Communication Architecture Optimization: Making the Shortest Path Shorter in Regular Networks-on-Chip

Umit Y. Ogras, Radu Marculescu
Department of Electrical and Computer Engineering
Carnegie Mellon University, USA
{uogras,radum}@ece.cmu.edu

Hyung Gyu Lee, Naehyuck Chang
School of Computer Science and Engineering
Seoul National University, Korea
{hglee,naehyuck}@cselab.snu.ac.kr

Abstract

Network-on-Chip (NoC)-based communication represents a promising solution to complex on-chip communication problems. Due to their regular structure, mesh-like NoC architectures have become very popular recently. However, they have poor topological properties such as long inter-node distances. In this paper, we address this very issue and explore the potential of partial NoC customization to improve both static and dynamic properties of the network significantly, while minimally affecting its regularity. Precise energy measurements on an FPGA prototype show that the improvement in network properties is achieved without a significant penalty in area and communication energy consumption.

1. Introduction

Adding more functionality to future SoCs depends critically on finding truly scalable on-chip communication architectures. Inspired by the success of large scale interconnection networks, such as internet and parallel computer networks, the Network-on-Chip (NoC) paradigm has been recently put forth as a potential solution to on-chip communication problems.

Interconnect networks are conveniently abstracted as graphs whose topological properties have a major effect on their communication capabilities. Until recently, only fixed (deterministic) or completely random graphs were thoroughly studied [13,16]. However, research on many real networks such as WWW, internet, US power grid, collaboration networks, network of movie actors, *etc.* revealed that many real networks are neither completely regular, nor completely random [14]. Instead, they cover a sizeable design space in-between these two extremes. These networks, commonly referred to as *small-world* networks, are characterized by small inter-node distances and large clustering coefficients. Small-world phenomenon (popularly known as *six degrees of separation*) implies that all nodes are connected to each other by a short chain of intermediate nodes; this is a highly desirable property for efficient information dissemination.

Similar to network research, initial studies in the NoC area have considered either completely structured (grid-like) [6,7] or fully customized networks [8,9,10]. Amazingly enough, the mesh network lacks both high clustering (see Section 4.3) and short paths between remotely situated nodes (see Section 5.1). For example, the diameter of a 2D mesh network increases linearly with the network dimensions. Large inter-node distances increase the messages latency and link blocking probability. On the other hand, fully customized topologies, resembling random graphs, achieve short inter-node distances at the expense of a complicated communication structure. Consequently, issues related to implementation complexity, lack of standardization, *etc.* make fully customized topologies less attractive in practice.

Fortunately, it is possible to find a sweet spot between these two extremes in the design space and exploit the best of both worlds. Theoretical studies have shown that the regular networks can be transformed into *small-world networks* by introducing a few random long-range links [14,15]. Similarly, long-range links can be used to improve the capabilities of the grid-like networks, while minimally affecting their regularity. In [11], we have proposed an algorithm for inserting application-specific long-range links which optimize the network performance under constraints on the number of allowable additional links. It has been shown that inserting long-range links successfully maximizes the critical traffic load at which the network becomes congested.

In this paper, we show that besides boosting the performance, the newly added long-range links have a significant impact on the static and other dynamic properties of the network. Consequently, the resulting NoC architecture enjoys the advantages offered by both regularity and partial customization. To demonstrate the benefits of long-range links, we implement an FPGA prototype and compare the initial mesh network against a mesh network with long-range links inserted to it. Accurate measurements show that significant improvements in network properties are possible with similar energy consumption and only a small increase in the area.

The remaining part of this paper is organized as follows. Section 2 reviews the related work. The long-range link insertion algorithm is presented in Section 3, while the impact of the long-range links on the static and dynamic properties of the network is discussed in sections 4 and 5, respectively. Section 6 presents a detailed experimental analysis based on an FPGA prototype. Finally, our conclusions appear in Section 7.

2. Related work

The NoC concept was introduced in [1,2] and several design flows for application-specific NoCs were presented in [3-5]. Automated techniques for NoC design have been discussed in [6-10]. More precisely, application mapping to standard network topologies is discussed in [6,7], while studies in [8-10] address the issue of designing application-specific NoC architectures.

The effect of inserting random links to 2-D mesh networks is first analyzed in [17]. In that work, the packets consist of a single atomic entity containing address information only. Also, due to the infinite buffer assumption, the authors do not deal with deadlock states explicitly. In [14,15], the authors show that random insertion of long-range links to a regular network can transform it into a small-world network.

