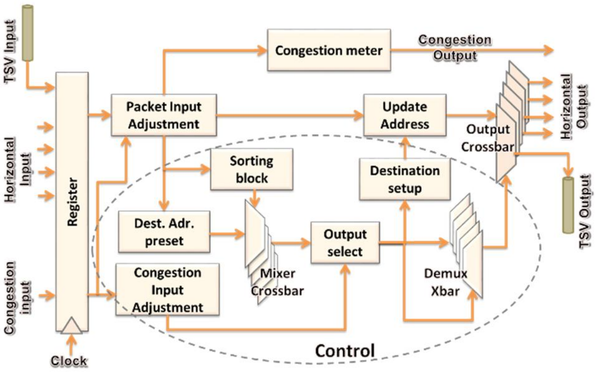

Figure 1. A multi-layer heterogeneous 3D many-core system

3. L-LEARNING: OPTIMIZATION FOR LATENCY UTILIZING Q-LEARNING

Deflection routing provides a full adaptively in routing packets and all minimal and non-minimal routes can be selected by packets [16]. This opportunity provides an excellent platform for applying Q-Learning methods where flexibility is a vital factor in finding the best solutions. In other words, Q-Learning may not be efficiently applied when there are few routing options provided by the routing algorithms as it lacks a proper exploring space.

3.1 Q-Table

In the Q-Routing approach, each router maintains a Q-Table (Table 1). In the Q-Table each row corresponds to one destination (i.e. from 0 to n) and each column indicates one output channel (i.e. S, W, E, D, R, U, and N). In a fully-connected 3D network, the size of a Q-Table is $n \times m \times k$ where n is the number of routers in the network, m is the number of output channels per router, and k is the size of each entry. As shown in Table 1, there are several crossed entries which are related to the unavailable links in the borderline routers. As in heterogeneous 3D NoCs, the network is partially connected, the Q-Table size is also reduced accordingly. Q-Table can be filled with different types of values, defined by the cost function. In this paper, we investigate the cost function as latency (L-Learning). However, the algorithm is general and can be applied to other cost functions such as temperature and IR-drop. Q-Tables entries are initially empty and can be filled at run time based on the minimum number of cycles to reach from a source to a destination router through each of the output channels.

Table 1. Q-Table for a fully connected 3D NoC

Output Dest.	S	W	E	D	R	U	N
0	▨	▨		▨			
1	▨			▨			
...							
n			▨			▨	▨

3.2 Cost Function

We consider the cost function as latency. Each entry of the table shows the estimated number of cycles to reach from a router to the destination router through one of the output channels. Since, a 3D NoC platform consists of routers, horizontal and vertical links, the cost function should be defined for each of them separately.

1) N1 cycles are taken to transfer a packet between two routers through a link. As the horizontal links are built differently than TSVs, the latency is also modelled differently. In our simulator, one cycle is considered to transfer a packet over a vertical link (i.e. $N1_V$) while depending on the length of a horizontal link, one or several cycles might be taken by the packet (i.e. $N1_H$).
2) N2 cycles are taken to process a packet at a router from an input port to an output port. Since packets are not stored at routers (routers are buffer-less) this value is constant for all routers. In our simulator this value is 4 cycles. Note that a router can be designed in 1 cycle but for a better throughput and a simpler logic we have chosen a 4-cycle router design.

The overall number of cycles from the router A to the destination router D includes: 1- the number of cycles taken in a router A (N2); 2- the number of cycles to transfer the packet from the horizontal ($N1_H$) or vertical link ($N1_V$) to the neighboring router B; 3- the number of cycles from the neighboring router B to the destination router D (N3). N2, $N1_H$, and $N1_V$ are static values and known by the router A at design time. However, N3 is unknown by the router and filled and updated gradually as packets are propagated inside the network.

3.3 Q-Routing Algorithm

The overall number of cycles from the router A to the destination router D can be estimated by:

$$C_{estimated} = (N1_H \text{ or } N1_V) + N2 + N3 \qquad (1)$$

Whenever a packet is sent from the router A to router B, a new estimation on the overall number of cycles from the router A to the destination D is obtained. This value is calculated at the router B and sent back to the router A. At the router A, the corresponding entry in the Q-Table will be updated. The current entry of the router A refers to the one associated with the destination D as the row and the output channel connecting the router A to the router B as the column. Thereby, the new Q-value is calculated by:

$$C_{New} = C_{Old} + \alpha(C_{estimated} - C_{old}) \qquad (2)$$

Learning is performed by updating Q-values. The learning rate (α) determines the rate in which the new information overwrites the old one. Learning rate can take a value between zero and one. Based on empirical evaluations, we have observed that the learning rate of 0.5 leads to the best performance so that we use it as a default value. Figure 2 shows one of the analysis where the learning rate of 0.5 is compared with the learning rate of 0.1 (Q-Table are dominated by old data) and 0.9 (based on the most recent data).