Different from all these studies, we consider herein *application-specific* traffic patterns and aim precisely at optimizing the network performance in a *deterministic* rather than random fashion. Moreover, the algorithm for smart link insertion proposed in [11] removes the restrictions in [17] by considering that the network routers have *bounded* buffers and work for wormhole routing with *arbitrary* packet sizes. In this paper, we extend the work in [11] and use a novel set of metrics with a measurable impact on the static and dynamic properties of the network. We also use an FPGA-based prototype to perform direct performance and energy measurements under various traffic conditions.

3. Structured network with long-range links

We consider a $m \times n$ 2D mesh network *(N)* whose tiles are occupied with processing and/or storage elements. The communication volume between tile i and tile j (i, $j \in N$) is denoted by V_{ij}, while the corresponding bandwidth requirement is denoted by W_{ij}. We assume wormhole switching, but the results derived here are also applicable to packet- and virtual cut-through switching and for other network topologies.

To preserve regularity, the long-range links are segmented into regular links connected by repeaters, as in Figure 1. The use of repeaters with buffering capabilities guarantees latency-insensitive operation [18]. The repeaters can be regarded as simplified routers consisting of only two ports which can accept an incoming flit, store it into a FIFO buffer, and finally forward it to the output port [11]. The area overhead caused by the long links is expressed as a multiple of the standard link segments that make up the long-range link.

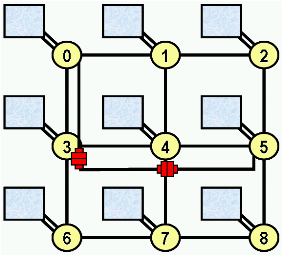

Figure 1. Illustration of long-range link insertion.

The main goal of long-range link insertion algorithm is to minimize the average packet delay subject to the bandwidth requirements and available resources. More formally:

Given the communication volumes $V_{ij}\ \forall i, j \in N$, the corresponding bandwidth constraints W_{ij}, the routing strategy for the mesh network, and the amount of available resources S

Determine the set of long-range links $L_S = \{LL_{ij} \mid i, j \in N\}$ to be inserted in the network, and a deadlock-free routing strategy for the newly added long-range links

such that the long-range links satisfy the bandwidth requirements W_{ij} and the *critical load* at which the network enters the congested phase, λ_c, is maximized, *i.e.*

$$max(\lambda_c) \quad subject\ to \quad \sum_{l \in L_S} s(l) < S \qquad \textbf{(1)}$$

As detailed in [11], the algorithm first computes the *critical load* (λ_c) for the initial mesh network. After that, it computes the improvement in the average inter-node distance due to the insertion of each feasible long-range link. Once the most beneficial link is determined, this link is inserted permanently in the network and the amount of utilized resources is updated. The algorithm continues inserting links until all available resources are used up. At the end, a routing strategy governing the use of long-range links is produced.

4. Impact of long-range links on *static* properties of the network

We first discuss four static properties and their relevance to NoC design. Next, we analyze the impact of long-range links on these properties. For comparison, we insert long-range links to mesh networks of sizes ranging from 4×4 to 12×12 under *hotspot* traffic[1] and analyze their behavior.

4.1. Degree distribution

The degree of a node, d_i, $i \in N$, is given by the number of links incident to that node [16]. We are particularly interested in three flavors of the degree distribution:

- Maximum degree $d_{max} = \underset{i}{max}\{d_i\}$
- The range of different degrees $d_{span} = d_{max} - d_{min}$
- The sum of all degrees $d_T = \sum_i d_i$.

The maximum degree determines the largest router needed in the network. For instance, d_{max}=4 implies that the router has at most 4 neighbors so it has to have a 5×5 switch, plus a connection to the local node. The number of different degrees is a measure of heterogeneity of the network. Finally, the sum of all degrees gives the total number of uni-directional links in the network, since the neighboring routers are connected by two uni-directional links.

Impact of long-range links on the *degree distribution*

The maximum degree of a standard mesh network is 4. This value increases to 5 for all networks obtained using our proposed technique, because we limit to 1 the number extra

[1] In hotspot traffic, a few nodes in the network receive more traffic compared to the remaining nodes.

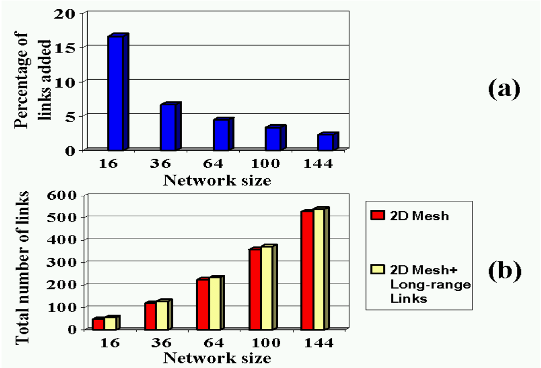

Figure 2. (a) Percentage of the additional long-range links (b) The change in the total number of links.

links attached to each tile. Likewise, the degree of all other nodes is either incremented by 1 or remains the same. Given that the number of modified routers is limited by the resource constraint S, the insertion of long-range links minimally alters the structure of the network.

The total number of links for various network sizes and their variation with the addition of long-range links is shown in Figure 2. We note that, the total number of links increases slightly. Furthermore, as the network size scales up, the percentage of the additional links drops dramatically. For the networks considered in the rest of the paper, the number of additional links is the same as shown in Figure 2.

4.2. Diameter and cost factor

By definition, the *diameter* (D) is the longest path between any two nodes in the network. We note that:

- The variation of D, as the network size increases, provides a measure of the scalability of the network
- Merely comparing the diameters of two networks does not provide a fair comparison: higher node degrees will result in smaller diameter values but this may incur an extra cost. Hence, we use the product between D and the mean degree, $d_m = mean(d_i)$, called *cost factor* [13] as a quality measure.

Impact of long-range links on the *diameter* and *cost factor*

The change in the network diameter and cost factor with the insertion of long-range links is shown in Figure 3 for various network sizes. We observe about 30% reduction in the diameter for the networks under study. A similar reduction is obtained for the cost factor. Moreover, the gain in the cost factor increases as the network scales up (e.g. 22% for 4×4 and 30% for the 12×12 network), since the change in the average node degree becomes smaller.

4.3. Clustering Coefficient

In many applications, a set of nodes tends to communicate more frequently with each other compared to the remaining nodes. For this reason, a high clustering in the topology that matches the application characteristics is highly desirable.

The degree of clustering (*i.e.* how tightly the nodes are interconnected) can be measured by the clustering coefficient [14]. To compute the clustering coefficient of node i (C_i), we focus on its neighbors. If the node has n_i neighbors, then there

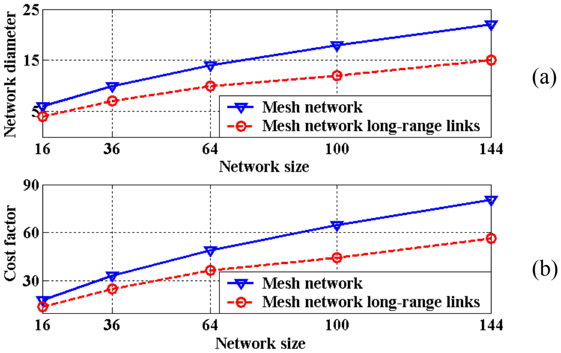

Figure 3. The variation of (a) the network diameter and (b) cost factor with the insertion of long-range links.

can be at most $n_i(n_i$ -1)/2 links between these neighbors. If we denote the number of actual links that exist between all the neighbors by l_i, then C_i is expressed as,

$$C_i = \frac{2l_i}{n_i(n_i-1)} \tag{2}$$

The clustering coefficient of the network (C_N) is then found by averaging the clustering coefficients over all nodes, *i.e.* $C_N = mean(C_i)$. Hence, a large C_N implies that the nodes situated close to each other are highly connected.

Impact of long-range links on the *clustering coefficient*

The clustering coefficient of all nodes in a mesh is zero, because none of the immediate neighbors of a given node are directly connected to each other. For example, the central node in Figure 4 is connected to 4 immediate neighbors via the solid lines. There are 6 possible connections (shown as dashed lines) between these 4 neighbors. Since none of these connections do actually exist in a 2D mesh network, the clustering coefficient is zero. Similar arguments hold for the remaining nodes.

Inserting long-range links of length 2 increases the clustering coefficient of the network. For example, adding any of the diagonals (dashed lines) in Figure 4 increases the clustering coefficient of the network from 0 to 0.2. This large value grows even larger by adding only one extra-link.