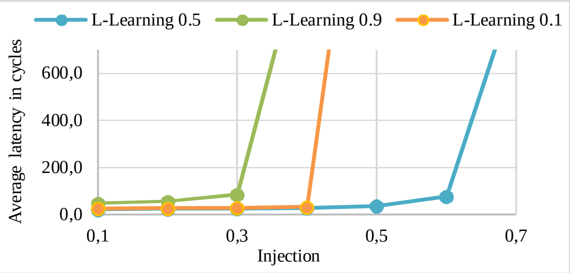

Figure 2. Analysis on the learning rate.

Figure 3 shows a 3D heterogeneous many-core system (bottom and mid layer) connected to a memory layer on top. Figure 4 shows the floorplan of the mid and bottom layers. The many-core system communicates with the memory using a WideIO TSV connecting the router 23 (mid layer) to the router 29 (top layer). The bottom and mid layer are connected to each other using three TSV between the following routers: routers 2 and 20, routers 8 and 23, and routers 11 and 27. As already mentioned, one cycle is taken to transfer the packet over the vertical link while the number of cycles over the horizontal links depends on the link length as shown in Figure 4. Now, let us explain the idea of the proposed method where a packet is sent from the source router 3 to the memory connected to the router 29 (e.g. a write message). Different steps are as follows: According to deflection routing, in the source router, similar to other routers, a packet can be delivered through any possible output channels. At the router 3, such available channels are W and N as shown in Table 2(a). The content of the row 29 shows the estimated number of cycles which takes for a packet to reach from the west or north output port to the destination 29. According to this table, approximately 41 and 53 cycles will be taken by the packet if delivered through W or N, respectively. Since the west port leads to the lower number of cycles, the packet is sent to the router 2.

At the router 2 (Table 2(b)), there are four possible output channels (i.e. W, E, U, and N) to forward the packet to the next router where the output channel U leads to the minimum number of cycles. Upon selecting the output port, the estimated number of cycles from this router to the destination router is extracted from the table and sent back to the router 3. This value is 20 cycles, referring to the row 29 and column U. By receiving this value at the router 3, the overall number of cycles can be calculated using Equation (1). The new estimation includes the number of cycles over the link from the router 3 to the router 2 (i.e. 4 cycles), the number of cycles at the router 2 to proceed the packet (i.e. 4 cycles), and the minimum estimated number of cycles from the router 2 to the destination router 29 (i.e. 20 cycles). Thereby, the new estimated value at the router 3 will be 28 cycles. With this new value, the corresponding entry of the Q-Table at the router 3 will be updated. This is done by taking the average of the current stored value (i.e. 41 cycles) and the new estimated value (i.e. 28 cycles) using the learning rate of 0.5. At the router 20 (Table 2(c)), the packet is sent to the north output port, showing the minimum number of cycles among the other options. Upon connecting the input channel to the output channel N, the estimated number of cycles (i.e. 14 cycles) is returned back to the router 2. Based on the estimated number of cycles from the router 20 to the destination router (i.e. 14 cycles), the processing time at the router 20 (i.e. 4 cycles), and the traversal time over TSV connecting the router 2 to the router 20 (i.e. 1 cycle), the overall number of cycles is calculated (i.e. 19 cycles). The corresponding entry at the router 2 is updated by taking the average of the new estimated value (i.e. 19 cycles) and an existing estimation (i.e. 20 cycles). Similarly, at the router 23 (Table 2(d)), the packet is sent to the output port U and the related entry at the router 20 is updated accordingly and so on until the packet reaches the destination.