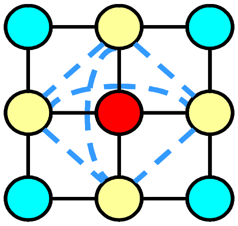

Figure 4. Illustration of computing the clustering coefficient.

5. The impact of long-range links on the *dynamic* properties of the network

The dynamic properties are determined by the traffic flow in the network, the routing strategy, and the network topology. We analyze next two dynamic properties and their variation with the insertion of long-range links. Besides the *hotspot* benchmark, we use the *transpose* traffic, where each node sends packets to the nodes located symmetrically to the first diagonal. We also use two applications with realistic traffic: the 4×4 *auto industry* benchmark and 5×5 *telecom* benchmark retrieved from the E3S benchmark suite [19].

5.1. Average inter-node distance

The average inter-node distance (μ) is an important performance metric not only because it determines the packet delay in absence of contention, but also due to its effect on traffic congestion. From Little's theorem, the number of packets in the network is proportional to the average time the packets spend in the network. Hence, for a given injection rate, a larger μ results in a larger number of packets in the network, so this increases congestion. Moreover, as shown in [17], μ is inversely proportional to the critical traffic load, λ_c, which indicates the transition from a free to a congested state.

Impact of long-range links on the *avg. inter-node distance*

The variation of the μ for the traffic patterns under study is summarized in the upper half of Table 1. We observe a consistent reduction for μ values which is a typical indication of the small-world effect. For example, the average inter-node distance for the 5×5 *telecom* benchmark drops from 3.67 *hops* to 2.21 *hops* after inserting long-range links. Similarly, we observe a 29.6% reduction for the *auto-industry* benchmark and a 39.9% reduction for the *transpose* traffic.

5.2. Link utilization

Ideally, the utilization (ρ) for all the links should be uniform across the network, such that the traffic load is uniformly distributed and none of the routers becomes a bottleneck. The maximum link utilization can be defined as

$$\rho_{max} = \max_{i}\{\rho_i\} \quad i \in N \tag{3}$$

Since the link with the maximum utilization becomes the performance bottleneck, smaller values of maximum node utilization imply higher critical traffic load for the network. Hence, this metric decreases after long-range link insertion.

Impact of long-range links on the *link utilization*

The impact of long-range links on the link utilization for the *hotspot*, *transpose*, *auto industry* and *telecom* traffic patterns, mentioned in the beginning of Section 5, is summarized in the lower half of Table 1. Similar to the average inter-node distance, we observe a reduction as large as 41% (8×8 *Hotspot*), although the proposed algorithm does not directly minimize the maximum link utilization. We also plot the link utilizations for all network links before and after inserting long-range links (Figure 5). In this figure, the *x-axis* is reserved for the routers with IDs 1 to 36 in a 6x6 network. The *y-axis* shows the corresponding ports of the routers. *L* stands for the local ports; *N, S, W, E* stand for the directions North, South, West, East; and *LR* stands for the long-range link. Since Figure 5a shows the utilizations before the insertion of long-range links, the link utilizations for the ports with long-range link denoted by *LR* are all 0. For this case, there is a hotspot region around routers 9, 10 and 11, reaching a peak of 0.072 *packets/cycle* at the *S* port of router 11. The utilization values after the insertion of long-range links are shown in Figure 5b. We observe that, the insertion of long-range links mitigates the hotspot around that region. The maximum link utilization across the network drops to 0.048 *packets/cycle*, which is about 33.3% reduction.

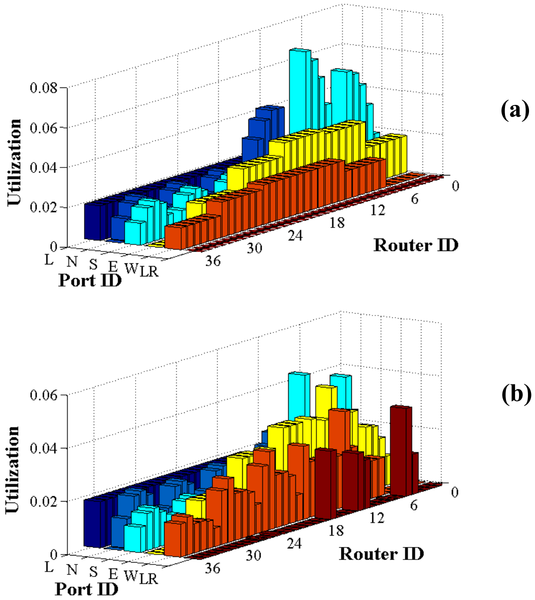

Figure 5. The change in the link utilizations for (a) 2D mesh and (b) after the insertion of long-range links.

6. Experiments using an FPGA prototype

In this section, we demonstrate the impact of the long-range links on the capabilities of the network. The experimental results are obtained using a Xilinx Virtex-2 based FPGA prototype which implements a 4×4 mesh network and a few additional long-range links, which have been analyzed in previous sections. Besides providing performance and area comparisons, we also evaluate the impact of the long-range links on energy consumption.

6.1. Details of the FPGA prototype

We first implement a 4×4 network with 16 routers and processing elements (PE). The PEs can generate *hotspot*, *transpose* and *auto-industry* traffic patterns as discussed before. The routers consist of I/O ports, decision logic, switch fabric and output controllers. The number of I/O ports varies from 3 to 6 depending on the location of the router in the network and whether or not a long-range link is attached to it. Finally, all the I/O ports of the routers are 16-bit wide.

The network employs wormhole routing. It takes 4 cycles to route the header flit. Then, the remaining flits follow the header in a pipelined fashion. The packet size in terms of number of flits is also parameterized. In our experiments, we used packet sizes ranging from 5 to 32 flits. For flexibility reasons, the routing strategy is implemented as a lookup table and fed to the network before configuring the FPGA.

Table 1: The impact of long-range (LR) links on dynamic properties (average inter-node distance and link utilization).

		4x4 Hotspot	*6x6 Hotspot*	*8x8 Hotspot*	*10x10 Hotspot*	*4x4 transpose*	*4x4 auto-indust*	*5x5 telecom*
Avg. inter-node distance (μ) (# of *hops)*	**2D Mesh**	2.65	4.15	5.27	6.76	3.33	2.53	3.67
	2D Mesh with LR links	2.22	3.60	4.61	5.90	2.00	1.78	2.21
	Reduction	**16.2%**	**13.3%**	**12.5%**	**12.7%**	**39.9%**	**29.6%**	**39.8%**
Max link utilization (ρ_{max}) *(packets/ cycle)*	**2D Mesh**	0.18	0.12	0.075	0.054	0.25	0.26	0.17
	2D Mesh with LR links	0.11	0.077	0.044	0.035	0.17	0.20	0.15
	Reduction	**38.8%**	**35.8%**	**41.3%**	**20.4%**	**32.0%**	**23.0%**	**11.7%**

6.2. Detailed performance comparisons

The driver applications are first implemented on a 4×4 mesh network. Then, the application-specific long-range links are inserted using the link insertion algorithm, with a resource constraint of $s(l)=12$; this means that *at most* 12 short link segments are allowed to be inserted to the initial mesh network. The performance of the network before and after the insertion of long-range links is evaluated in terms of the average packet latency and maximum network throughput. The comparison of the average packet latency as a function of the packet injection rate is shown in Figure 6. We observe that the packet latency at low injection rates drops about 20% for the *auto-industry* application. As the packet injection rate increases, the reduction in latency goes over 50% at 0.35 *packets/cycle*; beyond that value the increase in the latency is basically exponential. Likewise, the achievable network throughput increases from 0.36 to 0.40 *packets/cycle,* resulting in about 11% improvement. A similar behavior for the *hotspot* traffic is shown in Figure 6, while the *transpose* traffic is discussed in Section 6.4.

6.3. Detailed area comparisons

Inserting long-range links requires more resources, hence, increases the area of the design. Therefore, we analyze the size of the individual routers in a 4×4 mesh network and its customized version with long-range links under the constraint of $s(l) = 12$ (Table 2).

Inserting long-range links increases the number of ports at both ends of the link. We observe that moving from 3-to-4, 4-to-5, and 5-to-6 ports increases the slice utilization by 85, 93 and 106 slices, respectively. Moreover, we observe that the total number of slices used by a 4×4 network implementing transpose traffic rises about 7.0%, as summarized in Table 2.