4. EXPERIMENTAL RESULTS

Analyses are mainly performed between L-Learning and the standard model in both homogenous and heterogeneous 3D NoC. The basic framework of the router microarchitecture is the same for all implementations applying deflection routing in the buffer-less network. The main differences between the learning-based methods and the standard method can be described as follows:

Table 2. Q-Table at the router 3, 2, 20, and 23

(a) Q-Table at router 3

Output Dest.	W	N
0		
⋮		
29	41	53

(b) Q-Table at router 2

Output Dest.	W	E	U	N
0				
⋮				
29	51	37	20	21

(c) Q-Table at router 20

Output Dest.	W	E	D	N
0				
⋮				
29	40	30	24	14

(d) Q-Table at router 23

Output Dest.	S	E	U	N
0				
⋮				
29	26	12	6	20

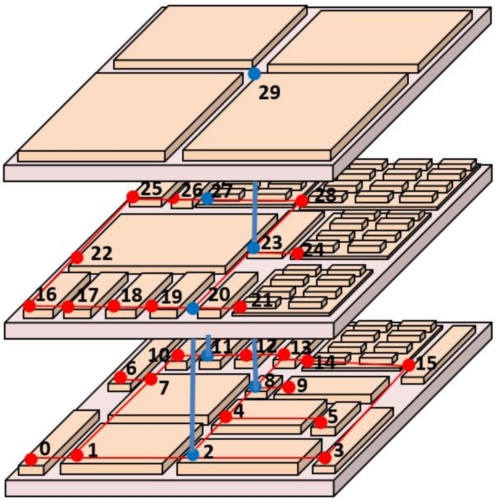

Figure 3. A multi-layer heterogeneous 3D many-core system.

Standard model (baseline): In the standard model, the look-up-table (LUT) is filled at design time with an expert knowledge on the underlying topology. The table determines the priority between output channels at a router for each specific source and destination router. Thereby, the output channel which leads to the shortest path receives the highest priority and so on. Any changes in the topology do not affect the content of LUT at run time.

L-Learning: Instead of using LUT tables in the standard model (i.e. determines priority between output channels based on the shortest paths), learning-based models take advantage of Q-Tables (i.e. determines the latency cost from each output port to the destination). Q-Tables are initially empty and are initialized during the setup phase. Each router sends a packet to its router, then to the next immediate router and so on until all routers are covered. Large networks require longer setup time. For a network *N* = *X*×*Y*×*Z*, it requires *N*×*N* iteration. Thereby, the setup operation is completed within *N^2* cycles. After initializing the Q-Tables, the L-Learning/P-Learning algorithm is applied for making the routing decision and the selection between output channels. Q-Tables are dynamically updated at run time and adjusted themselves to the underlying topology and traffic patterns. The latency is measured in cycles from the packet injection at the source to its ejection at the destination.

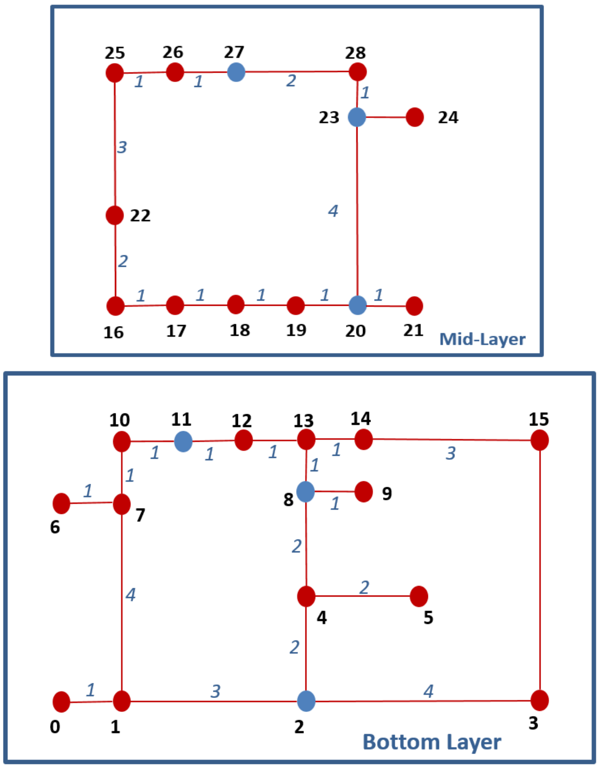

Figure 4. A floorplan for the heterogeneous many-core system showing an irregular topology.

4.1 Latency Analysis in Regular 3D NoC

The latency is analyzed in regular networks under different traffic patterns. This allows making simple comparison of L-Learning with the standard model. Three traffic patterns are considered: Uniform Random Traffic (URT) and Hotspot traffic profiles.