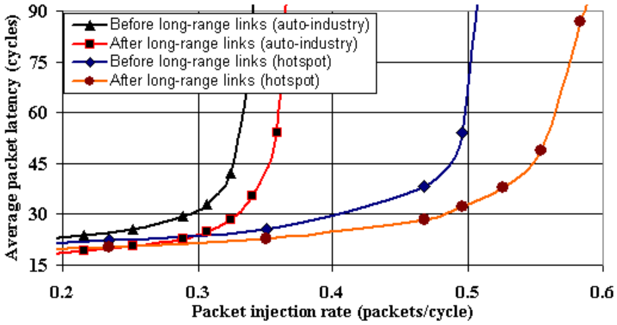

Figure 6. Comparison of the average packet latency for *hotspot* and *auto-industry* benchmarks *before* and *after* the insertion of long-range links.

Table 2: Impact of inserting of long-range links on area. The synthesis is performed for Xilinx Virtex2 4000 FPGA.

	Number of Slices	**Device Utilization**
3-port router	219	1.4%
4-port router	304	1.8%
5-port router	397	2.2%
6-port router	503	2.8%
4×4 ***Mesh network***	6683	29.0%
Mesh with long-range links	7152	31.0%

6.4. Energy Comparisons

The final issue to address is the impact of long-range links on the energy consumption. Since the long-range links consist of several regular links with repeaters between them, we expect a minimal change in the link and buffer energy consumption. At the same time, the small increase in energy dissipation due to the long-range links is compensated by routing data through repeaters. Indeed, sending a bit of data through a repeater consumes much less switching energy compared to using regular routers. Overall, we expect a small change in the total energy consumption.

To obtain accurate energy measurements and show the impact of the long-range links, we use the cycle accurate energy measurement tool for FPGAs presented in [20]. At each clock cycle, the tool charges a set of capacitors connected to the FPGA device through switches. During the next cycle, the FPGA is powered up by the previously charged capacitors. By measuring the initial and final voltages on the capacitors, the tool computes the exact energy consumed by the FPGA. Since the tool measures the dynamic and static energy consumption separately, we can derive the power consumption for a specific operating frequency with good accuracy. We also make sure that the FPGA prototype preserves the grid-like floorplan of the topology by properly using the Xilinx floorplanner.

To evaluate the energy consumption, we defined the job of each PE as sending 50 packets of 32 flits each, to certain destination nodes. Then, two sets of experiments are performed to assess the effect of the long-range links on the energy consumption.

In the first set of experiments, the PEs inject packets at uniform intervals of time until all 50 packets are sent. Since the time interval is set to be the same with or without long-range links, the total time to complete the job does not change considerably (7694 *cycles* for the regular mesh network compared to 7674 *cycles* after inserting the long-range links), as shown in Table 3. Our measurements show that there is about 2.2% reduction in the energy consumption due to the insertion of long-range links, which is in agreement with our expectations.

Table 3: Impact of long-range links on energy consumption.

Constant traffic rate (0.082 *packets/cycle*)	**Exec time** **(*cycles*)**	**Energy** **(μJ)**	**Power** **(*mW*)**
4×4 ***Mesh network***	7694	28.02	364.2
Mesh with long-range links	7674	27.39	356.9
Bursty traffic			
4×4 ***Mesh network***	5120	17.14	334.7
Mesh with long-range links	3416	11.72	343.2

In the second set of experiments, the packet injection is bursty, *i.e.* the PEs produce and send packets as long as the down-stream router can accept them. It takes 5120 *cycles* for the mesh network to complete the job, while the network with long-range links finishes the job in 3416 *cycles*. The corresponding energy consumption is found to be 17.14μJ without the long-range links and 11.72μJ after the insertion of long-range links, respectively. The decrease in the energy consumption is caused by the reduction in the execution time. Although, after inserting the long-range links, the power consumption increases from 334.7mW to 343.2mW (@100MHz clock frequency), the speed-up in completing the job can be exploited to maximize the duration of the possible power down mode and then save energy.

6.5. A final word on long-link customization

Finally, we note that customizing a 2-D mesh with long-range links is a more general approach than simply choosing a torus or other higher dimensional network. As a matter of explanation, since the higher dimensional mesh and torus networks need to be embedded into a 2D plan, their on-chip implementations looks similar to the implementation of a 2D mesh network with application-specific long-range links, except for a fundamental difference. The latter finds the optimal links to be inserted based on a rigorous analysis rather than by following a fixed rule. In fact, the application-specific customization with long-range links will generate standard higher dimensional networks, if we replace the link insertion algorithm with a static link insertion rule. As a result, the proposed technique can achieve a better performance compared to a higher dimensional network even if utilizes about the same or less resources. To verify this fact, we implemented a 4×4 2D torus network with folded links [2], and a mesh network with 8 uni-directional links generated by our proposed technique. Our simulations show that the average packet latency in our design, at 0.48 packet/cycle injection rate, is only 34.4 cycles compared to 77.0 cycles for the torus network. This significant gain is obtained by utilizing only half of resources; indeed, inserting the most beneficial links for a given traffic pattern makes more sense than blindly adding wrap-around channels all over the network as is the case for the folded torus.

7. Conclusion

In this paper, we have explored the potential of inserting application-specific long-range links to 2D mesh networks. As shown, inserting a small number of long-range links can significantly improve both static (*e.g.,* network diameter) and dynamic (*e.g.,* average inter-node distance, link utilization) properties of the network. Using accurate measurements on an FPGA prototype, we have also demonstrated that the improvements obtained incur only a small area overhead and have a minimal effect on the energy consumption.

Acknowledgements: This research is supported by Marco GSRC and SRC 2004-HJ-1189.

8. References

[1] A. Hemani, *et. al.* "*Network on Chip: An architecture for billion transistor era,*" Proc. IEEE NorChip Conference, Nov. 2000.

[2] W. Dally, B. Towles, "*Route packets, not wires: On-chip interconnection networks*," Proc. DAC, June 2001.

[3] M. Millberg, *et. al.* "*The Nostrum backbone - a communication protocol stack for networks on chip*," Proc. VLSI Design, Jan. 2004.

[4] D. Bertozzi, *et. al.* "*NoC synthesis flow for customized domain specific multiprocessor Systems-on-Chip*," IEEE Trans. on Parallel and Distributed Systems, vol. 16, Feb. 2005.

[5] E. Rijpkema *et al.* "Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip," Proc. DATE, March 2003.

[6] S. Murali, G. De Micheli, "*SUNMAP: A tool for automatic topology selection and generation for NoCs*," Proc. DAC, June 2004.

[7] J. Hu, R. Marculescu, "*Energy- and Performance-Aware Mapping for Regular NoC Architectures*," IEEE Trans. on Computer-Aided Design of ICs and Systems, 24(4), April 2005.

[8] A. Pinto, L. P. Carloni, A. L. Sangiovanni-Vincentelli, "*Efficient synthesis of networks on chip*," Proc. ICCD, Oct. 2003.

[9] U. Y. Ogras, R. Marculescu, "*Energy- and performance- driven customized architecture synthesis using a decomposition approach*," In Proc. DATE, March 2005.

[10] K. Srinivasan, *et. al.* "*Linear programming based techniques for synthesis of Network-on-Chip architectures*," Proc. ICCD, Oct. 2004.

[11] U. Y. Ogras, R. Marculescu, "*Application-Specific Network-on-Chip Architecture Customization via Long-Range Link Insertion*," Proc. ICCAD, Nov. 2005.

[12] C. J. Glass, L. M. Ni, "*The turn model for adaptive routing*," Proc. ISCA, May 1992.

[13] G. Kotsis, "*Interconnection Topologies for Parallel Processing Systems*," Proc.of Parallele Systeme und Algorithmen, 1993.

[14] D. J. Watts, S. H. Strogatz, "*Collective dynamics of small-world networks*", In Nature, vol. 393, pp. 440-442, 1998

[15] M. E. J. Newman, D. J. Watts, "*Scaling and percolation in the small-world network model*," Phys. Rev. E 60, 7332-7342, 1999

[16] R. Albert, A. Barabasi, "*Statistical mechanics of complex networks*," Reviews of Modern Physics, vol. 74, 2002.

[17] H. Fuks, A. Lawniczak, "*Performance of data networks with random links*," Math. and Computers in Simulation, vol. 51, 1999.

[18] L. Carloni, *et. al.* "*Theory of latency-insensitive design*," IEEE Trans. on Computer-Aided Design of ICs and Systems. 20(9), 2001.

[19] R. Dick, "Embedded system synthesis benchmarks suites (E3S)," http://www.ece.northwestern.edu/~dickrp/e3s/.

[20] H. G. Lee, *et. al.* "*Cycle-accurate energy measurement and characterization of FPGAs*," Analog IC and Signal Processing, vol. 42, 2005.