The latency is measured in cycles from the packet injection at the source to its ejection at the destination. Figure 5 shows the results for L-Learning in a regular 4×4×4 networks under URT - uniform random traffic pattern where each router sends packets randomly to all destinations in the network. As can be seen in this figure, L-Learning leads to a better latency in low injection rates while the standard model performs better when the network pass the saturation point (0.4). The better performance of L-Learning in low injection rates is obtained under the condition that L-Learning has started its operation without a prior knowledge on the network topology, i.e. regular in this case. On the other hand, in the standard model, LUTs are filled to guarantee the use of shortest possible paths as long as possible. From this example, it can be observed that shortest paths may not necessarily lead to the best performance.

Figure 6 shows the results for L-Learning in a regular 4×4×4 network under hotspot traffic pattern. Two hotspot routers are selected as router 59 in the top layer and router 21 in the second layer. All the other routers make frequent communication with the hotspot routers each with 40% of their traffic directed to the hotspot routers. Such configuration allows the evaluation of regular networks under a more realistic traffic configuration. Similar to the uniform traffic, L-Learning leads to a lower latency in low injection rates and it works as efficient as the standard model in high injection rates. In other words, L-Learning reaches to its optimal performance by performing Q-Routing algorithm and updating the Q-Tables at run time.

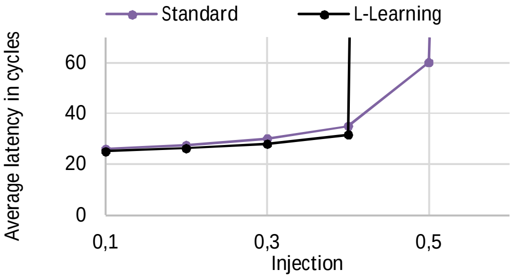

Figure 5. Average latency in 4×4×4 network under uniform random traffic.

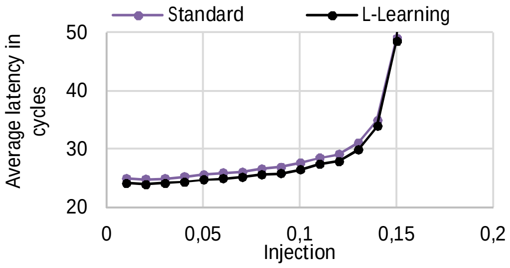

Figure 6. Average latency in 4×4×4 network under hotspot traffic.

4.2 Latency Analysis and Throughput in Irregular 3D NoC

Another set of analyses is performed in an irregular 3D NoC of Figure 3. The irregular network is selected to represent the irregularity in the number of TSVs, routers placement, and the link length. The router to router link length, for both horizontal and vertical TSV, is pre-determined as the link length is physically fixed. The underlying traffic pattern is URT where each core sends a packet to every other core randomly. Figure 7 and Figure 8 depict the latency and throughput results of the irregular network, respectively, where the baseline implementation uses a pre-filled LUT to make routing decisions. The performance of L-Learning is nearly the same as the standard model up to the injection rate 0.5. In higher injection rates, the standard model works better. In other words, L-Learning reaches near optimal performance starting with empty Q-Tables whereas the LUT in the standard model has been filled based on the knowledge on underlying connectivity of the irregular network.

4.3 Power Analysis in Regular 3D NoC

The power consumption at each router is measured in terms of milliwatts (mW). Figure 9 shows the power distribution at each router in a 4×4×4 regular network under the uniform random traffic profile. In the standard model, higher power consumption is observed at router 21, 22, 25, 26, 36, 37, 41, and 42.

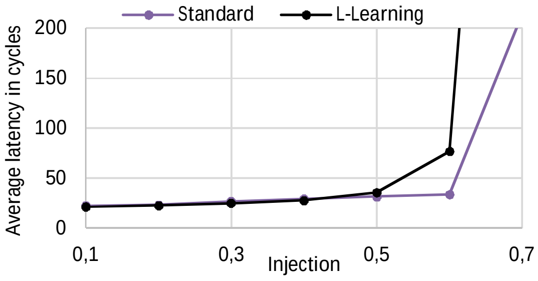

Figure 7. Average latency in the irregular network.

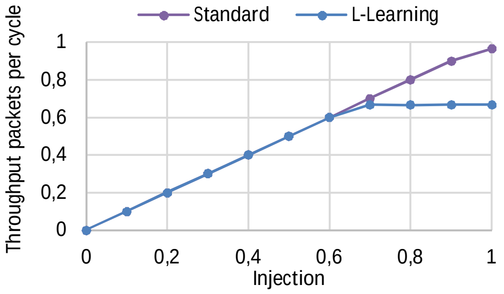

Figure 8. Throughput in the irregular network.

These routers are physically located at the center of the 3D networks. L-Learning, however, consumes lower power in these routers. This is due to the fact that the Q-Learning mechanism in L-Learning allows the routing algorithm to balance the traffic load by redirecting the traffic to relatively less congested areas of the network. This power distribution balance is reflected in the borderline and corner routers of the network.

Similar behaviors are also observed from simulation results in Figure 10 under the hotspot traffic profile. The inability of the standard model to redirect traffic in congested areas leads to a larger power consumption in the corresponding routers. In the hotspot traffic, the hotspot routers 21 and 59 along with the routers at the center of the network have a larger power consumption per router compared to L-Learning in regular networks. Under all the examined synthetic traffic patterns, the maximum power has been significantly decreased by applying L-Learning.

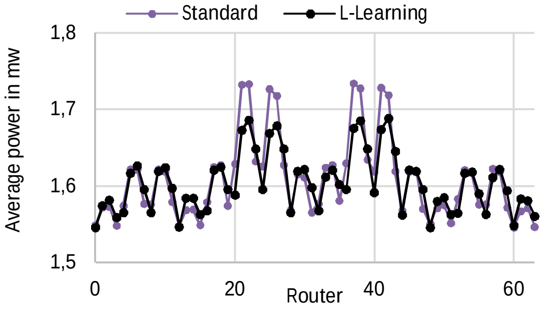

Figure 9. Average power consumption at each router in 4×4×4 regular network under uniform random traffic.

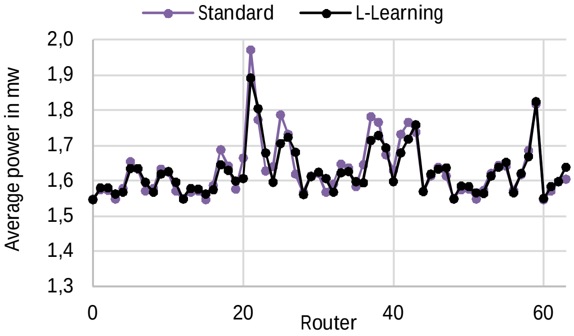

Figure 10. Average power consumption at each router in 4×4×4 regular network under hotspot traffic.

4.4 Power Analysis in Irregular 3D NoC

Figure 11 shows the power consumption at each router of the irregular 3D NoCs of Figure 1. As can be seen in this figure, the most congested routers are 2, 8, 20, and 23. This is a correct observation as all these routers are connected to TSVs and thus handling more traffic. When applying the standard model and L-Learning, we observe that L-Learning reduces the pick power consumption in all congested routers by moving the flow to the less-congested parts, and thus achieving a better power balance in the network. This improvement shows that L-Learning has been very successful in adjusting itself to an unknown irregular topology and even outperforms the standard model which has been designed by an expert knowledge. Comparing the results of power and latency in the same irregular 3D NoCs (Figure 7 and Figure 11), it can be seen that although L-Learning is saturated earlier than the standard model (i.e. because of using longer paths to distribute packets), but on the other hand it has a better control on reducing the maximum power consumption which is a critical factor in 3D designs.

4.5 Reliability

Taking advantages of deflection routing, both learning-based and the standard model are resilient against faults in routers, horizontal and vertical links. A router is disconnected from the network only when it loses all its connections to the neighboring routers. Faulty horizontal and vertical links in a router can be simply tolerated by the ability of sending packets through any other available link. More importantly, L-Learning can dynamically adjust itself with the new configuration, caused by disabled faulty links and faulty routers. Shortly after reconfiguration the network reaches its near optimal performance and power consumption.

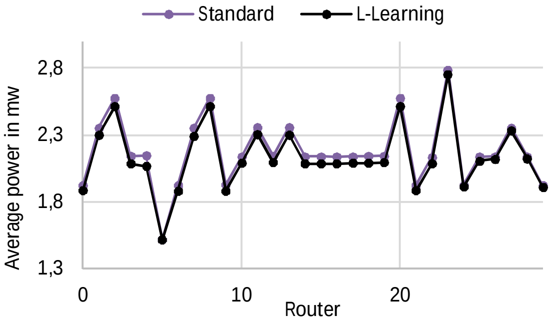

Figure 11. Average power consumption at each router in irregular network.

5. CONCLUSION

In the current state of the art, heterogeneous 3D NoCs lacks a proper solution to offer a low power, low area, and high-performance design while being able to tolerate faults in routers, links and TSVs. Usually these issues are investigated independently so that a factor can be optimized but other factors are scarified accordingly e.g. a fault-tolerant method can tolerate a fault but performance and power will be significantly affected. In this paper, we suggested a solution, called L-Learning, considering all these factors together. L-Learning combines the best features of Q-Learning method and deflection routing in a buffer-less network. The applied Q-Routing algorithm can dynamically balance the traffic over the network at run time without a prior knowledge on the underlying topology and traffic pattern. This implies that a network can reach to its optimal performance and power consumption shortly after the reconfiguration. Such run-time efficiency is of significant importance when considering faults and heterogeneity in 3D NoCs. Deflection routing has been employed, offering an adaptive type of routing, characterized by the ability to route packets to any possible output port in a router. Deflection routing provides a superior platform to apply the Q-Learning method and offers an inherent resilient against faults. Deflection routing provides a superior platform to apply the Q-Learning method and offers an inherent resilient against faults. No limitations such as turn models are applied and there is no need to buffer packets.

Acknowledgments

This work was supported by Ella and Georg Ehrnrooth Foundation, Finland.

REFERENCES

[1] Denis Dutoit et al, “A 0.9 pJ/bit, 12.8 GByte/s WideIO Memory Interface in a 3D-IC NoC-based MPSoC,” in proc. of VLSI Technology (VLSIT) - Circuits Digest of Technical Papers, pp. C22 - C23, 2013.

[2] M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila, and H. Tenhunen, “LEAR - A Low-weight and Highly Adaptive Routing Method for Distributing Congestions in On-Chip Networks”, in Proc. of PDP, pp. 520-524, 2012.

[3] M. Palesi and M. Daneshtalab (Eds.), “Routing Algorithms in Networks-on-Chip,” Springer 2014.

[4] M Ebrahimi, M Daneshtalab, P Liljeberg, J Plosila, J Flich, H Tenhunen, “Path-based partitioning methods for 3D networks-on-chip with minimal adaptive routing”, IEEE Transactions on Computers, 63 (3), 718-733, 2014.

[5] M. Ebrahimi, M. Daneshtalab, F. Farahnakian, P. Liljeberg, J. Plosila, M. Palesi, and H. Tenhunen, “HARAQ: Congestion-Aware Learning Model for Highly Adaptive Routing Algorithm in On-Chip Networks”, in Proc. of NOCS, pp. 19-26, 2012.

[6] T. Moscibroda and O. Mutlu, “A case for bufferless routing in on-chip networks,” in Proc. of ISCA 2009.

[7] P. Baran. On distributed communications networks. IEEE Transactions on Communications, pages 1–9, 1964.

[8] E. Bolotin et al., “Routing table minimization for irregular mesh nocs,” in DATE, pp. 942–947, 2007.

[9] J. Boyan et al., “Packet routing in dynamically changing networks: A reinforcement learning approach,” in Advances in Neural Information, pp. 671–678, 1994.

[10] S. Kumar et al., “Dual reinforcement Q-Routing: An on-line adaptive routing algorithm,” in the Artificial Neural Networks in Engineering Conference, pp. 231–238, 1997.

[11] C. Feng et al., “A reconfigurable fault-tolerant deflection routing algorithm based on reinforcement learning for network-on-chip,” in NoCArc, pp. 11–16, 2010.

[12] M. Majer et al., “Packet routing in dynamically changing networks on chip,” in Proc. IPDPS, pp.154-162, 2005.

[13] M. Puthal et al., “C-routing: An adaptive hierarchical noc routing methodology,” in VLSI-SoC, pp. 392–397, 2011.

[14] F. Farahnakian et al., “Bi-lcq: A low-weight clustering-based Q-Learning approach for nocs,” in MICPRO, pp. 64–75, 2013.

[15] A.Y. Weldezion, M. Grange, D. Pamunuwa, A. Jantsch, and H. Tenhunen. “A scalable multi-dimensional NoC simulation model for diverse spatio-temporal traffic patterns”. In IEEE Proc. 3DIC, pp. 1–5, 2013.

[16] F Farahnakian, M Ebrahimi, M Daneshtalab, J Plosila, P Liljeberg, “Adaptive reinforcement learning method for networks-on-chip”, in Proc. of SAMOS, pp. 236-243, 2012.